



Rapport de Stage

Évolution et mise à jour de l'architecture AÏDA

Beugnot Julien

Année universitaire 2006-2007

Université Nancy 2 - IUT Nancy-Charlemagne

Licence Professionnelle Concepteur Intégrateur de Systèmes Internet / Intranet



Observatoire Astronomique
11, Rue de l'université
67000 Strasbourg

Rapport de Stage
Avril – Juin 2007

Évolution et mise à jour de l'architecture AÏDA

Beugnot Julien
Année universitaire 2006-2007

Tuteurs entreprise : André Schaaff, Cyril Pestel

Tuteur enseignant : Denis Roegel

Université Nancy 2 - IUT Nancy-Charlemagne

Licence Professionnelle Concepteur Intégrateur de Systèmes Internet / Intranet

Remerciements

Je remercie, pour le bon accueil, toutes les personnes que j'ai pu côtoyer durant toute la période du stage à l'Observatoire de Strasbourg. Principalement mes tuteurs André Schaaff et Cyril Pestel, pour tous leurs bons conseils.

Table des matières

1.Introduction.....	6
2.Présentation du laboratoire.....	7
2.1.Structure juridique.....	7
2.2.Présentation générale.....	7
2.3.Organisation du personnel.....	8
2.4.Centre de Données astronomiques de Strasbourg (CDS).....	8
2.5.Les services du CDS.....	10
2.6.Observatoire virtuel.....	13
3.Présentation du sujet.....	14
3.1.Analyse de l'existant.....	14
3.2.Analyse des nouveaux besoins.....	16
4.Cahier des charges.....	17
4.1.Analyse de l'application	17
4.2.Nettoyage des sources.....	17
4.3.Mise en place de JLOW 2.....	17
4.4.Ajout de fonctionnalités.....	17
5.Description de la mise en oeuvre.....	18
5.1.Installation de l'environnement de travail	18
5.2.Analyse de l'application	19
5.3.Mise en place de JLOW 2.....	22
6.Bilan.....	25
6.1.Bilan du stage.....	25
6.2.Bilan humain personnel.....	25
6.3.Bilan technique personnel.....	25
7.Conclusion.....	26
Mots-Clés.....	27
Références.....	28

1. Introduction

Ce rapport concerne mon stage de Licence Professionnelle Concepteur Intégrateur de Système Intranet-Internet. Celui-ci s'étendait sur la période du 10 avril au 30 juin et il a été prolongé jusqu'au 31 juillet. Il a été accompli à l'Observatoire Astronomique de Strasbourg et plus particulièrement au Centre de Données astronomiques de Strasbourg (CDS).

Le CDS développe et suit plusieurs projets notamment le projet AIDA (Astronomical Image processing Distributed Architecture) sur lequel j'ai travaillé. AIDA est une architecture qui permet à un utilisateur non initié d'utiliser des outils hétérogènes de façon simple et uniforme. Il permet également de construire des chaînes de traitement grâce à la librairie JLOW. Or la version actuellement utilisée dans AIDA est obsolète, l'objectif principal de mon stage est donc de faire évoluer cette architecture pour améliorer ses capacités et son utilisation.

Je débuterai ce rapport par une présentation du centre de recherche dans lequel j'ai travaillé, puis je présenterai le sujet du projet en précisant les besoins et les objectifs. J'apporterai des précisions sur la solution développée, puis je décrirai la mise en oeuvre du projet.

2. Présentation du laboratoire

2.1. Structure juridique

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

2.2. Présentation générale

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

Enseignements dispensés :

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

La partie publique de l'Observatoire, le Planétarium, est ouvert pour la vulgarisation de l'astronomie.

L'Observatoire se compose de quatre équipes de recherche :

a) Hautes énergies

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

b) Étoiles et systèmes stellaires

Les recherches menées par l'équipe « Étoiles et systèmes stellaires » recouvrent un domaine étendu, incluant les étoiles, les milieux interstellaires, la Galaxie et les galaxies proches. De l'étoile, objet individuel, l'intérêt s'est porté aux groupes d'étoiles, témoins de l'évolution stellaire mais aussi traceurs des grandes structures de la Voie Lactée.

c) Galaxies

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

d) Le Centre de Données (CDS)

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

2.3. Organisation du personnel

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

2.4. Centre de Données astronomiques de Strasbourg (CDS)

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

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

Le but du CDS est de :

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

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

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

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

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

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

2.5. Les services du CDS

a) SIMBAD

The screenshot shows the SIMBAD Query Form in Microsoft Internet Explorer. The browser window title is "Simbad Query Form - Microsoft Internet Explorer". The address bar shows "http://simbad.u-strasbg.fr/simbad.pl". The page content includes the SIMBAD logo and navigation links. The main form area is titled "1. Enter an identifier" and contains a text input field with "M51" entered. Below the input field are several options: "only this object" (selected), "radius" (10 arc min), and "epoch" (2000). There are "SUBMIT" and "CLEAR" buttons. Below this is section "2. Optional output options" with checkboxes for "measurements" and "bibliography", and a range selector for years (1963 to 2001). At the bottom, there are options for "Display coordinates" with three frames for coordinate systems (FK5, FK4, Galactic) and epochs (2000.0, 1950.0, 2000.0). Two yellow callout boxes are present: one pointing to the search bar with the text "Specify a target" and another pointing to the "SUBMIT" button with the text "...and submit".

Figure 1.1 : Page Web permettant d'effectuer une requête sur Simbad

Simbad est une base de données de référence pour les identifications et la bibliographie d'objets astronomiques. Simbad contient plus de 11,2 millions d'identificateurs pour plus de 3,82 millions d'objets différents. Pour chaque objet figurent dans la base quelques mesures (position, magnitude dans différents domaines de longueurs d'ondes), ainsi que les références bibliographiques où l'objet est cité (plus de 210 000 articles sont concernés). L'utilisateur peut choisir le format du fichier des résultats de la requête (Fig. 1.1). Simbad peut générer des fichiers HTML, XML ou XLS (fichiers Excel).

