

Observatoire astronomique
de Strasbourg

Rapport technique

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

Introduction

Ce rapport a pour but de décrire le fonctionnement de l'application développées durant mes dix semaines de stage au Centre de Données astronomiques de Strasbourg. L'application en question est une interface graphique pour permettre de créer des fichiers d'ingestion de base de données.

Ce rapport présentera d'abord le contexte ayant abouti à la nécessité de créer cette application, puis, le cahier des charges des fonctionnalités souhaitées par le CDS. Ensuite, je présenterai le fonctionnement général de l'application en m'appuyant sur un schéma. Ensuite je présenterai les différentes interfaces de l'application et décrirai l'usage de chacun de leurs éléments graphiques. Et je terminerai par décrire le mode de fonctionnement entre elles des classes de l'application, avant de conclure avec les perspectives futures pour cette application.

Contexte

Le Centre de Données astronomique de Strasbourg est l'une des équipes de l'observatoire de Strasbourg. Son travail consiste à récolter, répertorier et distribuer des données astronomiques, afin de les rendre accessibles aux astronomes du monde entier. Pour ce faire, le CDS possède trois services : Simbad, VizieR et Aladin.

- Simbad est une base de référencement de noms pour des objets astronomiques ;
- VizieR regroupe des relevés astronomiques provenant d'observatoires ou de revues astronomiques dans une base de plus de 13 000 catalogues ;
- Aladin est un atlas tridimensionnel d'images du ciel.

Ces trois services sont massivement utilisés par le secteur de l'astronomie à l'échelle mondiale et font du CDS une référence dans son domaine.

Dans le cas de la base VizieR, sont de plus en plus répertoriés des données de type images ou spectres qui sont contenus dans des fichiers FITS, un format propre à l'astronomie. Or pour être exploitables par les astronomes, ces images et spectres ont besoin de méta-données. Pour ce faire, des fichiers de mapping qui les contiennent sont ajoutés dans VizieR avec les données FITS. Ces mappings sont composés d'une vingtaine d'informations sur la manière avec laquelle le relevé a été fait.

Jusqu'à présent, pour créer ces fichiers de mappings, les documentalistes du CDS utilisaient la commande Unix "vi". Pour proposer une autre méthode aux documentalistes et accélérer et faciliter leur travail, le CDS a donc décidé de créer une interface graphique Java pour permettre une création automatique des fichiers de mapping. Cette interface se doit d'être complète et simple pour rendre le travail des documentalistes simple et efficace.

La figure 1 représente l'arborescence du projet :

