

## Rapport de stage

Description de blocs de traitements dans les workflows  
appliqués au traitement d'images astronomiques

Pour l'obtention du

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

Lieu du stage : **Centre de Données Astronomiques de Strasbourg**

Réalisé par :

BENJELLOUN TOUIMI OMAR.

Composition du jury:

- M. Jean-Pierre Jacquot
- M. Pascal Urso Professeur responsable
- M. André Schaaff Tuteur du stage

**A.U. 2007 - 2008**



## Remerciements

A travers ce travail, je tiens à remercier tous ceux qui ont orienté ses différentes étapes jusqu'à son terme, en particulier à Monsieur André SCHAAFF mon tuteur de stage pour ses précieux conseils, sa disponibilité et son soutien durant toute la période du stage.

Merci à Mireille LOUYS, François BONNAREL, Cyril PESTEL et Brice GASSMANN pour leur attention, leur compréhension et leur patience.

Mes remerciements et ma gratitude vont également à mon encadrant Universitaire M. Pascal URSO pour ses conseils et sa disponibilité et à l'ensemble du corps professoral du Master 2 Informatique qui a contribué à ma formation.

## Avant propos

L'analyse d'images en astronomie met en oeuvre des chaînes de traitements, qui en fonction de la complexité des nouvelles données disponibles se complexifient également.

Les procédures de traitement peuvent être modélisées au travers de workflows, exécutés soit sur des architectures de Calcul locales (e.g. clusters), soit sur des architectures distribuées de type 'Grille'. Pour ce faire, les workflows doivent être décrits précisément, au niveau des blocs de traitement d'une part et au niveau des données qu'ils consomment et produisent d'autre part. Cela revient à définir des méta-données décrivant les blocs, l'enchaînement des traitements et les données traitées.

L'un des outils d'**AÏDA** (Astronomical Image processing Distribution Architecture) permet de dessiner graphiquement un enchaînement de blocs, de générer une description **XML** de ces blocs et de soumettre le tout à un moteur d'exécution de workflows. Afin d'optimiser au mieux l'utilisation de cet outil, une nouvelle couche a été introduite pour le contrôle des blocs du workflow avant qu'il ne soit exécuté, la portée de cette solution étant partielle, l'un des objectifs du **CDS** (Centre de Données de Astronomiques de Strasbourg) est de la généraliser pour la totalité du workflow.

Dans ce cadre, le sujet de stage que m'a proposé le **CDS** avait pour but d'élargir l'étendue de cette couche aux différents blocs du workflow et de prévoir également des évolutions de certains éléments de l'architecture **AÏDA** (portage du serveur d'AÏDA sous **TOMCAT** et sous **Windows**).

L'analyse de ces besoins m'a conduit d'une part à concevoir et à réaliser un prototype d'une solution prenant en charge le contrôle de la totalité du workflow et d'autre part à définir une procédure de portage du serveur d'**AÏDA** sous **TOMCAT**, ainsi que sous **Windows**.

## Sommaire

<b>REMERCIEMENTS</b> .....	<b>3</b>
<b>AVANT PROPOS</b> .....	<b>4</b>
<b>SOMMAIRE</b> .....	<b>5</b>
<b>INTRODUCTION</b> .....	<b>8</b>
<b>I. PRESENTATION DU CDS</b> .....	<b>9</b>
A. STRUCTURE JURIDIQUE.....	9
B. PRESENTATION GENERALE.....	9
1. <i>Hautes énergies</i> .....	9
2. <i>Étoiles et systèmes stellaires</i> .....	10
3. <i>Galaxies</i> .....	10
4. <i>Le centre de données (CDS)</i> .....	10
C. ORGANISATION DU PERSONNEL.....	10
D. CENTRE DE DONNÉES ASTRONOMIQUE DE STRASBOURG (CDS).....	10
E. LES SERVICES DU CDS.....	12
1. <i>Simbad</i> .....	12
2. <i>VizieR</i> .....	13
3. <i>Aladin</i> .....	13
F. INTERNATIONAL VIRTUAL OBSERVATORY ALLIANCE (IVOA).....	14
<b>II. SITUATION AU DEBUT DU STAGE</b> .....	<b>16</b>
A. AÏDA.....	16
1. <i>Définition de la notion de workflow</i> .....	16
2. <i>Objectifs et fonctionnalités</i> .....	16
3. <i>Description générale</i> .....	17
a. <i>Liste des outils</i> .....	17
b. <i>« One tool »</i> .....	17
c. <i>Workflow</i> .....	18
d. <i>Serveur AÏDA</i> .....	19
e. <i>Serveur de workflow</i> .....	19
f. <i>Fichier descripteur</i> .....	20
4. <i>Couche de contrôle</i> .....	20
a. <i>Fichier Characterisation</i> .....	21
b. <i>Fichier de contraintes</i> .....	22
c. <i>Validation d'un outil</i> .....	23
d. <i>Validation d'un workflow</i> .....	24
B. OBJECTIFS ASSIGNÉS.....	25
<b>III. ENVIRONNEMENT DE STAGE</b> .....	<b>26</b>
A. ENVIRONNEMENT DE DEVELOPPEMENT.....	26
B. ORGANISATION ET DEROULEMENT DU STAGE.....	26
<b>IV. TRAVAIL EFFECTUE</b> .....	<b>28</b>
A. GENERATION DE LA CHARACTERISATION XML D'UN FICHIER FITS.....	28
1. <i>Besoin</i> .....	28
2. <i>Recherche de solutions</i> .....	28
3. <i>Approche générale de la solution</i> .....	28

a.	<i>L'entrée FITS</i> .....	29
b.	<i>La sortie XML</i> .....	30
c.	<i>Comparaison Entrée FITS/ Sortie XML</i> .....	31
4.	<i>Solution adoptée</i> .....	32
a.	<i>Characterisation Utype</i> .....	32
b.	<i>Architecture de la solution</i> .....	33
5.	<i>Réalisation</i> .....	34
a.	<i>Phase 1: génération du fichier Uty</i> .....	34
i)	Création du fichier de mapping.....	34
ii)	Extraction des « mots clés » de l'entête FITS .....	35
iii)	Traitement des valeurs extraites.....	36
iv)	Le mapping mots clés/Utypes .....	36
v)	Segmentation des utypes en axes.....	36
vi)	Création du fichier Uty.....	36
b.	<i>Phase 2: génération du fichier XML</i> .....	37
6.	<i>Illustration</i> .....	37
7.	<i>Evaluation de la solution</i> .....	39
a.	<i>Qualités</i> .....	39
b.	<i>Critiques</i> .....	39
B.	INTEGRATION DE FITS TO CHARACTERISATION DANS AIDA ET VALIDATION DU WORKFLOW .....	40
1.	<i>Intégration de la librairie au niveau des outil du workflow</i> .....	40
a.	<i>Construction d'un fichier de mapping générique</i> .....	40
b.	<i>Génération du fichier Characterisation (en sortie des outils)</i> .....	40
2.	<i>Validation d'un workflow</i> .....	40
3.	<i>Critiques de l'intégration</i> .....	41
C.	OUTIL INTERACTIF DE MAPPING .....	42
1.	<i>Besoin</i> .....	42
2.	<i>Solution</i> .....	42
3.	<i>Spécifications de l'outil</i> .....	42
4.	<i>Conception de l'outil</i> .....	42
a.	<i>Diagramme des cas d'utilisation</i> .....	43
b.	<i>Description de quelques cas d'utilisation</i> .....	44
c.	<i>La base de connaissance</i> .....	44
i)	Choix d'implémentation.....	45
ii)	Schéma de la base de connaissances.....	45
iii)	Alimentation de la base.....	46
5.	<i>Description de l'outil de mapping</i> .....	46
a.	<i>Vue d'ensemble</i> .....	46
b.	<i>Quelques éléments de menu</i> .....	47
i)	Load From.....	47
ii)	Save As .....	48
iii)	CharacTools.....	48
D.	PORTAGE DU SERVEUR AIDA .....	50
1.	<i>Besoin</i> .....	50
2.	<i>Analyse de la situation</i> .....	50
3.	<i>Démarche du portage</i> .....	50
a.	<i>Portage sous TOMCAT</i> .....	50
b.	<i>Portage sous Windows</i> .....	51
4.	<i>Serveur AIDA après le portage</i> .....	51
	<b>PERSPECTIVES</b> .....	<b>52</b>
	<b>BILAN</b> .....	<b>53</b>

---

<b>CONCLUSION</b> .....	54
<b>BIBLIOGRAPHIE</b> .....	55
<b>TABLE DES FIGURES</b> .....	56
<b>ANNEXES</b> .....	57
ANNEXE n° 01 : DESCRIPTION DE LA LIBRAIRIE FITS TO CHARACTERISATION .....	57
ANNEXE n° 02 : QUELQUES MANIPULATIONS DE LA LIBRAIRIE NOM.TAM.FITS .....	57
ANNEXE n° 03 : UTILISATION DE LA LIBRAIRIE CHARACTERISATION LIBRARY .....	58
ANNEXE n° 04 : ILLUSTRATION DE LA GENERATION DU FICHIER CHARACTERISATION. ....	59
ANNEXE n° 05 : TUTORIAL PORTAGE DU SERVEUR AIDA SOUS TOMCAT .....	66
ANNEXE n° 06 : SCHEMA DU MODELE DE LA CHARACTERISATION XML.....	69
ANNEXE n° 07 : LISTE DES UTYPES POUR LE MODELE DE LA CHARACTERISATION VERSION 1.11 .....	77

## Introduction

Ce rapport a pour objectif de rendre compte de la première partie de mon stage de fin d'étude du Master 2 Informatique de l'Université Henri Poincaré Nancy1. Celui-ci s'est déroulé au Centre de Données astronomiques de Strasbourg (C.D.S.) à partir du 11 février pour une durée de 5 mois.