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

b) VIZIER

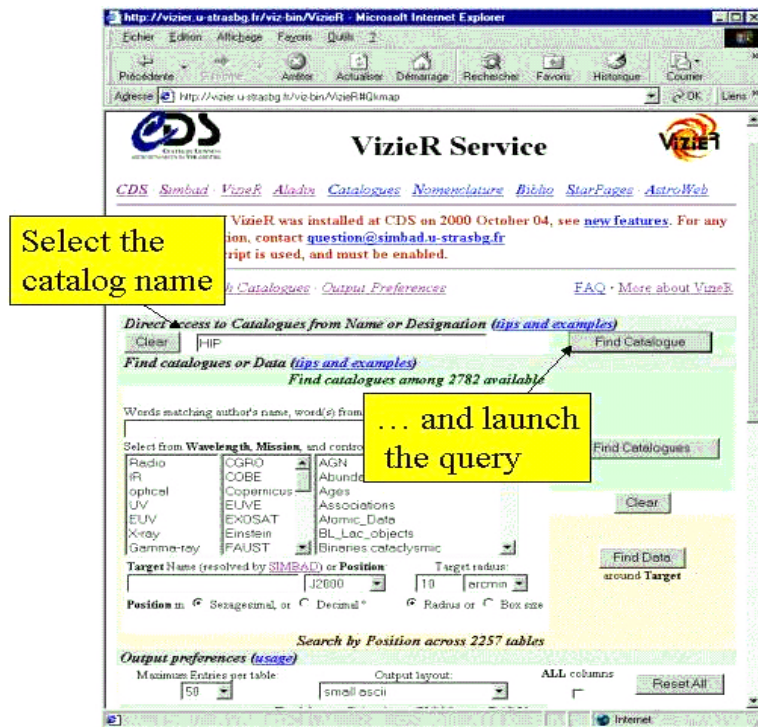


Figure 1.2 : Page Web permettant d'effectuer une requête sur VizieR

VizieR est une base de données rassemblant plusieurs milliers de catalogues astronomiques sous un format homogène. Une description standardisée du contenu des catalogues permet leur inclusion dans un système de gestion de base de données relationnelles. Un ensemble de liens, entre les tables de VizieR, et avec des services externes (bibliographiques, archives externes, serveurs d'images), permettent de naviguer entre les données des catalogues et d'autres données associées (Fig. 1.2). Il faut noter que les très grands catalogues (plusieurs milliards d'enregistrements) ne sont pas gérés par un SGBD relationnel pour des raisons de performances, c'est pourquoi des outils spécifiques sont utilisés (stockage des données sous forme binaire et accès via des programmes dédiés).

c) ALADIN

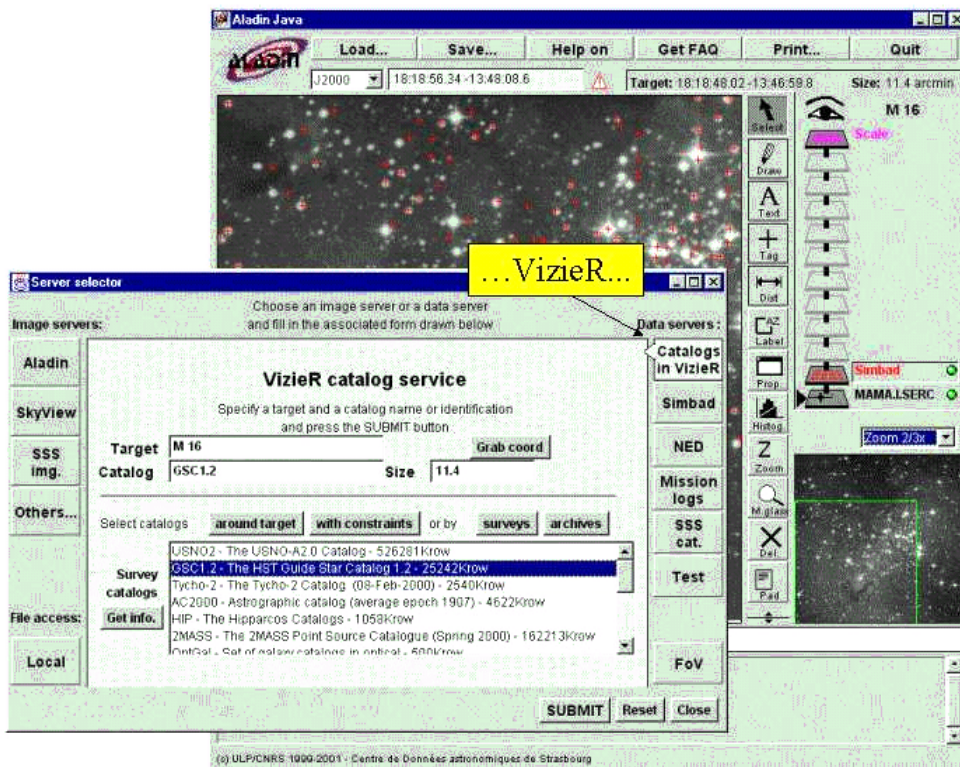


Figure 1.3 : Exemple d'utilisation d'Aladin

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

2.6. 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 devenue une priorité car elle permet un accès aisé à l'information, un croisement des données de toutes natures et elle conduit au concept d'Observatoire Virtuel. L'astronome disposera à terme de nouveaux instruments utiles à sa recherche au travers d'une simple interface Web. Afin d'y parvenir, de nombreuses collaborations sont indispensables afin d'aboutir aux consensus indispensables à la pose des bases nécessaires à la "construction" de l'Observatoire Virtuel.