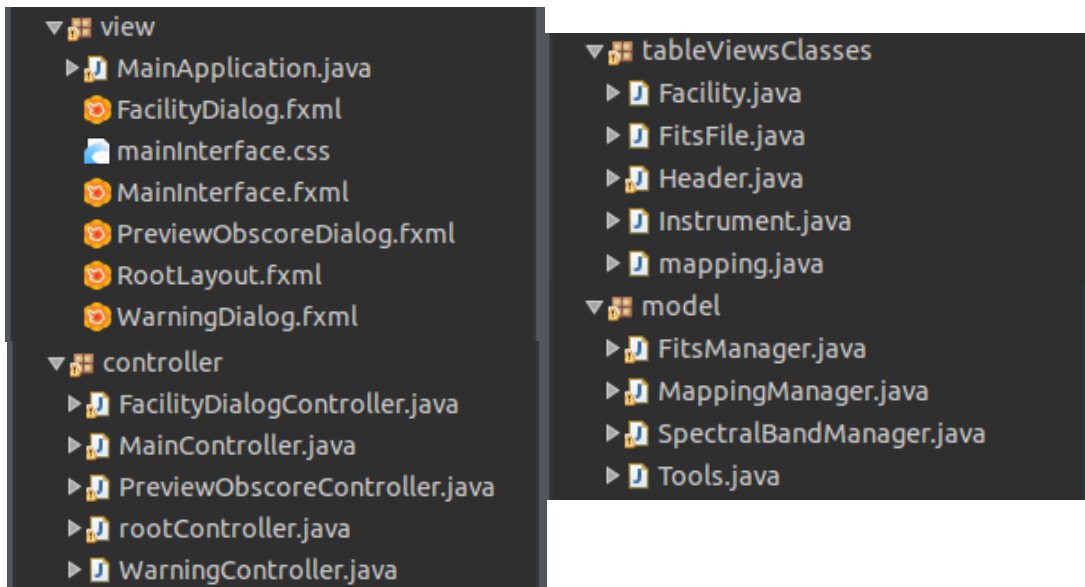


Figure 1 : organisation des packages du projet

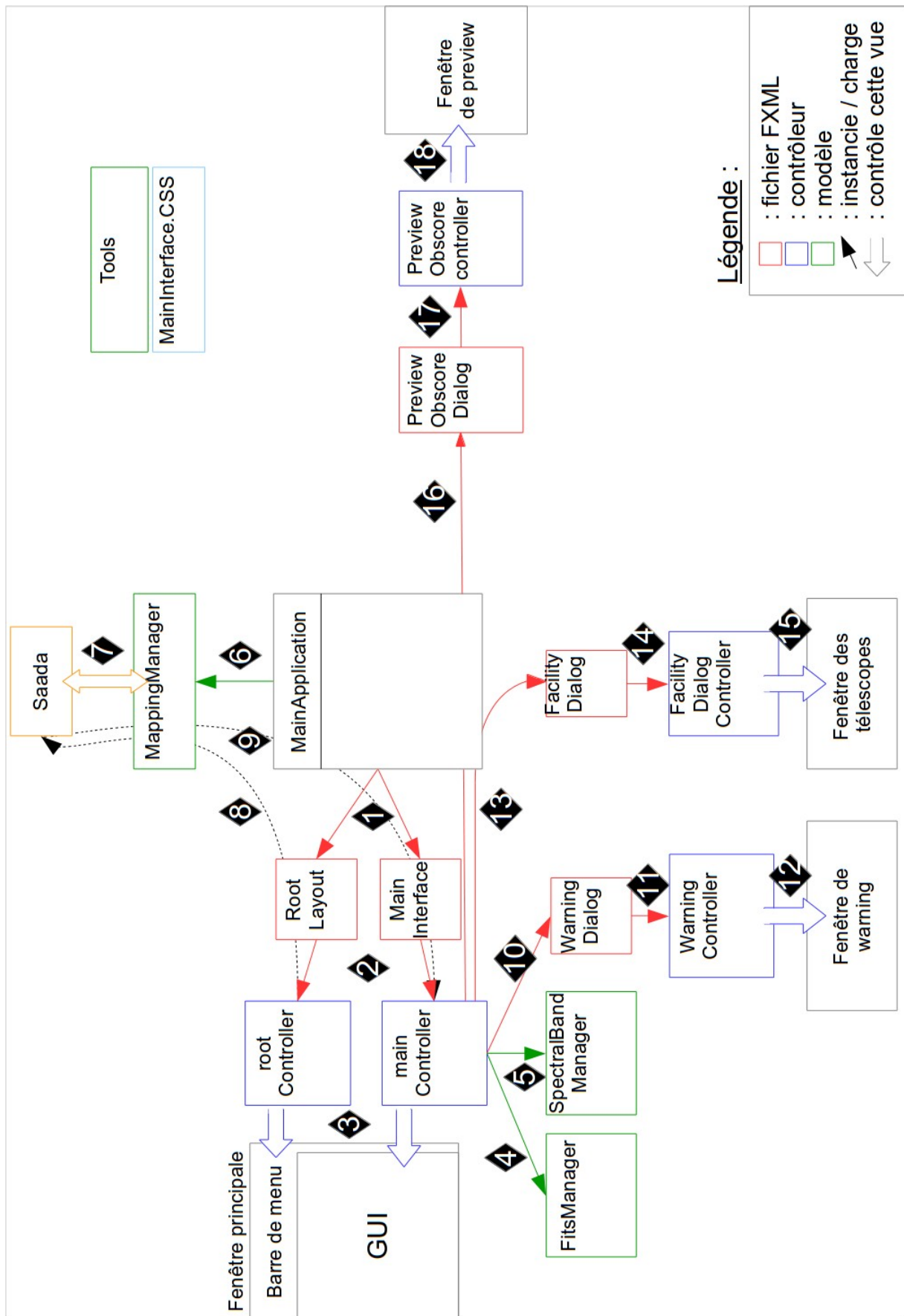


Figure 2 : interactions entre les classes du projet

<p>Voici le schéma du fonctionnement général de l'application :</p> <p>1 : Le main instance mainApplication qui charge rootLayout.fxml et mainInterface.fxml. Le premier est la barre de menu de l'application, le second est l'interface d'édition en elle-même.</p> <p>2 : rootLayout.fxml instancie le rootController durant son chargement. MainInterface.fxml fait de même avec le mainController.</p> <p>3 : les deux contrôleurs vont chacun gérer les interactions de l'utilisateur avec une partie de la fenêtre principale. Le rootController vas prendre en charge les menus et leurs fonctionnalité et le mainController vas s'occuper de toutes les interactions d'édérations qui proviennent sur le reste de la fenêtre.</p> <p>4 : le mainController instancie un FitsManager. Ce dernier permet de récupérer le contenu des header FITS</p> <p>5 : le mainController instancie également un SpectralBandManager, qui lui renvoi la liste des bandes spectrales ainsi que les limites de leurs longueur d'onde</p> <p>6 : la main application quant à elle instancie un mappingManager. Qui lui renvoi le</p>	<p>contenu des mappings. Qu'elle transmettra ensuite au mainController pour qu'il les affiche</p> <p>7 : le mappng manager utilise l'API Saada pour interprété les fichiers FITS et les mappings dont il renvoi le contenu à