Je débute ce rapport par une présentation du centre de recherche dans lequel j'ai travaillé ensuite je donne une description de l'existant et des principaux objectifs assignés. Après avoir détaillée la manière dont nous nous sommes organisés, je présente, étape par étape, le travail que j'ai effectué. Enfin je dresse un bilan général de ce stage.

## I. Présentation du CDS

### A. 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).

### B. 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 ».
- Licence en Sciences et autres licences (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 :

#### 1. 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.

## **2. 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.

## **3. Galaxies**

Les activités de l'équipe sont centrées sur les problèmes de la structure du Groupe Local, de ses populations stellaires et sur la dynamique gravitationnelle. De plus, l'équipe possède un savoir-faire 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.

## **4. 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.

### ***C. 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 du CDS comprend environ une douzaine de chercheurs, 12 ingénieurs, 3 techniciens, et plusieurs collaborateurs à contrat temporaire et invités.

### ***D. Centre de Données astronomique de Strasbourg (CDS)***

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

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

- rassembler toutes 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 « HautesÉnergies » de l'Observatoire de Strasbourg.

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

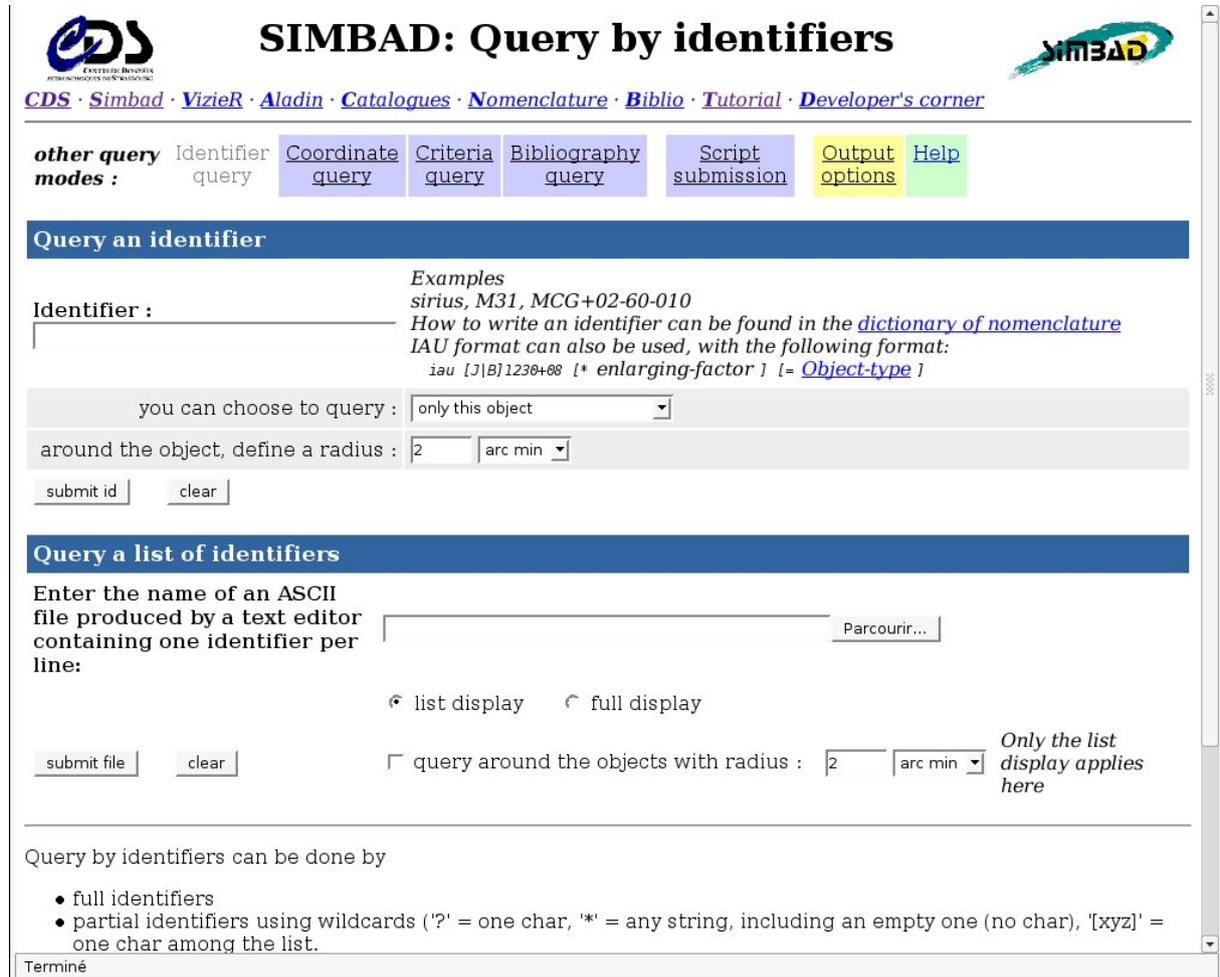
- NASA,
- National Astronomical Observatory (Tokyo, Japon),
- l'Académie des Sciences de Russie,
- le réseau PPARC Starlink au RoyaumeUni,
- l'Observatoire de Beijing (Chine),
- l'Université de Porto Alegre au Brésil,
- l'Université de La Plata en Argentine,
- InterUniversity Center for Astronomy and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

Le CDS 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 : en particulier le CDS abrite 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. Le CDS abrite aussi les copies miroirs Européennes des journaux de l'American Astronomical Society (AAS).

## E. Les Services du CDS

### 1. Simbad



**CDS** **SIMBAD: Query by identifiers** 

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

other query Identifier Coordinate Criteria Bibliography Script Output Help  
modes : query query query query submission options

**Query an identifier**

Identifier :  *Examples: sirius, M31, MCG+02-60-010*  
*How to write an identifier can be found in the [dictionary of nomenclature](#)*  
*IAU format can also be used, with the following format:*  
*iau [J|B]1230+08 [\* enlarging-factor ] [= Object-type ]*

you can choose to query :

around the object, define a radius :

**Query a list of identifiers**

Enter the name of an ASCII file produced by a text editor containing one identifier per line:

list display  full display

query around the objects with radius :   *Only the list display applies here*

Query by identifiers can be done by

- full identifiers
- partial identifiers using wildcards ('?' = one char, '\*' = any string, including an empty one (no char), '[xyz]' = one char among the list).