Les projets nationaux (OVF pour la France, GAVO pour l'Allemagne, le NVO pour les États-Unis, ...) et transnationaux (ESA VO, ESO VO, ...) sont réunis au sein de l'International Virtual Observatory Alliance. Ils participent à l'élaboration de Recommandations dans divers domaines (Data Model, Accès aux données, Sémantique, Grilles, etc.) au travers de groupes de travail qui se réunissent semestriellement.

A titre d'exemple significatif, la première recommandation, "VOTable", décrit la formalisation de tables de données astronomiques en XML. Celle-ci a été adoptée par de nombreux fournisseurs de données et intégrée dans les outils de l'OV.

3. Présentation du sujet

3.1. Analyse de l'existant

Le CDS a développé l'architecture AIDA (Astronomical Image processing Distributed Architecture), dans le cadre du projet Masses de Données Astronomiques (MDA). Ce projet a été développé pendant deux ans par un ingénieur et a fait également l'objet d'un stage de cinq mois.

Dans le monde de l'astronomie, diverses techniques de traitement d'images sont explorées, et des implémentations de ces techniques sont réalisées dans des outils indépendants et hétérogènes. L'objectif de l'architecture AIDA est d'uniformiser l'accès à ces outils pour leur donner une meilleure visibilité et permettre leur utilisation par un plus grand nombre de personnes.

L'architecture globale pour l'intégration des outils est découpée en blocs fonctionnels et leurs interactions, ce qui est présenté dans la figure 1.4.

- **Outils** : il s'agit des divers programmes de traitement d'image, pas forcément homogènes dans leur implémentation, dont on a connaissance des entrées / sorties et des paramètres attendus, appelé aussi port. On pourra parler d'outils, de tâches ou encore de programmes selon le contexte (Figure 1.5) .
- **Accès aux outils** : c'est la couche d'abstraction qui propose des méthodes d'accès uniformes pour l'exécution des outils. Il s'agit du point d'entrée pour qu'une application cliente fasse une demande à la partie du système chargée de l'exécution.
- **Exécution des outils** : cette partie est chargée de l'exécution effective des programmes au sein du système. Cette partie sera décomposée de façon à permettre la distribution des tâches, pour utiliser les ressources de type cluster / grille dans les cas avantageux.
- **Interface utilisateur** : il s'agit d'une application offrant une interface graphique pour l'utilisation des modules.
- **Serveur de workflow** : serveur JLOW (Java Library fOr Workflow) recevant la description d'un workflow d'outils du système, et qui se charge de son exécution (chaque tâche à exécuter pour compléter le workflow fera l'objet d'une requête en utilisant une méthode d'accès standard (CGI), tout comme les autres applications clientes le font).
- **Aladin et autres clients** : applications externes pouvant avoir l'usage des outils de traitement d'images via les méthodes d'accès proposées.

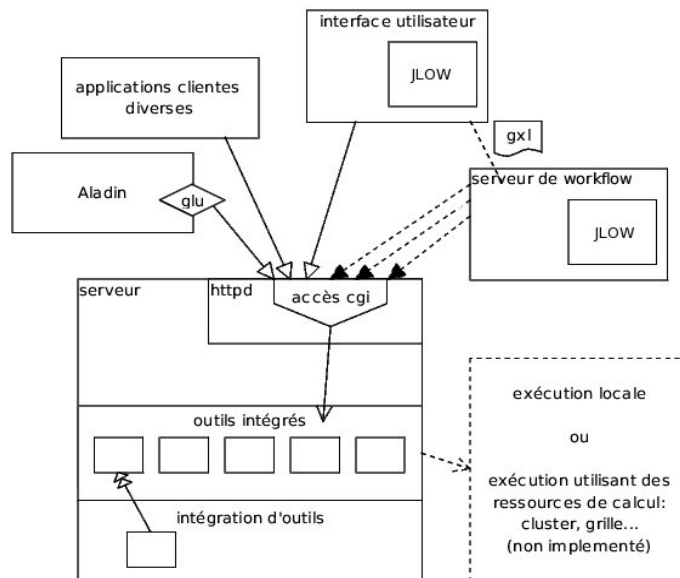


Figure 1.4 : Vue d'ensemble de l'architecture.

Cette architecture a été réalisée par Jean-Julien Claudon, ingénieur informaticien, excepté le serveur de workflow qui a été réalisé par Cyril Pestel.

L'interface utilisateur propose deux méthodes d'accès aux outils, l'une permettant d'accéder à un outil à la fois (Figure 1.6), et l'autre permettant d'y accéder en workflow, c'est-à-dire en enchaînant l'exécution de plusieurs outils (Figure 1.7). L'interface peut accéder à un serveur de workflow dont l'adresse est renseignée dans le fichier de configuration, mais elle peut également lancer en local un serveur de workflow embarqué.