l'application</p> <p>8 :le rootController permet l'ouverture ou la création d'un mapping, qui sera interprété par le mappingManager avec l'aide de Saada</p> <p>9 : le mapping interprété sera renvoyé sous forme d'une liste de tableau contenant les données du mapping</p> <p>10 : lorsque que l'utilisateur appuie sur le bouton "Show warning", le mainController, par le biais de la mainApplication charge le fichier warningDialog.fxml</p> <p>11 : le fichier warningDialog.fxml instancie le warningController</p> <p>12 : le warningController gère la vue de la fenêtre de warning</p> <p>13 : lorsque que l'utilisateur clique sur le bouton du télescope, le mainController charge par le biais de la mainApplication le fichier facilityDialog.fxml</p>	<p>14 : le fichier facilityDialog.fxml instancie le facilityDialogController</p> <p>15 : le facilityDialogController gère la vue de la fenêtre des télescope</p> <p>16 : lorsque que l'utilisateur clique sur le bouton "Preview", le mainController charge par le biais de la mainApplication le fichier previewObscoreDialog.fxml</p> <p>17 : le fichier previewObscoreDialog.fxml instancie le previewObscoreController</p> <p>18 : le previewObscoreController gère la vue de la fenêtre de preview</p>
--	---	---

Présentation de interfaces

Je vais présenter ici les différentes fenêtres et leurs éléments graphiques majeurs majeurs. L'interface est en anglais pour des en raison de la présence potentielle au CDS de documentalistes ne parlant pas français.