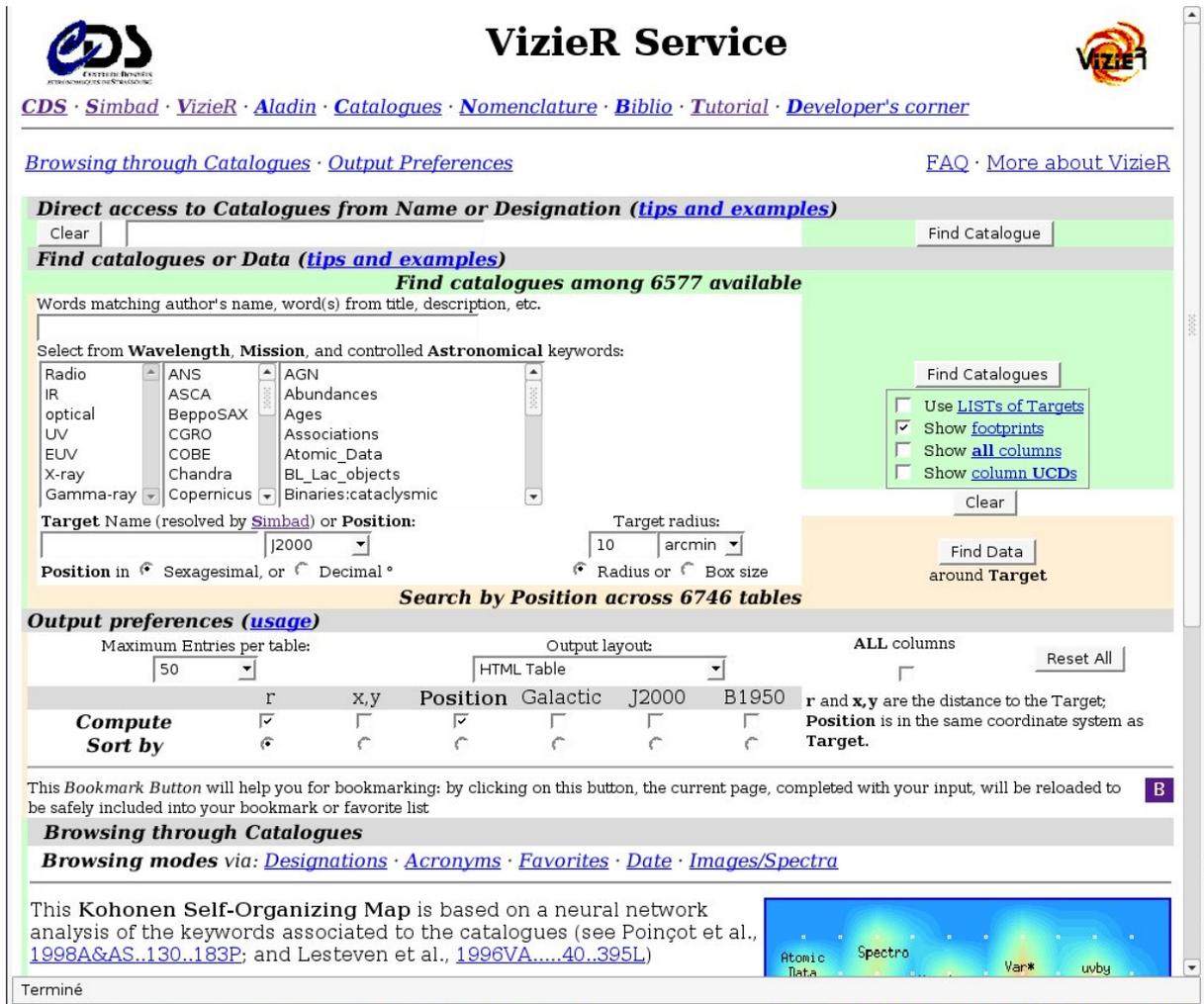
Terminé

**Figure 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 millions d'identificateurs pour plus de 3 millions d'objets différents. Pour chaque objet figurent dans la base quelques mesures (position, magnitude dans différents domaines de longueurs d'ondes), ainsi que les références bibliographiques où l'objet est cité (plus de 110 000 articles sont concernés). L'utilisateur peut choisir le format du fichier où seront entreposés les résultats de la requête (Fig. 1). En effet, 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.

## 2. VizieR



**VizieR Service**

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

Browsing through Catalogues · Output Preferences FAQ · More about VizieR

**Direct access to Catalogues from Name or Designation (tips and examples)**

Clear  Find Catalogue

**Find catalogues or Data (tips and examples)**

**Find catalogues among 6577 available**

Words matching author's name, word(s) from title, description, etc.

Select from **Wavelength**, **Mission**, and controlled **Astronomical** keywords:

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

Find Catalogues

Use [LISTs of Targets](#)

Show [footprints](#)

Show [all columns](#)

Show [column UCDS](#)

Clear

Target Name (resolved by [Simbad](#)) or **Position**:  J2000

Target radius:  10 arcmin

Position in  Sexagesimal, or  Decimal  Radius or  Box size

Find Data around Target

**Search by Position across 6746 tables**

**Output preferences (usage)**

Maximum Entries per table:  50

Output layout:  HTML Table

ALL columns

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

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

This **Bookmark Button** will help you for bookmarking: by clicking on this button, the current page, completed with your input, will be reloaded to be safely included into your bookmark or favorite list

**Browsing through Catalogues**

**Browsing modes** via: [Designations](#) · [Acronyms](#) · [Favorites](#) · [Date](#) · [Images/Spectra](#)

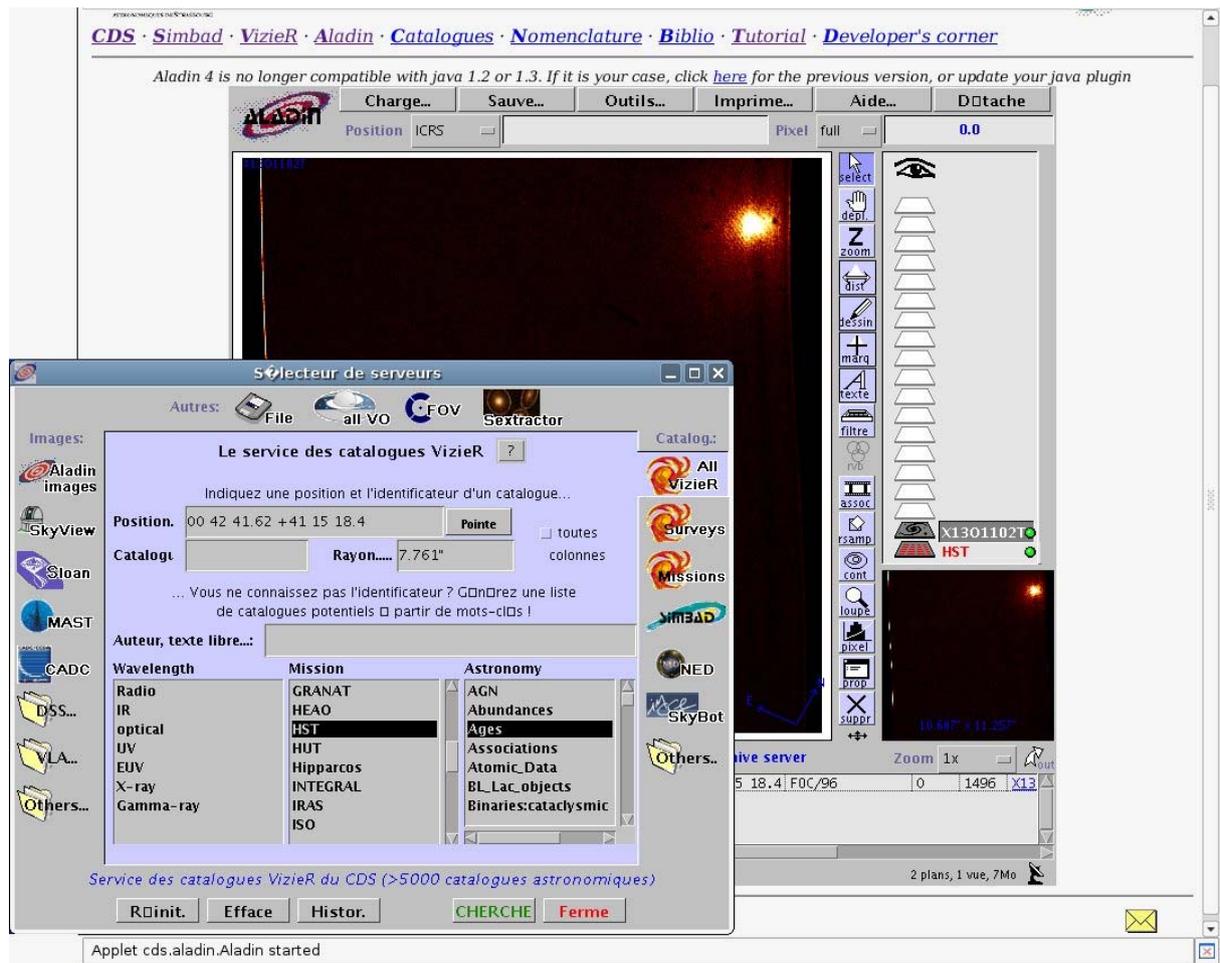
This **Kohonen Self-Organizing Map** is based on a neural network analysis of the keywords associated to the catalogues (see Poinçot et al., [1998A&AS..130..183P](#); and Lesteven et al., [1996VA.....40..395L](#))

Terminé

**Figure 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 (SGBD) relationnel. 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. 2). Il faut noter que les très grands catalogues (plus de 107 enregistrements) ne peuvent pas être gérés par un SGBD relationnel pour des raisons de performances. Des outils spécifiques doivent être utilisés.

## 3. Aladin



***Figure 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, . . .).

## ***F. International Virtual Observatory Alliance (IVOA)***

Jusqu'à une époque récente, un grand nombre de projets ne prévoyaient pas de rendre accessibles les données issues des missions (spatiales, terrestres). Depuis l'avènement des nouvelles technologies, Internet et les réseaux rapides, d'importants efforts ont permis de mettre à disposition des astronomes de nombreuses sources de données (on peut citer par exemple les services VizieR, Simbad et Aladin du CDS). Depuis quelques années, l'interopérabilité entre ces services est devenu une priorité car elle permet un accès aisé à l'information, un croisement des données de toutes natures et elle conduit au concept d'Observatoire Virtuel.