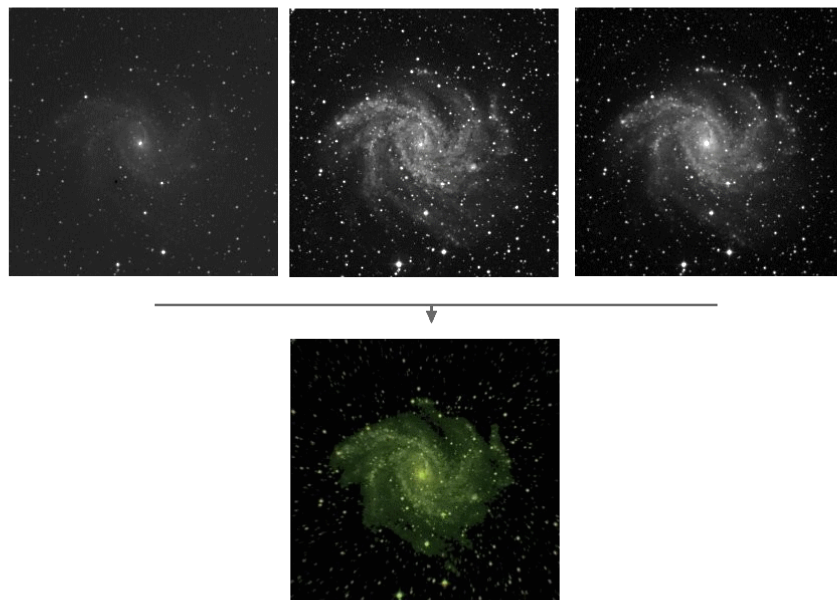


Figure 1.5 : Exemple d'utilisation d'un outil : Visualisation TSL (Teinte Saturation Lumière), les trois premières photos en entrées donnent la quatrième en résultat.

3.2. Analyse des nouveaux besoins

L'interface utilisateur est basée sur une version obsolète de JLOW, en effet, lors de la réalisation de cette dernière JLOW était en phase de développement. Ainsi, la gestion du workflow a été en partie définie dans AïDA mais également dans les versions plus récentes de JLOW. Un des nouveaux besoins est donc le nettoyage des sources d'AïDA qui permettra une définition uniforme des objets communs de AïDA et JLOW. L'intégration de la nouvelle version de JLOW apportera ainsi de nouvelles fonctionnalités, comme la sauvegarde et le chargement d'un workflow existant au format GXL¹. De plus, le nettoyage des sources devra également porter sur une dissociation plus marquée de l'aspect graphique et des traitements. Cela permettra ainsi une incorporation plus aisée des futures mises à jour de JLOW et une meilleure maintenance de l'application.

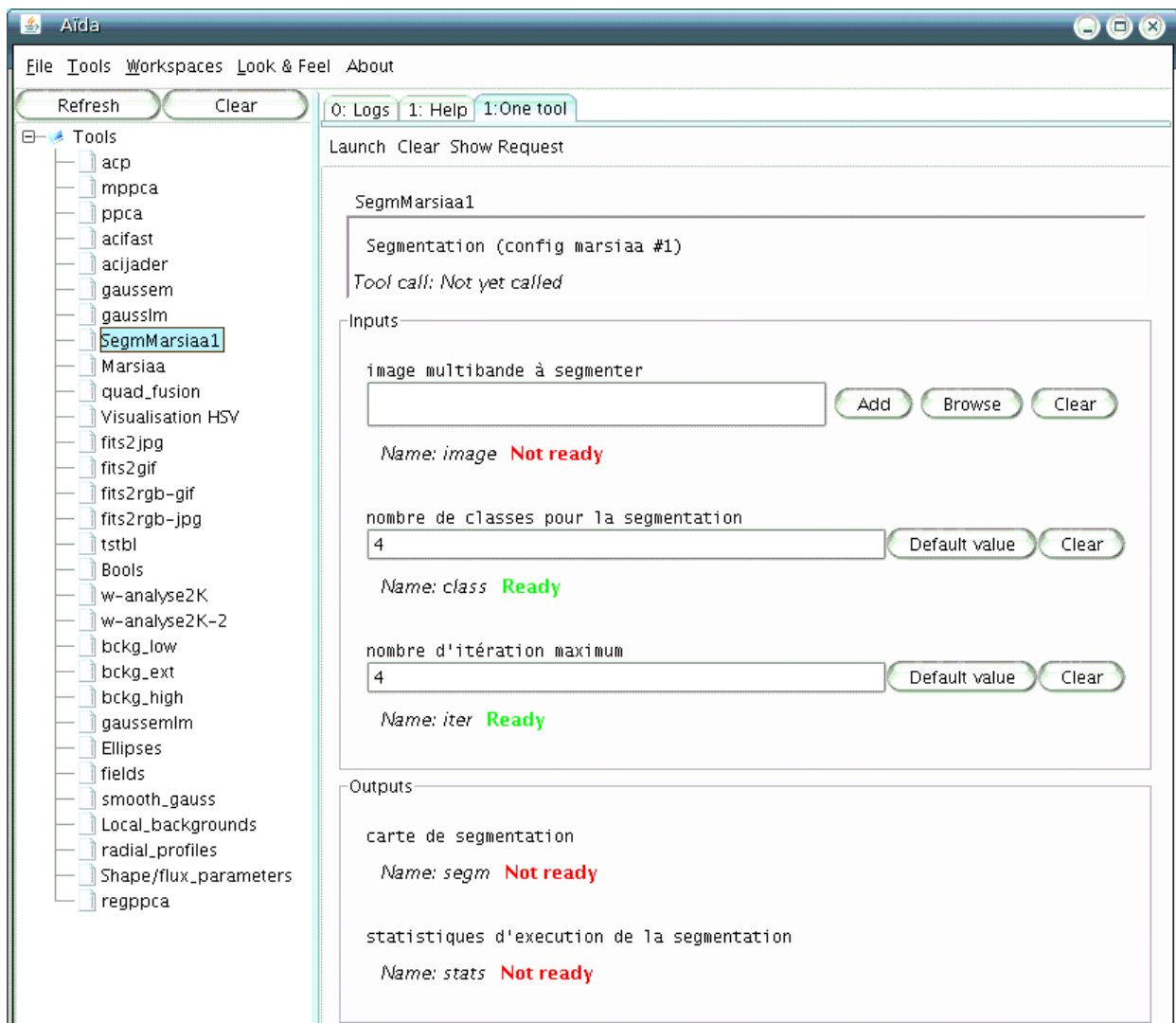


Figure 1.6 : Interface graphique, cette partie permet d'utiliser un outil. Elle propose donc différents champs à remplir en fonction du type de l'entrée.