1 - Interface principale

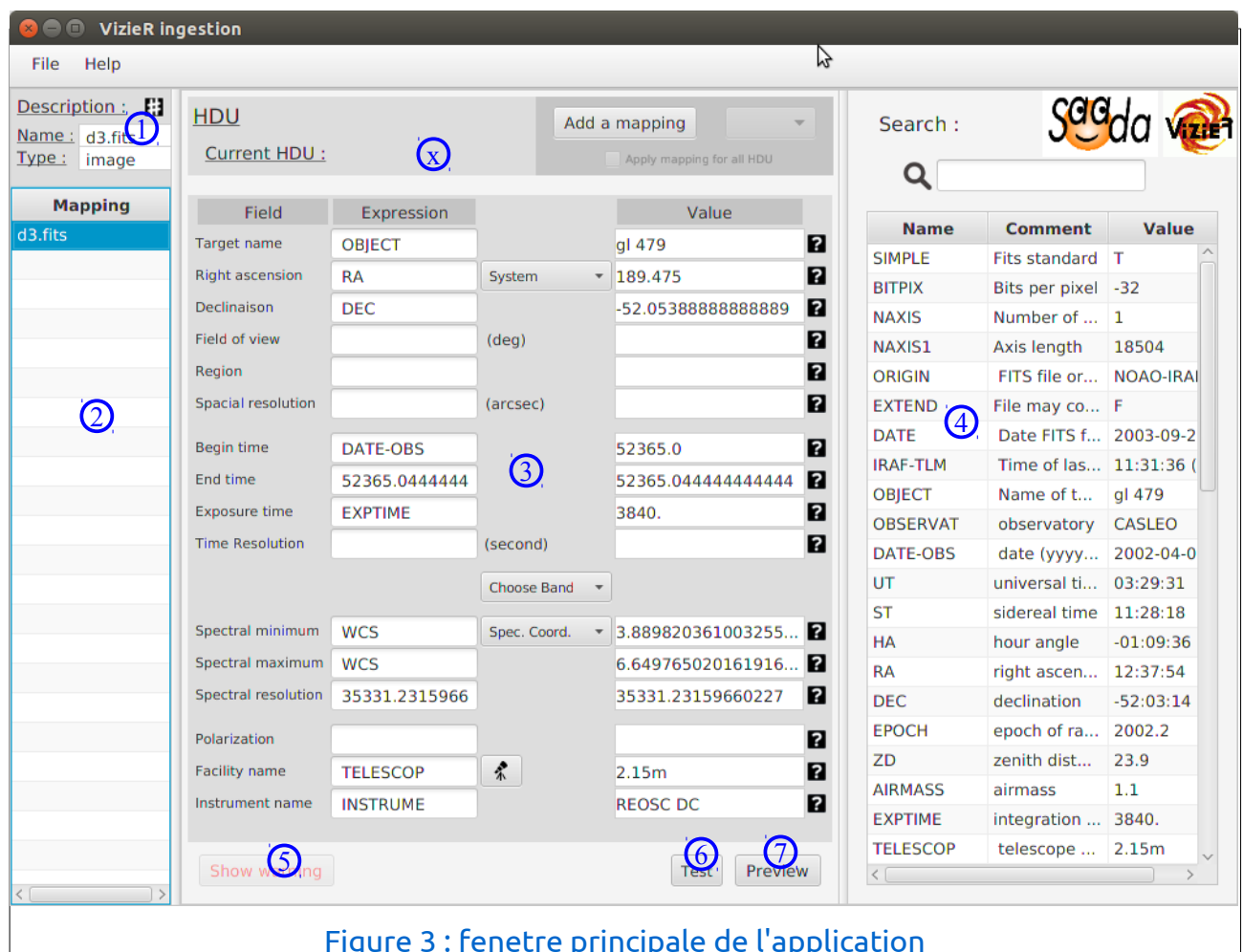


Figure 3 : fenetre principale de l'application

La figure 1 représente l'interface principale de l'application. Elle est divisée en deux fichiers FXML pour la générer : rootLayout.fxml permet de créer la barre de menu et mainInterface.fxml permet de créer le reste de l'interface.

La barre de menu mise à part, les principaux éléments sont :

- 1 : le cadre de description en haut à gauche permet d'afficher des informations

concernant le mapping;

- 2 : la menu de sélection de mapping, sur la gauche. Il permet de naviguer entre les mappings si plusieurs d'entre eux sont ouverts en même temps dans l'application;

- 3 : le cadre d'édition est la partie principale de l'interface. Il permet l'édition du mapping

- 4 : la liste des couples venant des entêtes-FITS, il s'agit d'une banque de données que l'on peut ajouter dans le mapping

- 5 : le bouton "show warning" permet l'ouverture de la fenêtre de warning

- 6 : le bouton "test" permet de valider la présence de valeurs clés dans le mapping

- 7 : le bouton "preview" permet l'ouverture de la fenêtre de preview

- x : le menu de gestion de mapping, qui ne sera pas développé ici, car son développement n'est pas terminé, il n'est pas encore fonctionnel;

1.1 - Le cadre de description

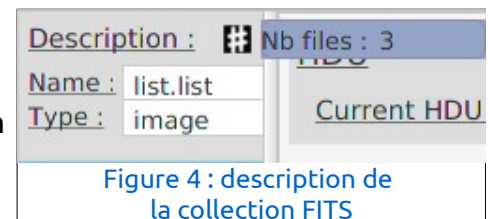
Il s'agit d'une simple présentation d'information concernant la collection FITS ouverte dans l'interface.

On y voit :

- un label contenant le nom de la collection

- un label indiquant le type de la collection
(image ou spectre)

- un label mobile qui permet de voir le nombre de fichiers dans la collection, donc qui sont concernés par le mapping.



1.2 - La liste de sélection de mapping

Un seul fichier peut en contenir plusieurs mappings, dans ce cas là, chaque mappings affectera différents fichiers de la même collection. Si l'utilisateur ouvre un de ces fichiers, l'interface permet la navigation entre ces différents mappings pour les éditer indépendamment.

Chaque mapping est une ligne de la liste, lorsque que l'utilisateur clique sur une ligne, le mapping en cours d'édition change. Le précédent est sauvegarder et les valeurs du nouveau sont affichées à l'écran.

Mapping
o1.fits
o2.fits
o3.fits
o4.fits

Figure 5 : liste des mappings

1.3 - Le cadre d'édition

Le cadre d'édition est la partie principale de la fenêtre. Il a la forme d'un tableau à 5 colonnes qui contiennent pour chaque attribut du mapping :

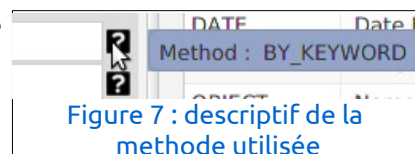
- le nom de l'attribut
- un champ éditable contenant l'expression de l'attribut (qui permettra d'obtenir la valeur)
- une colonne contenant des options supplémentaires :

- 1 : une liste déroulante permettant de sélectionner le système de coordonnées utilisé pour les données de visée de l'observation
- 2 : les unités utilisées pour les valeurs
- 3 : une liste déroulante permettant de sélectionner la bande spectral utilisée
- 4 : une liste déroulante permettant de choisir l'unité dans laquelle les limites de la bande spectrale sont exprimées
- 5 : le bouton de télescope qui ouvre la fenêtre de gestion des télescopes

- une colonne qui contient des labels avec les valeurs du mapping pour le premier fichier FITS de la collection, à titre informatif
- une colonne contenant des ImageViews qui indique la méthode pour renseigner l'attribut lorsque l'utilisateur passe la souris dessus.