L'astronome disposera à terme de nouveaux instruments utiles à sa recherche au travers d'une simple interface Web. Afin d'y parvenir, de nombreuses collaborations sont indispensables afin d'aboutir aux consensus indispensables à la pose des bases nécessaires à la "construction" de l'Observatoire Virtuel. Les projets nationaux (OVF pour la France, GAVO pour l'Allemagne, le NVO pour les Etats-Unis, ...) et transnationaux (ESA VO, ESO VO, ...) sont réunis au sein de l'International Virtual Observatory Alliance. Ils participent à l'élaboration de recommandations dans divers domaines (Data Model, Accès aux données, Sémantique, Grilles, etc.) au travers de groupes de travail qui se réunissent semestriellement. A titre d'exemple significatif, la première recommandation, "VOTable", décrit la formalisation de tables de données astronomiques au format XML. Celle-ci a été adoptée par de nombreux fournisseurs de données et intégrée dans les outils de l'OV.

## II. Situation au début du stage

### A. AÏDA

#### 1. Définition de la notion de workflow

On appelle « **workflow** » (d'après Wikipédia) la modélisation et la gestion informatique de l'ensemble des tâches à accomplir et des différents acteurs impliqués dans la réalisation d'un processus métier (aussi appelé processus opérationnel ou bien procédure d'entreprise). Le terme de « **workflow** » pourrait donc être traduit en français par « gestion électronique des processus métier ». De façon plus pratique, le workflow décrit le circuit de validation, les tâches à accomplir entre les différents acteurs d'un processus, les délais, les modes de validation, et fournit à chacun des acteurs les informations nécessaires pour la réalisation de sa tâche.

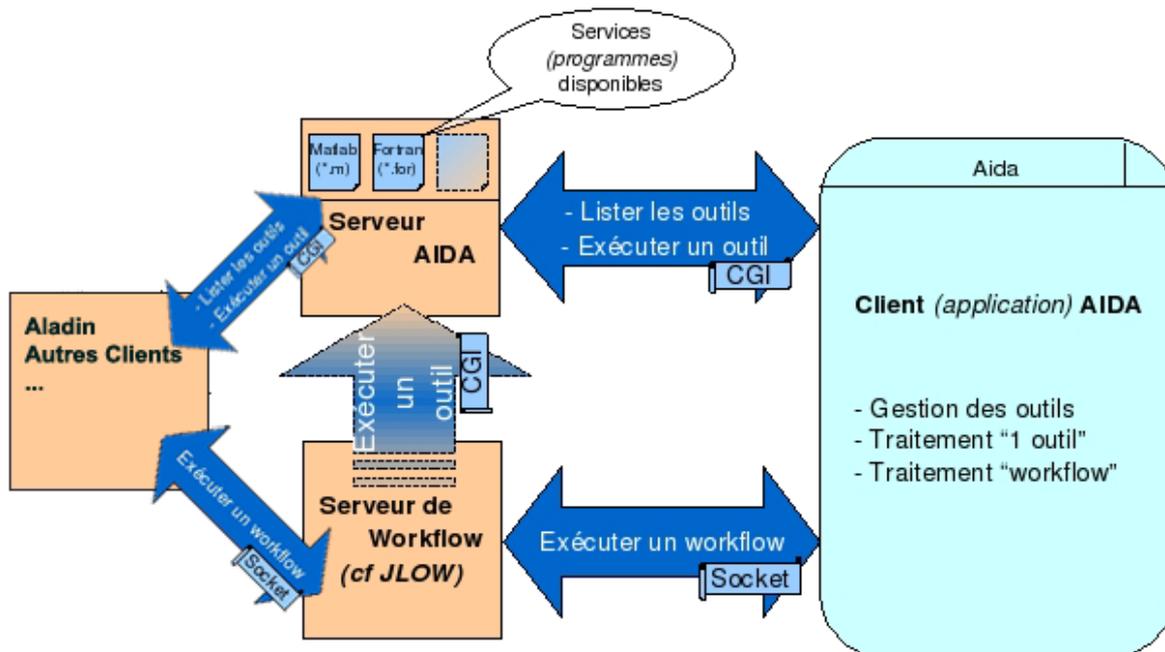
Le moteur de workflow est le dispositif logiciel permettant d'exécuter une ou plusieurs définitions de workflow. Par abus de langage, on peut appeler ce dispositif logiciel tout simplement "workflow".

#### 2. Objectifs et fonctionnalités

En astronomie, il est fréquent de manipuler des images afin de révéler un certain nombre d'informations. Il existe beaucoup de programmes de traitement d'images différents et qui ne sont pas tous écrits dans le même langage. Il est fréquent que seuls les auteurs et les utilisateurs avertis sachent comment les utiliser correctement. En outre, l'accès à ces outils est souvent difficile en raison de leur éparpillement dans la communauté scientifique.

Le développement d'**AÏDA** (Astronomical Image processing Distribution Architecture) a été initié dans le cadre de l'**ACI Masses de Données** et se poursuit dans le cadre du projet européen **VOTECH**. Elle regroupe un ensemble de programmes de traitement d'images afin de les rendre accessibles et utilisables facilement par un grand nombre de personnes, autant dans la communauté du traitement d'images que dans la communauté astronomique. En plus de la possibilité d'exécuter chaque outil individuellement, AÏDA permet de chaîner l'exécution de plusieurs outils, au sein de ce que l'on appelle un workflow.

### 3. Description générale



**Figure 4: Fonctionnement général d'AIDA.**

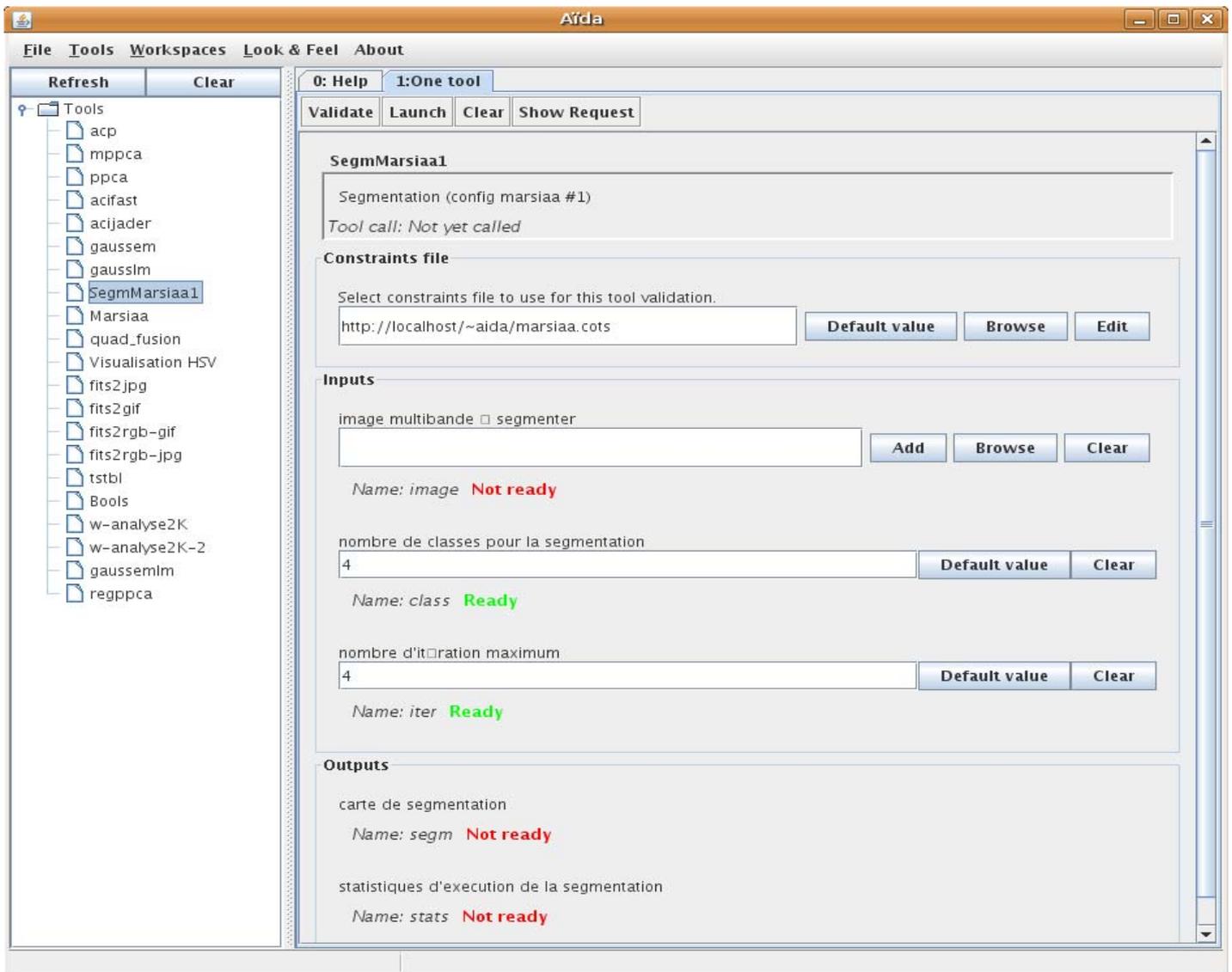
AIDA est une architecture constituée d'une partie serveur, qui contient les outils et leur documentation (fichier descripteur de l'outil), et d'une partie cliente permettant aux utilisateurs d'utiliser les différents outils (cf. **Figure 4**). Cette dernière est divisée graphiquement en deux parties (cf. **Figure 5**). La première est une liste qui permet à l'utilisateur de sélectionner le ou les outils qu'il souhaite utiliser. La deuxième est un ensemble d'onglets correspondants soit à un outil, soit à un workflow.