1 : GXL (Graph eXchange Language) est un langage dérivé de XML, c'est un format standard d'échange de graphe.

4. Cahier des charges

L'interface utilisateur ainsi que le moteur d'exécution de l'architecture AĪDA sont écrits en Java. Les développements relatifs à l'évolution de l'interface sont donc dans le même langage de programmation.

4.1. Analyse de l'application

Une analyse de l'application doit être effectuée pour mieux cerner son fonctionnement, ainsi que ses mécanismes. Elle s'orientera vers une étude des différentes interactions de l'architecture d'une part, et vers une analyse du code de l'interface d'autre part.

4.2. Nettoyage des sources

Cette partie doit permettre une amélioration de la lisibilité des développements existants pour optimiser la maintenance ultérieure, ainsi que les mises à jours potentielles :

- L'aspect graphique doit être dissocié de la partie traitement des données.
- Les différents objets associés aux concepts tels que les tâches, les ports ou encore les données doivent être redéfinis.

4.3. Mise en place de JLOW 2

La mise en place de la nouvelle version de JLOW constitue la partie principale du projet. En effet, le but du projet consiste à faire évoluer AĪDA vers les nouvelles fonctionnalités de JLOW et permettre également que les futures mises à jour de la librairie se fassent de manière plus simple. Ainsi, toute la partie librairie servant à l'affichage des tâches doit être mise à jour, mais également le moteur d'exécution intégré dans l'interface graphique.

4.4. Ajout de fonctionnalités

L'interface graphique doit garder le même aspect graphique ainsi que le même mode de fonctionnement. De plus, elle doit intégrer de nouvelles fonctionnalités sur le même principe que celles déjà présentes afin que les utilisateurs ne soient pas désorientés.

5. Description de la mise en oeuvre

5.1. Installation de l'environnement de travail

Tout d'abord, j'ai mis en place l'architecture AïDA sur mon poste. En effet, selon le protocole d'installation, pour la partie serveur, il faut installer les différents outils disponibles avec leurs fichiers de description dans le répertoire *Home* de l'utilisateur AïDA. Ensuite, il faut mettre en place les scripts d'accès CGI, ainsi qu'un répertoire *exec* avec les droits nécessaires à l'exécution des outils.

Pour la partie cliente, j'ai compilé les sources de l'interface graphique puis il a fallu mettre à jour les fichiers de configuration.

Une fois l'architecture fonctionnelle, j'ai mis en place mon environnement de développement. Ainsi, j'ai installé Eclipse pour le développement et CVS pour la gestion des différentes versions de sources.

Enfin, j'ai testé le fonctionnement de l'application tout en approfondissant mes connaissances en Java. Ainsi, j'ai réalisé de petites applications qui m'ont permis de mieux cerner les mécanismes de l'interface utilisateur mais également d'améliorer mon approche du langage utilisé.

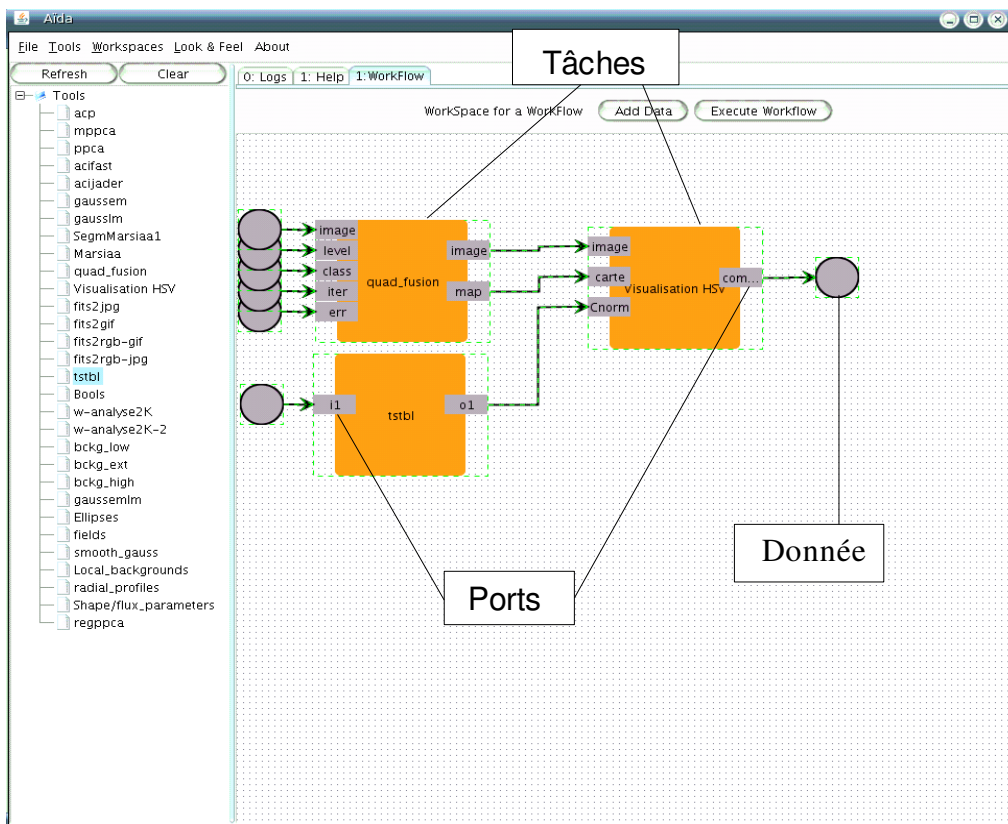


Figure 1.7 : Interface graphique, vue de la partie workflow originale.