Field	Expression	Value	
Target name	OBJECT	gl 479	?
Right ascension	RA	189.475	?
Declinaison	DEC	-52.05388888888889	?
Field of view	(deg)		?
Region			?
Spacial resolution	(arcsec)		?
Begin time	DATE-OBS	52365.0	?
End time	52365.0444444	52365.044444444444	?
Exposure time	EXPTIME	3840.	?
Time Resolution	(second)		?
	Choose Band		?
Spectral minimum	WCS	3.889820361003255...	?
Spectral maximum	WCS	6.649765020161916...	?
Spectral resolution	35331.2315966	35331.23159660227	?
Polarization			?
Facility name	TELESCOP	2.15m	?
Instrument name	INSTRUME	REOSC DC	?

Figure 6 : cadre d'édition



1.4 - Le gestionnaire d'en-tête FITS

Cette partie permet de stocker les couples clé-valeur venant des en-têtes FITS. Ces couples sont stockés dans une liste donc chaque ligne est un couple. Les colonnes de la liste sont le nom, le commentaire et la valeur. Quand une valeur est ajoutée, le nom du couple est placé dans le champ de l'expression et la valeur du

couple dans le label de la valeur. La méthode pour cet attribut est alors "par mot-clé".

Au dessus de la liste, se trouve un champ de recherche. Lorsque l'utilisateur tape un mot dans ce champ, la liste se met à jour avec les couples contenant la chaîne qu'il vient de taper.

Pour ajouter un des couple au mapping, l'utilisateur peut utiliser la technique du glisser-déposer (drag and drop). Pour cela il lui suffit de sélectionner l'une des ligne et de la glisser vers un des champs pour affecter cette ligne à l'attribut en question.