#### a. Liste des outils

Chaque outil du client **AIDA** correspond à un programme de traitement d'images. Au démarrage, le client récupère la liste des outils et leurs documentations respectives depuis le serveur **AIDA**.

#### b. « One tool »

L'espace de travail « one tool » (cf **Figure 5**) permet d'avoir une brève description de l'outil utilisé (nom et commentaire), de saisir la valeur des différentes entrées et d'afficher la valeur des sorties une fois l'exécution terminée.



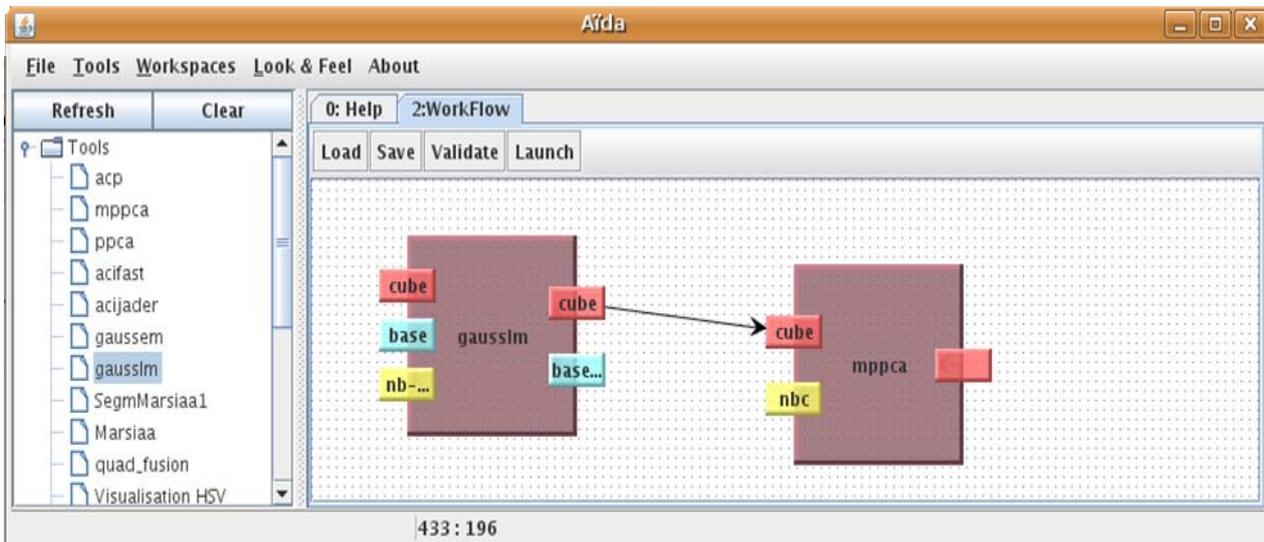
**Figure 5: Application cliente: liste des outils (g) et un outil (d).**

Lorsque l'utilisateur demande l'exécution d'un outil, le client communique la demande ainsi que toutes les informations relatives à l'outil à exécuter. L'exécution se fait au niveau serveur, soit en local, soit en utilisant des ressources de calcul comme les clusters. Une fois l'exécution terminée, le serveur retourne un fichier contenant les détails de l'exécution, dont en particulier, les sorties (en général des images).

### c. Workflow

La partie cliente d'AÏDA permet de dessiner graphiquement un workflow (cf Fig. 6): chaque outil est représenté par une boîte et le chaînage des outils par des flèches. Comme pour l'espace de travail « one

tool », il est possible de saisir les entrées de chaque outil du workflow grâce un menu contextuel.



**Figure 6: Application cliente: workflow.**

Pour exécuter un workflow, il faut d'abord le communiquer au serveur de workflow. Ensuite, le serveur analyse le workflow, détermine l'ordre d'exécution des outils et lance chacun d'entre eux de la même manière que si on ne traitait qu'un seul outil (cf. ci-dessus).

#### **d. Serveur AÏDA**

Le serveur AÏDA héberge essentiellement les outils ou programmes de traitement d'images ainsi que leur documentation. Ce serveur est distant. Pour cette raison, son accès se fait par CGI. De cette manière, on restreint l'accès au serveur à des demandes d'exécution et à la réception des sorties (standard ou d'erreur) de programmes.

#### **e. Serveur de workflow**

Comme nous l'avons vu précédemment, l'interprétation du workflow (qui n'inclut pas l'exécution en elle-même des outils) est déléguée au serveur de workflow (cf. figure 4). Contrairement au serveur AÏDA, il peut être distant ou local. Par conséquent, le choix de la communication avec celui-ci s'est porté sur les Sockets permettant aussi bien une communication locale que distante.

## f. Fichier descripteur

Pour être intégré dans AÏDA, chaque outil doit être documenté par un fichier descripteur. C'est un fichier texte au format 'clef=valeur'. Lors de la lecture de la liste des outils, les informations du descripteur sont enregistrées avec l'outil lui correspondant.

Voici un exemple de fichier descripteur:

<i>// Description générale de l'outil:</i>	
tool.label=SegmMarsiaa1	Nom de l'outil
tool.comment=Segmentation (config marsiaa #1)	Commentaire
<i>// Description des entrées:</i>	
input.1.type=fits:seq	Entrée n1 Type (image fits)
input.1.label=image	Nom de l'entrée
input.1.comment=image multibande à segmenter	Commentaire
	Entrée n2
input.2.type=numeric:int	
input.2.label=class	
input.2.comment=nombre de classes pour la segmentation	
input.2.default=4	Entrée n3
input.3.type=numeric:int	
input.3.label=iter	
input.3.comment=nombre d'itération maximum	
input.3.default=4	
<i>// Description des sorties:</i>	
output.1.type=fits	Sortie n1 Type
output.1.label=segm	Nom de la sortie
output.1.comment=carte de segmentation	Commentaire
	Sortie n2
output.2.type=data:text	
output.2.label=stats	
output.2.comment=statistiques d'exécution de la segmentation	
<i>// Chemin pour accéder à l'exécutable:</i>	
tool.pathname=/local/aida/lib/OtiamTools/segm1/wrap	

**Figure 7: Exemple de fichier descripteur.**

## 4. Couche de contrôle

L'exécution de la plupart des outils d'AÏDA produit une ou plusieurs images en sortie à partir de celles fournies en entrée. Cependant, il n'est

pas certain que ce résultat soit correct. Certains outils requièrent que les images données présentent certaines caractéristiques pour que le traitement soit correctement appliqué.

Pour vérifier la cohérence entre les images fournies en entrée et celles que l'outil attend, une nouvelle couche a été récemment introduite, elle permet d'effectuer ce contrôle avant que le traitement ne soit effectué.

Le mécanisme de contrôle repose sur deux fichiers : le fichier « **Characterisation** » associé à chaque image fournie en entrée et le **fichier de contraintes** relatif à l'outil.

### a. Fichier Characterisation

Lorsqu'un astronome exploite une image il a besoin de connaître un minimum d'informations la concernant. Ces informations précisent la façon dont cette image a été obtenue (source, domaine spectral, calibrage, lieu et date d'observation, etc.), mais aussi une description de son contenu (résolution, échantillonnage, niveau de détail de l'image, etc.). Le format **FITS** (Flexible Image Transport System) est très souvent utilisé dans ce domaine. Outre l'image, ce format contient dans, son entête, un certain nombre d'informations la concernant. Cet entête est une suite de champs 'clef=valeur'. Il est donc lisible par un humain (caractères ASCII).

Cependant ce format ne permet pas une représentation structurée et standardisée de l'information à cause de l'utilisation arbitraire des clefs.