5.2. Analyse de l'application

a) Étude des interactions d'AÏDA

Afin de bien comprendre le fonctionnement d'AÏDA, nous avons commencé par analyser les différentes interactions entre tous les modules. En effet, nous devons déterminer quels étaient les paramètres d'entrée et les résultats des outils par rapport aux scripts CGI, de ces scripts par rapport à l'interface utilisateur et enfin un point important dans le cadre de ce stage, les interactions entre l'interface utilisateur et le serveur de workflow JLOW.

Il en découle que :

- pour les interactions entre les outils et les scripts CGI, un outil doit être accompagné d'un descripteur qui regroupe les données nécessaires pour décrire l'outil, ses entrées-sorties et ses paramètres.
- pour les interactions entre les scripts CGI et l'interface utilisateur, l'emplacement des scripts CGI doit être renseigné dans les fichiers de configuration de l'interface. Ensuite, chaque script retourne un résultat, par exemple la liste des outils disponibles.
- pour les interactions entre l'interface utilisateur et JLOW, les outils et leurs descripteurs sont récupérés et enregistrés au lancement de l'interface. Ils sont utilisés dans la partie workflow et dans la partie outil seule. Les outils sont donc définis de façon standard par l'interface et les fonctions d'accès à ces derniers sont identiques dans les deux parties concernées.

La partie graphique du workflow est gérée par JLOW, l'affichage des tâches, des ports, des données ainsi que la gestion du chaînage dépendent donc de JLOW alors que l'attribution des valeurs aux données et aux ports est gérée par un module spécifique de l'interface.

b) Étude du diagramme de classes de l'interface

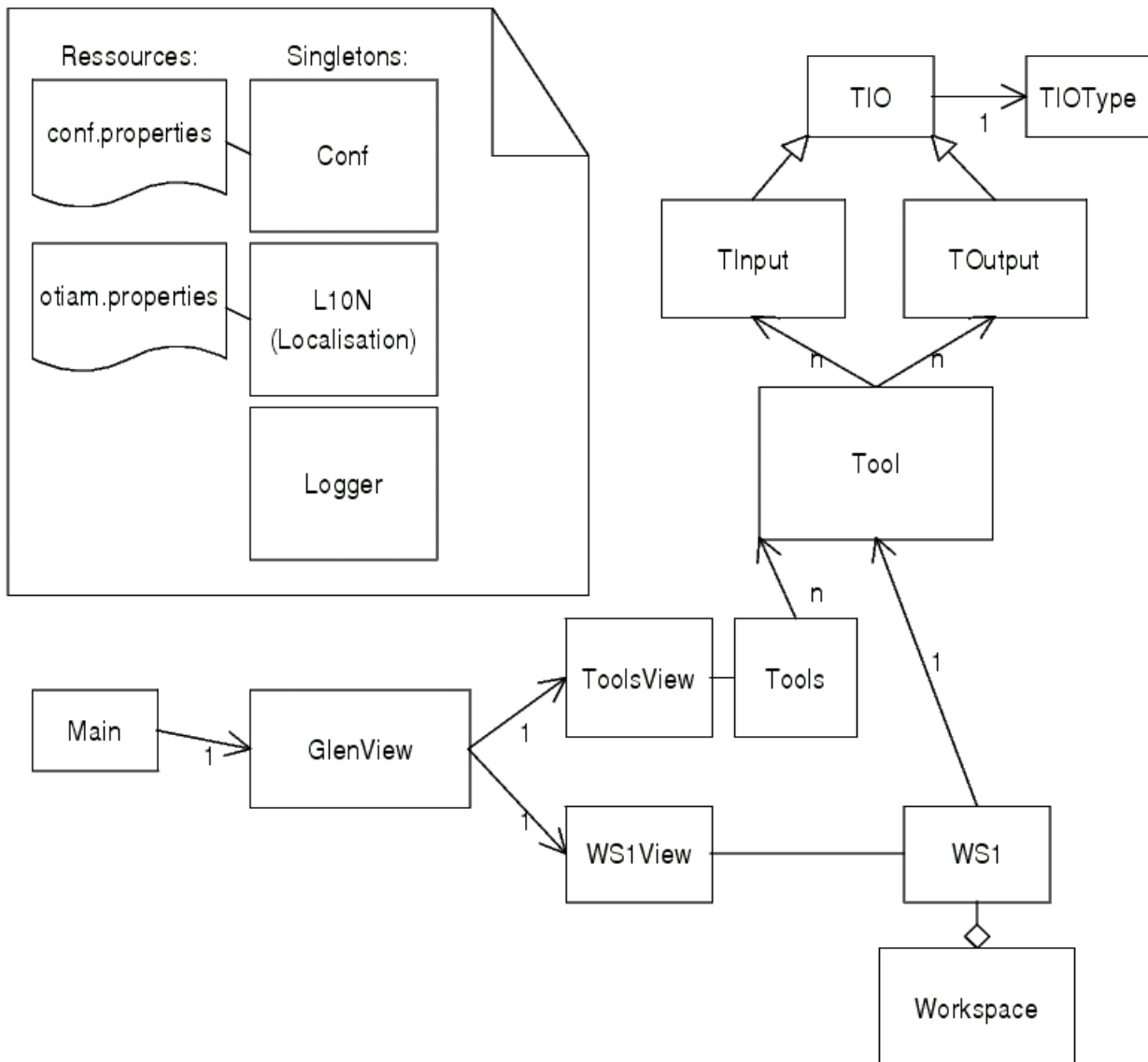


Figure 1.8 : Diagramme des classes principales de l'interface.