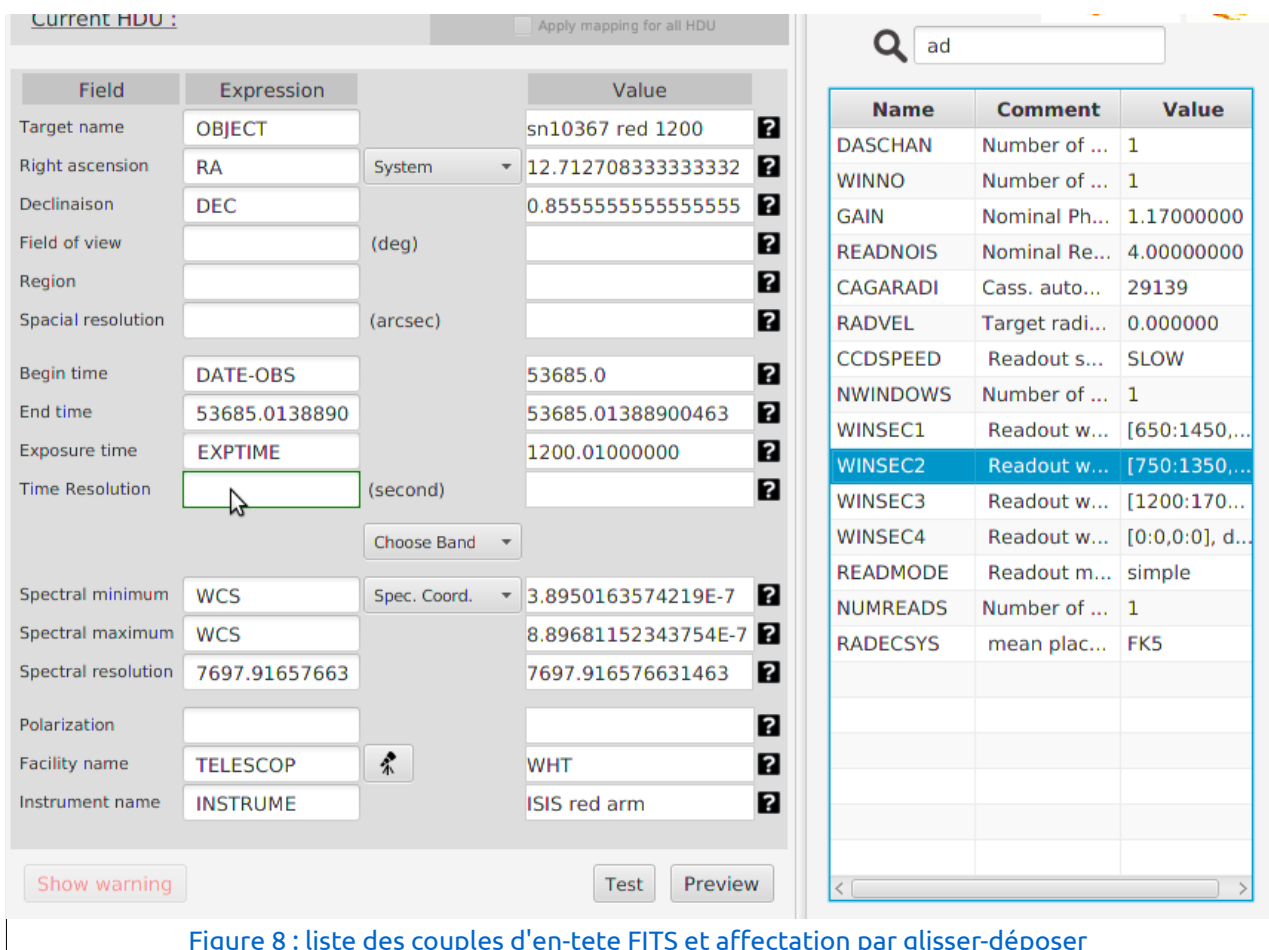


Figure 8 : liste des couples d'en-tete FITS et affectation par glisser-déposer

Le cadre du champ en question sera alors en vert pour indiquer que l'affectation est possible.

1.5 - Le bouton show warning

Lors de l'ouverture d'un mapping, il se peut que l'API renvoie des erreurs ou des alertes pour diverses raisons . La plus courante est que les fichiers contenus dans la

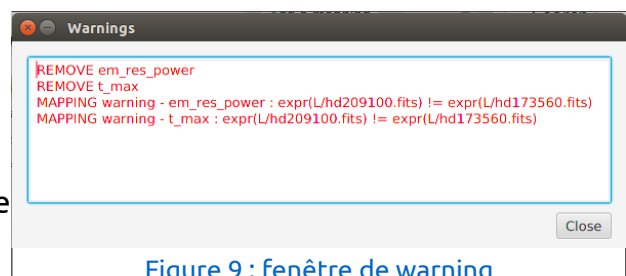


Figure 9 : fenêtre de warning

collections sont trop différents et que par conséquent, les valeurs qu'ils contiennent sont également trop différentes. Dans ce cas là, l'interface n'affiche pas clés valeurs pour ces attributs et indique à l'utilisateur quels sont les attributs concernés par la fenêtre de warning.

Cette fenêtre est une simple zone de texte indiquant quels sont les messages d'erreurs renvoyés par l'API.

1.6 - Le bouton test

Ce bouton permet simplement de vérifier si certaines données clés sont présentes dans le mapping, car sans ces informations, le mapping n'as pas de grand intérêt. Il faut notamment que l'ascension droite et la déclinaison ainsi que le système de coordonnées soit tous trois présents, sinon le test les contournera et s'affichera en rouge. Les autres informations capitales sont les limites de la bande spectrale observée et l'unité dans la quelles elles sont exprimées. Si l'une de ces trois données manquent, le test et les champs correspondants seront entourés en orange.

Il faut noter que l'absence des ces données n'empêcherons pas la sauvegarde du fichier de mapping, elles sont indiqués à titre informatifs.

1.7 - Le bouton de preview

Ce bouton ouvre la fenêtre de preview. Cette fenêtre affiche le résultat de chaque attribut pour

Field	Expression	Value
Target name	OBJECT	gl 845
Right ascension	RA	330.92083333333335
Declinaison	DEC	-56.78
Field of view	(deg)	
Region		
Spacial resolution	(arcsec)	
Begin time	DATE-OBS	52601.0
End time		53685.01388900463
Exposure time	EXPTIME	600.
Time Resolution	(second)	
Spectral minimum	WCS	3.889752074837635...
Spectral maximum	WCS	6.649790545135746...
Spectral resolution		7697.916576631463
Polarization		
Facility name	TELESCOP	2.15m
Instrument name	INSTRUME	REOSC DC

Buttons: Show warning, Test, Preview

Figure 10 : résultat du test

et_name	s_ra	s_dec	s_fov	s_r...	s_r...	t_min	t_max	t_e...	t_r...	em_min	em_max	em...	pol_states	facility_name	insf
00122	'Not...	'Not...	'NotSet'			'557...	'557...			'0.000...	'0.000...		-1	'SDSS'	'INS
00122	'Not...	'Not...	'NotSet'			'557...	'557...			'0.000...	'0.000...		-1	'Liverpool Tel...	'INS
00122	'Not...	'Not...	'NotSet'			'557...	'557...			'0.000...	'0.000...		-1	'Liverpool Tel...	'INS

Figure 11 : fenêtre de preview

tous les fichiers FITS de la collection en fonction du mapping.

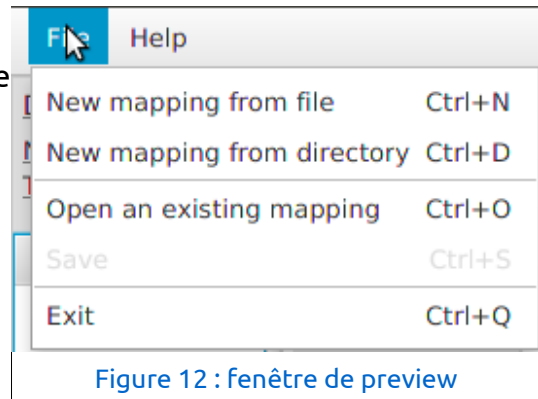
Cette fenêtre est un tableau dont chacune des lignes est un fichier FITS et chaque colonne l'un des attributs du mapping.