**Characterisation** est un modèle **XML** de données, créé à l'Observatoire de Strasbourg dans le cadre de l'**IVOA** (International Virtual Observatory Alliance), permettant, tout comme **FITS**, de décrire le contenu d'une image astronomique mais d'une manière plus précise, structurée et plus complète que **FITS**. Contrairement à ce dernier, **Characterisation** ne peut pas être décrite dans un entête. Il est nécessaire d'avoir un fichier à part.

**Characterisation** a deux finalités possibles, documentaire et applicative.

Documentaire, car il contient suffisamment d'information sur des images pour pouvoir estimer leur correspondance à des contraintes très précises émises par une base de données ou des services, dans le but de sélectionner un échantillon fiable d'observations.

Et applicative, car la précision et la complétude des informations fournies par **Characterisation** permettent d'effectuer des traitements sur ces images. C'est dans ce cadre que sera utilisé **Characterisation** dans ce stage.

Ce modèle peut présenter ses méta-données de deux manières: en axes ou en propriétés. En effet, une image peut être décrite d'une

manière spatiale, temporelle, spectrale ou de flux. Ces quatre axes sont fixes dans le modèle Characterisation. Mais bien entendu, l'utilisateur est libre d'en définir d'autres.

Chaque axe se décrit ensuite en propriétés (cf. Fig. 8). Il y a tout d'abord des propriétés d'ordre général (unité, état de la calibration, marge d'erreur, etc.) et des propriétés plus spécialisées.

Ci-dessous est représenté la structure générale d'un fichier **Characterisation** au format **XML**.

```
<characterisation>
  <characterisationAxis>
    <axisName>...</axisName>
    <ucd>...</ucd>
    <unit>...</unit>
    <coordsystem id="..." type="..." href="..." />
    <independantAxis>...</independantAxis>
    ...
  <coverage>
    ...
  </coverage>
  <resolution>
    ...
  </resolution>
  ...
</characterisationAxis>
<characterisationAxis>
  ...
</characterisationAxis>
...
</characterisation>
```

**Figure 8 : Structure principale simple d'un fichier Characterisation**

Pour plus d'informations concernant ce modèle, veuillez vous reporter au document de référence situé à l'adresse suivante :

<http://www.ivoa.net/Documents/latest/CharacterisationDM.html>

## **b. Fichier de contraintes**

Afin de vérifier que les images possèdent bien les caractéristiques exigées par un outil, il faut les contrôler avant l'exécution de l'outil. C'est le rôle du fichier de contraintes. Ce fichier contient une liste de règles à

appliquer sur les caractéristiques de chaque image (fichier Characterisation) fournie en entrée.

Ci-dessous un extrait d'un fichier de contraintes :

```

AxisShortcut SPATIAL: Axis[ucd="pos"]
AxisShortcut FLUX: Axis[independantAxis="false"]

# 1. Vérifier que tous les Ik ont une résolution spatiale proche et exprimée dans la même
unité:
IF (EXISTS(:SPATIAL.Resolution))
  NEAR(:SPATIAL.Resolution.resolutionRefVal.period.C1, 0.3)
  NEAR(:SPATIAL.Resolution.resolutionRefVal.period.C2, 0.3)
  EQUAL(:SPATIAL.Resolution.unit) OR EQUAL(:SPATIAL.unit)
ELSE
  NEAR(:SPATIAL.SamplingPrecision.samplingPrecisionRefVal.samplingPeriod.C1,
0.3)
  NEAR(:SPATIAL.SamplingPrecision.samplingPrecisionRefVal.samplingPeriod.C2,
0.3)
  EQUAL(:SPATIAL.SamplingPrecision.unit) OR EQUAL(:SPATIAL.unit)
FI
...

```

**Figure 9 : Extrait d'un fichier de contraintes**

### c. Validation d'un outil

La **validation** d'un outil (vérification des contraintes que doivent satisfaire les images fournies en entrée) nécessite un fichier de contraintes et un fichier Characterisation pour chaque image que l'on donnerait en entrée.

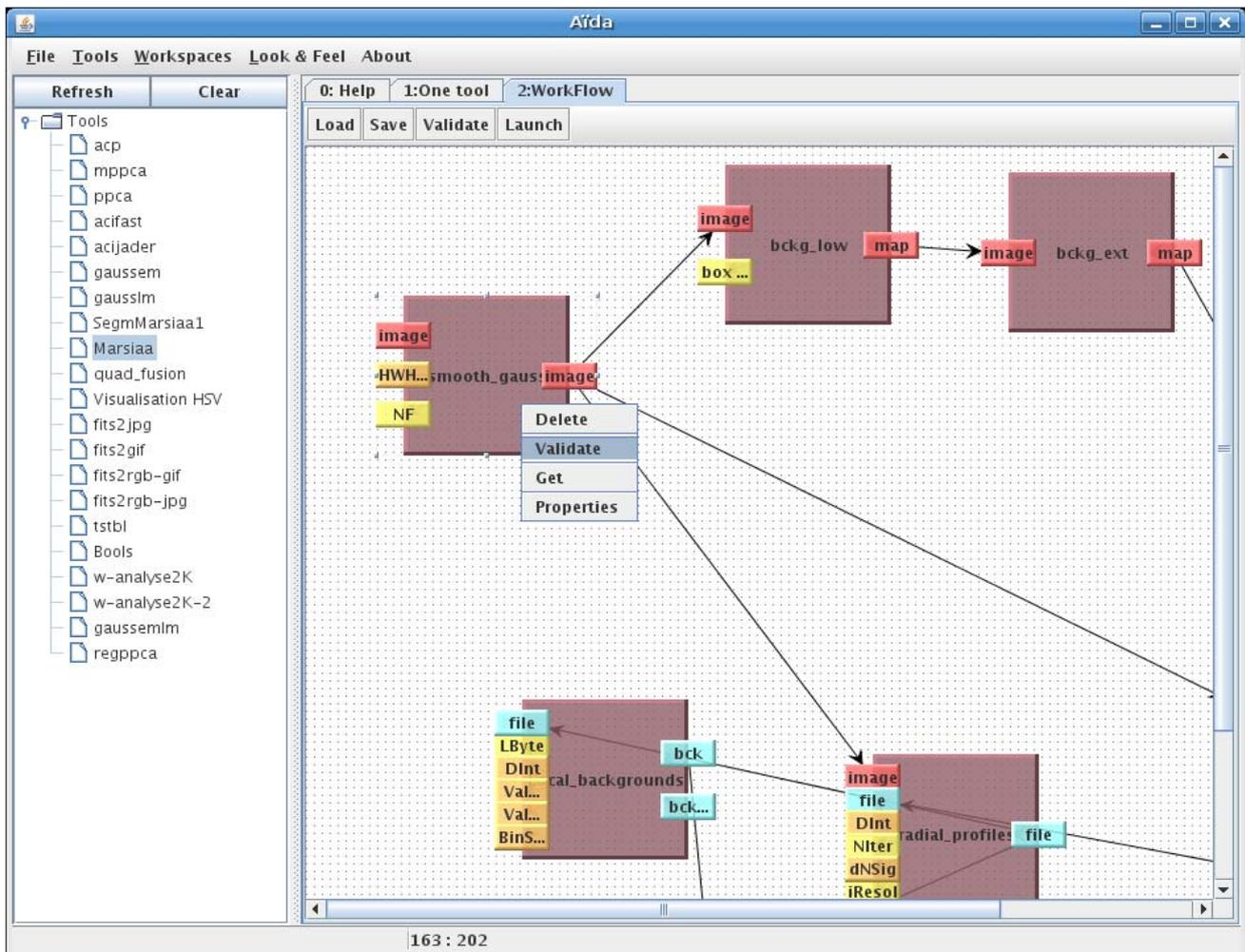


**Figure 10: Constitution d'un outil.**

Lors de la phase de validation, les contraintes non validées sont présentées à l'utilisateur afin qu'il puisse y remédier, soit en changeant l'image en question, soit en corrigeant le fichier Characterisation s'il est erroné, soit en rectifiant le fichier de contraintes.

#### d. Validation d'un workflow

Valider un workflow revient à valider chaque outil le constituant. On peut ainsi renforcer le bon fonctionnement d'un workflow, car si tous les outils reçoivent des images correspondant à leurs « **contraintes** », alors il y a de fortes chances que chaque outil produise le résultat attendu et qu' il en soit de même pour le workflow tout entier.



**Figure 11: espace de travail "Workflow", validation des contraintes**

Or, pour vérifier la cohérence d'un enchaînement de plusieurs outils, il faut impérativement que chaque outil produise également un fichier **Characterisation** pour chaque sortie de type image. Cependant aucun outil n'est actuellement capable de réaliser une telle tâche. Ainsi, il devient impossible de valider les outils dont une ou plusieurs entrées sont connectées à la ou les sorties d'un ou plusieurs autres outils, à cause de l'absence de fichier **Characterisation**.