Les ressources permettent à *Conf*, *L10N* et *Logger* respectivement d'enregistrer les différentes informations nécessaires à la connexion au serveur, d'enregistrer les noms des boutons et les libellés relatifs à l'apparence, et de gérer les fichiers de *Logs*. Leurs conceptions reposent sur des singletons car une seule instance de chaque classe est nécessaire, voire indispensable, pour un fonctionnement optimal.

Concernant les particularités de ce diagramme :

La classe *Workspace* est une classe abstraite qui définit ce qu'un workspace doit contenir comme variables et comme méthodes. Deux classes en héritent, la classe *WS1* mais également la classe *WSWF* qui gèrent respectivement l'espace de travail pour un outil et l'espace de travail pour un workflow. La classe *Workspace* propose la méthode d'ajout d'un outil et toute la gestion des *Observers*. En effet, le fonctionnement de l'interface repose sur des *Observers*, chaque ajout d'outil déclenche ainsi une action sur l'espace de travail concerné.

La classe *Tool* est liée à deux classes qui gèrent soit les ports d'entrées soit les ports de sorties. Ces derniers héritent de la classe *TIO* qui gère les fonctions de base d'un port.

Le diagramme représenté dans la figure 1.8 montre les mécanismes de fonctionnement concernant l'accès à un outil. Le mécanisme général est identique pour la partie workflow. Au démarrage de l'interface, la classe *GlenView* met en place l'aspect graphique et appelle la classe *ToolsView*. Cette dernière gère l'affichage de la liste des outils. La classe *Tools* récupère la liste des outils disponibles sur le serveur, l'appel CGI est effectué dans cette classe. L'enregistrement des caractéristiques de chaque outil (nom, commentaires, catégorie) est effectué par la classe *Tool* alors que l'enregistrement des informations concernant les ports est effectué dans la classe *TInput* si cela concerne un port d'entrée ou *Toutput* si c'est un port de sortie.

Un espace de travail pour accéder à un outil est créé et géré par la classe *WS1*. En effet, cette dernière gère l'ajout ou la suppression d'un outil sur l'espace de travail concerné. Ensuite, la gestion de l'outil sélectionné est effectuée par la classe *Tool*.

5.3. Mise en place de JLOW 2

a) Intégration directe

- La démarche consiste à supprimer les sources de l'ancienne version de JLOW puis à nettoyer ceux de l'interface, et enfin, à rajouter la dernière version de la librairie en corrigeant les erreurs générées.
- Beaucoup d'erreurs ont été solutionnées sans difficulté et le nettoyage des sources de l'interface a consisté à dissocier le traitement des données de leur affichage.

Mais la conception fondamentalement différente de l'interface d'AIDA et de JLOW, a entraîné une redondance dans certains mécanismes et des défaillances dans d'autres. Ainsi le concept de donnée qui était incontournable dans AIDA est désormais une fonction secondaire dans JLOW 2. En effet, un port contient une valeur intrinsèque et le concept de donnée sert uniquement lorsque plusieurs ports ont la même valeur d'entrée.

- La conception d'AIDA impose un traitement et, par conséquent, une définition similaires des données et des ports ce qui est incompatible avec la nouvelle version de JLOW. En effet, cette dernière propose sa propre définition des ports et des données ainsi que leurs méthodes d'accès et de traitements.

Ainsi, mon analyse préliminaire ne m'a pas permis de détecter cette incompatibilité, probablement à cause d'un manque d'approfondissement. J'ai donc été contraint de reprendre mon projet plus en amont. Néanmoins cela m'a permis de mieux cerner les différents mécanismes utilisés ainsi que les besoins de la future application.

b) Élaboration d'une nouvelle interface

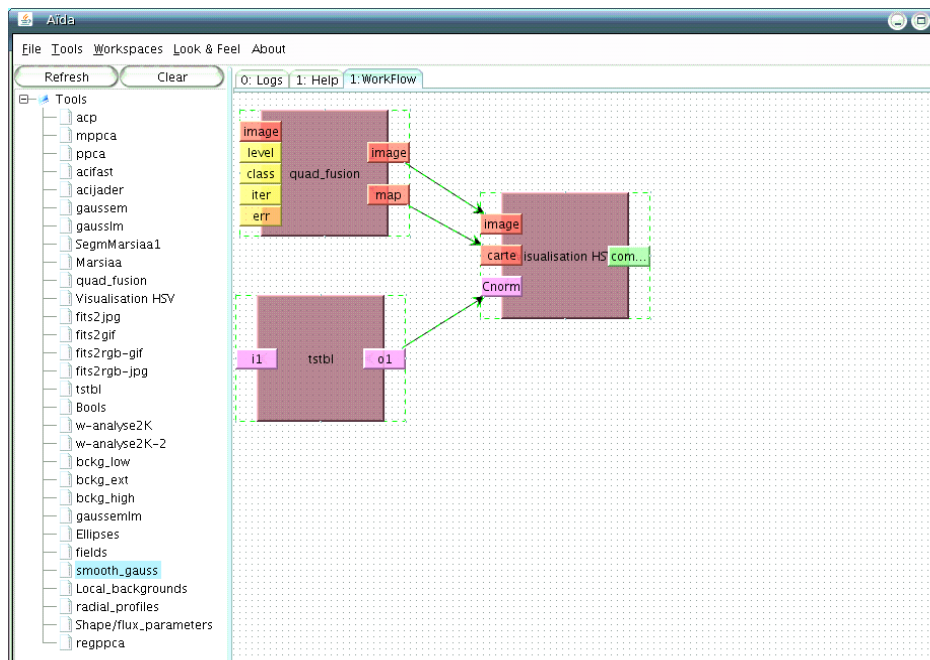


Figure 1.9 : Nouvelle Interface – workflow

- L'approche de cette élaboration est totalement différente. En effet, le but est de repartir sur des bases saines en prenant en compte les différents mécanismes fonctionnels de l'interface tel que l'aspect graphique et toutes les fonctionnalités et les outils de JLOW 2.
- En premier lieu, j'ai développé de petites applications basées sur les nouvelles fonctions de JLOW, ce qui m'a apporté une meilleure compréhension du fonctionnement de la librairie.