Ce tableau contient donc les valeurs de tous les attributs de tous les fichiers FITS.

2 - Les menus

Les menus de l'application permettent de faire toutes les manipulations extérieures à l'application :

- créer un mapping depuis un fichier FITS ou un fichier liste
- créer un mapping depuis un répertoire
- ouvrir un fichier de mapping déjà existant
- sauvegarde le mapping en cours d'édition dans un fichier
- quitter l'application.



Il est à noter que trois options sont disponibles pour l'ouverture du mapping. Ouvrir un nouveau mapping et ouvrir un mapping existant sont séparés car le programme traite différemment ces deux possibilités.

Créer un mapping depuis un répertoire est séparé de la création d'un mapping depuis un fichier, car l'outil permettant de choisir la source, le fileChooser, ne peut pas prendre indifféremment un répertoire ou un fichier.

En revanche, ouvrir un mapping depuis un fichier liste ou un fichier FITS sont placés ensemble car le traitement par l'interface et par le fileChoose sont les mêmes dans les deux cas.

3 - Le gestionnaire de télescopes

La fenêtre de sélection de télescope permet d'entrer très rapidement les données du télescope et de l'instrument utilisé ainsi que la bande spectrale observée.

Comme on peut la voir sur la figure 11, cette fenêtre est divisée en deux



Figure 13 : fenêtre des télescopes

parties :

La partie de gauche gère les télescopes et celle de droite les instruments. ⑤

Voici les éléments de cette fenêtre :

- 1 : une liste contient tous les télescopes référencés dans VizieR (récupérés grâce à une requête par URL).
- 2 : un champ permet de n'afficher dans la liste que les télescopes contenant la chaîne contenu dans ce champ
- 3: un seconde liste contient tous les instruments (nom, bande spectrale, limites de cette bande) que possède le télescope sélectionné dans la première liste
- 4 : un champ permet également de rechercher un télescope dans la liste
- 5 : une fois le télescope et l'instrument choisi, le bouton "set facility" permet d'appliquer les valeurs sélectionnées aux champs du mapping dans l'interface principale.

Fonctionnement des classes

Je présenterai ici les principales fonctions des différentes classes de l'application.

1 - Les classes générales

1.1 - MainApplication

Cette classe est la classe principale de l'application et a trois fonctions principales :

- charger les fichiers FXML. Elle charge ces fichiers et les configure pour qu'ils s'affichent correctement dans les fenêtres. Les méthodes "show" sont chargées de cette fonction;
- instancier le mapping manager;
- permettre la communication entre les différentes classes. Elle récupère les contrôleurs qui sont affectés aux vues par les fichiers FXML qui les instancie eux-même. Elle permet ensuite aux contrôleurs de communiquer avec Saada par le biais du mappingManager.

1.2 - Tools

Tools est une classe contenant des méthodes générales utilisées à divers endroits dans le programme. Elle permet :

- de vérifier qu'une instance d'un objet est contenu dans un tableau de cet objet ;
- de voir si un string en contient une autre sans que cette vérification soit sensible à la casse ;
- de vérifier si un string est un nombre (int ou double) ;
- d'ajouter des simples quotes sur un string en vérifiant si ce dernier en possède ou non ;
- de supprimer des simples quotes sur une string en vérifiant si cette dernière en possède ou non.

1.3 - Les classes pour les tableViews

JavaFX propose un outils graphique très pratique pour faire des listes d'éléments : les tableViews.

Pour gérer facilement et efficacement les données contenues dans ces tableViews, il faut affecter à ces listes des objets. Par conséquent, 5 classes ne

servent qu'à fournir les données pour ces listes et ne contiennent que les quelques attributs nécessaires pour la liste en question.

Ainsi :

- FitsFile permet de remplir la liste des mappings sur la droite de l'interface principale ;
- Header permet de remplir la liste des couples provenant des en-tête FITS sur la gauche de l'interface principale ;
- Facility permet de remplir la liste des télescopes dans la fenêtre de gestion des télescopes ;
- instrument permet de remplir la listes des instruments dans la même fenêtre ;
- mapping permet de remplir la liste des valeurs des mappings dans la fenêtre de preview.