## **B. OBJECTIFS ASSIGNÉS**

Le principal objectif de ce stage est d'améliorer la couche de contrôle d'**AÏDA** en étendant sa portée à l'intégralité du Workflow.

Pour cela, il faut répondre au problème du fichier **Characterisation** absent en sortie des outils et qui bloque le processus de validation lors de leur enchaînement.

Parallèlement à ce travail des évolutions de l'architecture d'**AÏDA** sont aussi envisagées, tel que le portage de son serveur à la fois sous **TOMCAT** et sous **WINDOWS**.

Le travail effectué durant ce stage doit prévoir une évolution souple du modèle de données **Characterisation** et de l'architecture **AÏDA**, afin d'éviter tout problème de compatibilité avec de futures versions. Il faut également tenir compte du fait qu'un utilisateur de la communauté scientifique doit pouvoir ajouter de nouveaux outils à l'application **AÏDA** sans trop de difficultés.

### III. Environnement de stage

Avant de commencer à présenter le travail effectué durant ce stage, je tiens à donner un bref aperçu de l'environnement de développement et de la manière dont nous nous sommes organisés afin de répondre au mieux au sujet donné.

#### A. Environnement de développement

Voici la configuration logicielle utilisée durant ce stage :

- OS : Ubuntu 7.04 et Windows XP.
- IDE : Eclipse.
- Langages : java (jdk 1.6), perl (5.8.8) et XML.
- Serveurs web : APACHE et TOMCAT.
- Atelier de génie Logiciel : Rational Rose
- Outils: ant,cvs.
- Documentation : Open Office.

#### B. Organisation et déroulement du stage

S'agissant d'un stage plutôt de recherche que de développement, il est délicat d'établir un cahier des charges précis et définitif. Par conséquent, les objectifs énoncés (en fin du chapitre précédant) se sont précisés au fur et à mesure du stage, grâce à des réunions régulières avec toutes les personnes impliquées dans ce projet.

Ces réunions se déroulaient ainsi: je présentais l'avancement de mon travail puis nous discutons des modifications éventuelles à apporter et enfin nous réfléchissions aux étapes suivantes.

A la fin de chaque réunion, j'en faisais un résumé sur le **Twiki**. Il s'agit d'un site communautaire du même genre que Wikipédia. Ce site est interne au **CDS**: seul les membres du **CDS** peuvent y accéder au moyen d'un nom d'utilisateur et d'un mot de passe. Une personne autorisée peut ainsi suivre l'évolution des projets du **CDS** et éventuellement ajouter ou modifier des articles sur le ou les projets auxquels elle participe. Ainsi, il est possible de suivre l'évolution des réunions et du travail effectué durant ce stage. Ce dernier m'a servi également comme moyen pour partager des versions de test avec les futurs utilisateurs et d'avoir ainsi un feedback permanent.

En dépit de l'absence d'un cahier des charges définitif, et de ce fait d'un planning précis, les premières réunions nous ont tout de même permis d'établir les principales étapes à suivre durant ce stage.

Avant de pouvoir plonger au coeur du sujet, il fallait d'abord me familiariser avec les outils que j'allais utiliser. Ainsi, pendant le premier mois, j'ai étudié le modèle Characterisation, le fonctionnement et le code source d'**AÏDA**. Le deuxième mois, je me suis un peu plus familiarisé avec l'architecture **AÏDA** en y apportant quelques modifications :

- Validation de toutes les tâches qui comportent des entrées avec fichier de Characterisation.
- Portage du serveur d'AÏDA sous Windows et sous TOMCAT

Ensuite, l'étape suivante consistait à concevoir (et à réaliser) un prototype pour générer le fichier Characterisation d'une façon automatique, cette phase constituant le noyau du stage et fut sa partie la plus longue soit environ 2 mois.

Enfin la dernière étape du stage (Post soutenance) aura pour but d'intégrer ce travail dans **AÏDA**.

Dans ce qui suit je présente le travail effectué durant ce stage en détaillant ces trois étapes.

## IV. Travail effectué

### A. Génération de la Characterisation XML d'un fichier FITS

#### 1. Besoin

La validation d'un outil nécessite un fichier de Contraintes et un fichier Characterisation pour chaque image fournie en entrée. Un workflow étant constitué d'un enchaînement d'outils, sa vérification impose que chacun d'entre eux soit en mesure de produire un fichier de Characterisation pour chaque sortie de type image afin d'assurer le processus de validation le long des différents outils le constituant et par conséquent sa validation totale.

Or, à l'heure actuelle aucun outil d'**AÏDA** n'est adapté à la réalisation d'une telle tâche, donc pour assurer la validation d'un workflow, il faut nécessairement prévoir un mécanisme qui réponde à ce besoin.

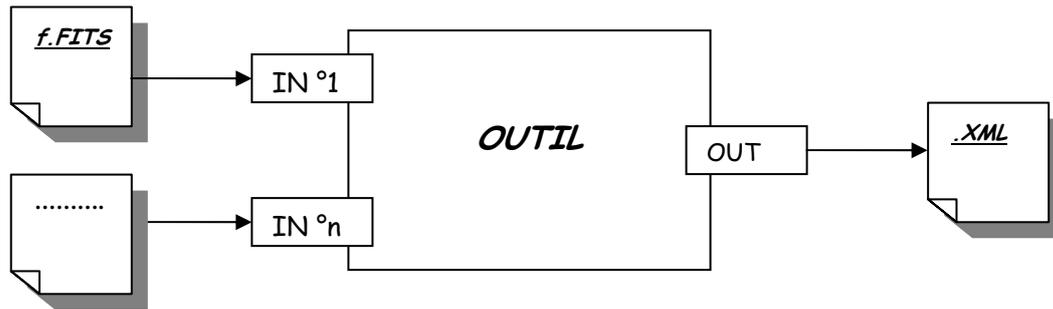
#### 2. Recherche de solutions

La première idée était d'ajouter cette fonctionnalité en interne au niveau de chaque outil, la génération du fichier Characterisation s'effectuant d'une manière transparente et ne nécessitant aucune autre interaction avec le concepteur du workflow, or cette solution n'est pas immédiatement réalisable, car cela nécessite une série de mise à jour des différents programmes de traitement d'images qui sont intégrés dans **AÏDA**. Etant donné le nombre important de ces programmes, il était difficile d'aller plus loin dans cette voie.

Nous avons pensé donc à une solution plus générique et applicable indépendamment de l'outil de traitement d'image utilisé. L'idée est de générer nous même le fichier Characterisation à partir de l'image produite en sortie de chaque outil. Pour cela, il faut qu'elle soit de type **FITS** afin d'exploiter son entête pour remplir le fichier **Characterisation**.

#### 3. Approche générale de la solution

La solution que nous proposons, consiste à réaliser un outil qui construit un fichier **Characterisation** à partir d'un fichier **FITS**, en effet ce dernier doit accepter comme entrée principale un fichier **FITS** (et éventuellement d'autres ressources auxiliaires) et produire en sortie le fichier de Characterisation correspondant au format **XML** (cf. Figure 12).



**Figure 12: Schéma générale de l'outil de génération**

L'outil représenté ci-dessus, est considéré pour l'instant comme une boîte noire qui reçoit des données et qui produit des résultats.

Afin de mieux décortiquer son contenu (et déterminer les traitements à réaliser), analysant ses éléments d'entrées et de sorties.

### **a. L'entrée FITS**

Cette entrée est destinée à recevoir des images au format **FITS** (Flexible Image Transport System).

Un avantage important de ce format est le stockage des méta-données de l'image dans un entête lisible par un humain, au format ASCII, de sorte qu'un utilisateur puisse examiner les entêtes d'un fichier de provenance quelconque . Chaque fichier **FITS** se compose d'un ou plusieurs **entêtes** contenant des "**card**" ASCII (80 caractères de longueur constante) qui portent des paires de "**keyWord / value**" (mots clés / valeurs), intercalées entre les blocs de données. Les paires de mots clés / valeurs fournissent des informations telles que la taille, l'origine, les coordonnées, le format de données binaire, les commentaires en format libre, l'historique des données, et toute autre chose souhaitée par le rédacteur ; tandis que beaucoup de mots-clés sont réservés pour l'usage interne, la norme permet l'utilisation arbitraire du reste (cf. Figure 13).