Puis, pour créer la nouvelle interface, nous avons créé la liste dynamique des outils. Ce mécanisme consiste en l'appel du script CGI qui retourne la liste des descripteurs des différents outils disponibles. Ensuite, il faut *parser* cette liste. En effet, chaque outil est accompagné d'un fichier descripteur qui contient différentes informations, ces informations étant enregistrées selon un certain format. Ainsi, le fait de *parser* consiste à récupérer les informations que l'on souhaite en fonction du format défini et de les enregistrer.

Ensuite, nous avons revu le principe d'utilisation d'un outil. En effet, lors de sa conception, cette partie avait fait, l'objet d'une étude approfondie et elle ne pouvait pas être remise en cause dans son intégralité. Le fonctionnement général tout comme le fonctionnement du point de vue de l'utilisateur reste identique, malgré quelques modifications de fond.

Nous avons totalement remanié la partie workflow, et implémenté la gestion de nouvelles informations pour les tâches. Le mécanisme de glisser/déposer a été repensé et s'appuie désormais sur une fonction proposée par JLOW 2.

De plus, une couleur est attribuée à chaque type de port, facilitant ainsi la création d'une chaîne. La sauvegarde et le chargement d'un workflow sont maintenant possibles.

L'utilisation a été améliorée et standardisée. Toutes les actions sont désormais disponibles via le menu contextuel. De plus, le renseignement des valeurs de chaque port se fait désormais via une interface permettant de renseigner toutes les valeurs d'une tâche.

Enfin, le serveur de workflow embarqué fait actuellement l'objet d'une mise à jour. J'ai commencé l'analyse de ce dernier, il apparaît que son remaniement est nécessaire car JLOW 2 propose désormais plus de fonctionnalités pour sa gestion.

- Le problème majeur fut la synchronisation des mécanismes d'appel et d'utilisation des outils entre l'accès à un outil et l'accès par workflow. En effet, le mécanisme principal pour un outil a été repris et a dû être adapté à la nouvelle définition des outils.

6. Bilan

6.1. Bilan du stage

Tout d'abord, on peut constater que le stage est positif, en effet, l'interface graphique de l'architecture AIDA est mise à jour et fonctionnelle.

De surcroît, ce projet a apporté des améliorations, d'une part pour l'utilisation de l'interface, notamment grâce aux nouvelles fonctionnalités intégrées. D'autre part, la maintenance de l'application est facilitée ().

6.2. Bilan humain personnel

Ensuite, mon bilan humain est également positif dans la mesure où ce stage m'a permis de rencontrer de nouvelles personnes.

De plus, j'ai découvert un nouvel environnement de travail qui m'a beaucoup plu, j'ai également découvert le domaine de l'astronomie que j'ai trouvé très intéressant.

6.3. Bilan technique personnel

Enfin, d'un point de vue technique, mon stage m'a énormément apporté. En effet, il m'a permis d'aborder un projet conséquent, et ainsi j'ai pu mettre en application mes connaissances du langage Java, mais surtout j'ai pu les approfondir.

En outre, le développement sur un tel projet dans le cadre d'un laboratoire a été très enrichissant. Mais cela rompt avec l'environnement scolaire et peut être déstabilisant au premier abord.

7. Conclusion

En conclusion, mon stage s'est déroulé au Centre de Données astronomique de Strasbourg, où j'ai été entouré de personnes attentives à mon projet et qui m'ont soutenu quand ce fut nécessaire.

Cette insertion dans la vie active a été pleine d'enseignements, j'ai pu découvrir un environnement de travail jusqu'alors inconnu. Le projet que j'ai réalisé m'a également beaucoup appris. En effet, je me suis rendu compte que l'immersion dans le code source d'autres personnes n'est pas facile et demande beaucoup d'investissement. Mais finalement, j'ai beaucoup appris pendant ce stage et je suis très content d'avoir côtoyé le monde de l'astronomie.

La prochaine évolution du projet est de le rendre facilement disponible en créant un package qui permette une diffusion et une installation simplifiées. J'apporterai les évolutions nécessaires pendant ma prolongation de stage.

Mots-Clés

Java, XML, workflow, interface graphique, traitement d'images.

Références

Site Internet de l'Observatoire astronomique de Strasbourg :

<http://astro.u-strasbg.fr/observatoire/>

Site Internet du CDS : <http://cdsweb.u-strasbg.fr/>

Site Internet du projet MDA : <http://cdsweb.u-strasbg.fr/MDA/mda.html>

Site Internet de JGraph, librairie graphique : www.jgraph.com/jgraph.html

Poster du groupe de travail Workflow :

http://www.france-ov.org/twiki/pub/GROUPEStravail/Workflow/SF2A_ASOV_WF_finale.pdf

Articles du groupe de travail Workflow :

<http://www.france-ov.org/twiki/pub/GROUPEStravail/Workflow/IAU2006louysetal.pdf>

http://www.france-ov.org/twiki/pub/GROUPEStravail/Workflow/iau-prague_slezak.pdf

Ce rapport est consacré à l'évolution d'un outil graphique de gestion de chaînes de traitement dans le cadre de l'architecture AIDA (Astronomical Image processing Distributed Architecture). Ce projet a été réalisé dans le cadre d'un stage au Centre de Données astronomique de Strasbourg, laboratoire de l'Observatoire astronomique.

On pourra y trouver une description détaillée du laboratoire et du sujet, ainsi que les différentes phases d'analyse, de conception et de mise en oeuvre du projet.