2 - Les contrôleurs

2.1 - mainController

Cette classe est le contrôleur principal, il gère l'édition sur l'interface principale, pour ce faire, il doit :

- lier tous les éléments à des variables pour pouvoir permettre les interactions et initialiser les différent listeners et les liens entre les éléments et les données (les listes de données pour les listes graphiques dans l'interface);
- gérer le Drag and Drop dans l'interface pour éditer le mapping ;
- permettre d'ouvrir la fenêtre de preview grâce au bouton "preview" ;
- permettre d'ouvrir la fenêtre de warning grâce au bouton "show warning" ;
- permettre d'ouvrir la fenêtre de télescope grâce au bouton "télescope" ;
- permettre l'édition du mapping par l'écriture dans les champs du tableau ;
- mettre à jour les méthodes en fonction de la manière avec laquelle les information ont été entrées ;
- permettre de rechercher dans la liste du header FITS grâce au champ de

- rechercher ;
- afficher la description du mapping ;
- tester le mapping en vérifiant si les valeurs clés sont présentes.

2.2 - warningController

Ce contrôleur sert à afficher les erreurs et warnings renvoyées par l'API Saada lors de l'ouverture d'une collection de mapping dont les fichiers sont très différents. Ce contrôleur doit donc :

- récupérer la sortie d'erreur redirigée ;
- afficher le contenu de cette sortie d'erreur dans une zone de texte dans la fenêtre.

2.3 - previewController

Ce contrôleur récupère la liste des valeurs de chaque mapping ouvert dans l'application et l'affiche dans un tableau. Ce contrôleur doit donc :

- récupérer la liste des mappings ;
- ajouter une ligne dans le tableau pour chaque mapping.

2.4 - facilityController

Ce contrôleur permet d'intégrer facilement le télescope, le type d'instrument et leurs valeurs de bandes spectrales associées dans la mapping. Pour cela, cette classe doit :

- récupérer la liste des télescopes qui sont référencés dans VizieR et l'afficher dans un premier tableau ;
- une recherche peut être faite dans la liste pour n'afficher que les télescopes dont le nom inclue la chaîne entrée dans le champ de recherche ;
- faire une requête pour récupérer les noms des instruments, la bande qu'il observe et les limites de cette bande, lorsque l'utilisateur sélectionne un télescope dans la liste précédente et les afficher dans la liste des instruments ;

- permettre une recherche dans la liste des instruments ;
- envoyer les données de l'instrument sélectionné dans l'interface lorsque que l'utilisateur clique sur le bouton "set facility".

2.5 - rootControlleur

Ce contrôleur permet de gérer les menus et leurs fonctionnalités. Ces fonctionnalités sont :

- ouvrir les mappings depuis toutes les collections possibles : fichier FITS, fichier liste, répertoire ou fichier mapping ;
- sauvegarder les mappings en édition dans un fichiers ;
- fermer l'application.

3 - Modèle

3.1 - FitsManager

Cette classe permet de récupérer le contenu des header FITS du fichier entré en paramètre.

3.2 - Spectral Band manager

Cette classe permet de récupérer les bandes spectrales et leurs limites de longueur d'onde. Pour cela, cette classe doit :

- accéder à la base de données du CDS avec un URL ;
- interjeter le JSON retourné et stocker les données dans des listes ;
- permettre l'accès à ces listes par le mainController par le biais de getters ;

3.3 - Mapping manager

Cette classe permet l'ouverture pour en récupérer les données et la description du mapping ainsi que la sauvegarde des données de l'interface vers un fichier de mapping ; mais également récupérer les valeurs du mapping en fonction

des expressions et des fichiers FITS concernés.

Cette classe doit donc :

- se connecter à la base Saada pour permettre l'interprétation des mapping ;
- stocker le jsonObject renvoyé par l'API Saada ;
- interpréter le jsonObject en une liste de tableau qui contient les expressions, les valeurs et les méthodes de chaque attribut ;
- récupérer la description du mapping dans le jsonObject renvoyé par l'API ;
- mettre à jour le jsonObject avec les données contenues dans l'interface ;
- retourner le jsonObject afin de permettre la sauvegarde du mapping dans un nouveau fichier de mapping ;
- retourner les valeurs du mapping pour permettre la preview de s'afficher.

Conclusion et prévisions

L'application possède la plupart des fonctionnalités nécessaires à son utilisation basique : ouverture, édition facilité, sauvegarde. Mais le travail sur le projet reste conséquents, outre de nombreux bugs à corriger, quelques fonctionnalités peuvent encore être implémenter pour améliorer l'interface.

Ces fonctionnalités n'étaient pas incluses dans le sujet du stage, car la masse de travail était trop importante pour 10 semaines de stage. Parmi ces fonctionnalités, il y a notamment un test plus poussé du mapping, une meilleur gestion des mappings lorsque plusieurs sont ouverts en même temps dans l'application ou encore une possibilité d'ajouter automatiquement le mapping et les fichiers FITS dans la base VizieR.

Compte tenu du travail qu'il reste à faire, mon tuteur m'a proposé un CDD de mois cet été au CDS afin de poursuivre le développement réalisé au cours du stage.