KeyWords	Values
AXIS1NAM	spatial
AXIS1UCD	pos
AXIS1UNI	deg
AXIS1CAL	CALIBRATED
AXIS1SYS	FK5
AXIS1STE	Unknown
AXIS1SYE	Unknown
AXIS1IND	TRUE
AXIS1BIN	512 1024
AXIS1UND	FALSE
AXIS1REG	TRUE
POSITIO1	11.8879898
POSITIO2	-25.337183770
LOWERBOX	11.952260421486-25.19496154463112
UPPERBOX	11.823719238514-25.47940599536888
SEEING	Unknown
PIXSCALE	-0.0002777777783923699 -0.00027777...
AXIS2NAM	time
AXIS2UCD	time
AXIS2UNI	day
AXIS2CAL	CALIBRATED
AXIS2SYS	TT-ICRS-WAVELENGTH-TOPO
AXIS2STE	Unknown
AXIS2SYE	Unknown
AXIS2IND	TRUE
AXIS2BIN	1
AXIS2UND	TRUE
AXIS2REG	FALSE
OBSTIME	51381.3422136574
TIMEMIN	Unknown
TIMEMAX	Unknown
TIMERES	Unknown
TIMESAMP	Unknown
AXIS3NAM	Spectral
AXIS3UCD	em
AXIS3UNI	mu

**Figure 13: un extrait de l'entête du fichier FITS : 011.8882-25.2883.fits**

## b. La sortie XML

La sortie de l'outil doit produire un fichier Characterisation au format **XML**, ce dernier, tout comme les entêtes **FITS**, permet une description du contenu d'une image astronomique mais d'une manière plus précise, structurée et complète, ainsi ce modèle hiérarchise les méta-données de l'image selon 2 niveaux de détails :

- En axes : spatial, temporel, spectral, de flux (axes spécifiques) ou axes génériques (définis par le rédacteur du fichier)
- En propriétés : chaque axe se décrit ensuite en propriétés (cf. Figure 8 et Figure 14).

```

<?xml version="1.0" encoding="UTF-8" ?>
- <char:characterisation xmlns:char="http://www.ivoa.net/xml/Characterisation/
- <char:characterisationAxis>
  <char:axisName>spatial</char:axisName>
  <char:ucd>pos</char:ucd>
  <char:unit>deg</char:unit>
  <char:independentAxis>true</char:independentAxis>
  <char:calibrationStatus>CALIBRATED</char:calibrationStatus>
- <char:numBins2>
  <char:I1>512</char:I1>
  <char:I2>1024</char:I2>
</char:numBins2>
  <char:undersamplingStatus>false</char:undersamplingStatus>
  <char:regularsamplingStatus>true</char:regularsamplingStatus>
- <char:coverage>
  - <char:location>
    - <char:coord>
      - <stc:Position2D xsi:type="stc:posVector2CoordinateType" xmlns:stc="ht
        v1.30.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instai
        <stc:Name1>lon</stc:Name1>
        <stc:Name2>lat</stc:Name2>
      - <stc:Value2 xsi:type="stc:double2Type">
        <stc:C1>11.8879898</stc:C1>
        <stc:C2>-25.33718377</stc:C2>
      </stc:Value2>
    </stc:Position2D>
    </char:coord>
  </char:location>
  - <char:bounds>
    - <char:limits>
      - <char:Coord2VecInterval xsi:type="stc:coord2VecIntervalType" xmlns:st
        v1.30.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instai
      - <stc:LoLimit2Vec>
        <stc:C1>11.952260421486</stc:C1>

```

**Figure 14: un extrait du fichier de Characterisation  
:011.8882-25.2883.xml**

### c. Comparaison Entrée FITS/ Sortie XML

Les deux figures 13 et 14 représentent une description des méta-données de la même image **011.8882-25.2883.fits**, la comparaison de leurs valeurs relève beaucoup de similitudes (cf. Figure:15), ainsi, ce constat nous a conduit à associer pour chaque image donnée en entrée de notre outil un fichier de **mapping** contenant les correspondances possibles entre les **mots clés** du fichier **FITS** et le modèle de **Characterisation**.

KeyWords	Values	
AXIS1NAM	spatial	<?xml version="1.0" encoding="UTF-8" ?>
AXIS1UCD	pos	- <char:characterisation xmlns:char="http://www.ivoa.net/xml/Charact
AXIS1UNI	deg	- <char:characterisationAxis>
AXIS1CAL	CALIBRATED	<char:axisName>spatial</char:axisName>
AXIS1SYS	FK5	<char:ucd>pos</char:ucd>
AXIS1STE	Unknown	<char:unit>deg</char:unit>
AXIS1SYE	Unknown	<char:independentAxis>true</char:independentAxis>
AXIS1IND	TRUE	<char:calibrationStatus>CALIBRATED</char:calibrationStatus>
AXIS1BIN	512 1024	- <char:numBins2>
AXIS1UND	FALSE	<char:I1>512</char:I1>
AXIS1REG	TRUE	<char:I2>1024</char:I2>
POSITIO1	11.8879898	</char:numBins2>
POSITIO2	-25.337183770	<char:undersamplingStatus>false</char:undersamplingStatus>
LOWERBOX	11.952260421486-25.19496154463112	<char:regularsamplingStatus>true</char:regularsamplingStatus>
UPPERBOX	11.823719238514-25.47940599538888	- <char:coverage>
SEEING	Unknown	- <char:location>
PIXSCALE	-0.000277777783923599-0.00027777	- <char:coord>
AXIS2NAM	time	- <stc:Position2D xsi:type="stc:posVector2CoordinateType" xm
AXIS2UCD	time	v1.30.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema
AXIS2UNI	day	<stc:Name1>lon</stc:Name1>
AXIS2CAL	CALIBRATED	<stc:Name2>lat</stc:Name2>
AXIS2SYS	TT-ICRS-WAVELENGTH-TOPO	- <stc:Value2 xsi:type="stc:double2Type">
AXIS2STE	Unknown	<stc:C1>11.8879898</stc:C1>
AXIS2SYE	Unknown	<stc:C2>-25.33718377</stc:C2>
AXIS2IND	TRUE	</stc:Value2>
AXIS2BIN	1	</stc:Position2D>
AXIS2UND	TRUE	</char:coord>
AXIS2REG	FALSE	</char:location>
OBSTIME	51381.3422136574	- <char:bounds>
TIMEMIN	Unknown	- <char:limits>
TIMEMAX	Unknown	- <char:Coord2VecInterval xsi:type="stc:coord2VecIntervalType
TIMERES	Unknown	v1.30.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema
TIMESAMP	Unknown	- <stc:LoLimit2Vec>
AXIS3NAM	Spectral	<stc:C1>11.952260421486</stc:C1>
AXIS3UCD	em	
AXIS3UNI	mu	

**Figure 15: correspondances keyWords/Characterisation du fichier FITS : 011.8882-25.2883.fits**

## 4. Solution adoptée

### a. Characterisation Utype

C'est un nouveau format de Characterisation qui a été récemment mis au point par l'Observatoire Astronomique de Strasbourg (équivalent à la Characterisation XML), il permet d'une façon équivalente au modèle XML de représenter les méta-données d'une image astronomique, mais sous un format textuel respectant la structure hiérarchique du schéma de la Characterisation XML (cf. Figure: 16). Afin de mentionner l'origine du modèle de données, chaque utype doit être préfixé par : « **CharacterisationAxis** » dans le cas d'un axe générique et « **AxisNameAxis** » pour un axe spécifique, où **AxisName** peut prendre les valeurs : Spatial, Time, Spectral ou Flux. Par exemple, le 1<sup>er</sup> niveau de la structure d'arbre d'un utype issu de l'axe spécifique « **Time** » serait : **TimeAxis**.

Pour plus de détails concernant ce modèle ainsi que la liste de tous les utypes possibles, veuillez consulter le document de référence situé en annexe 07 de ce document ou l'adresse suivante :

<http://www.ivoa.net/Documents/latest/UtypeListCharacterisationDM.html>

```
%CharacterisationAxis 1
%SpatialAxis.AxisName spatial
%SpatialAxis.independentaxis TRUE
%SpatialAxis.calibrationStatus CALIBRATED
%SpatialAxis.samplingPrecision.samplingPrecisionRefVal.samplingPeriod (-
0.000277777783923599,-0.000277777783923599)
%SpatialAxis.unit deg
%SpatialAxis.coverage.bounds.limits.Coord2VecInterval.LoLimit2Vec
(11.952260421486,-25.19496154463112)
%SpatialAxis.undersamplingStatus FALSE
%SpatialAxis.coordsystem FK5
%SpatialAxis.ucd pos
%SpatialAxis.numBins (512,1024)
%SpatialAxis.regularsamplingStatus TRUE
%SpatialAxis.coverage.bounds.limits.Coord2VecInterval.HiLimit2Vec
(11.823719238514,-25.47940599536888)
%SpatialAxis.coverage.location.coord.Position2D.Value2.C1 11.8879898
%SpatialAxis.coverage.location.coord.Position2D.Value2.C2 -25.337183770

%CharacterisationAxis 2
%TimeAxis.AxisName time
%TimeAxis.undersamplingStatus TRUE
%TimeAxis.coordsystem TT-ICRS-WAVELENGTH-TOPO
.....

%CharacterisationAxis n
.....
```

**Figure 16: un extrait du fichier de Characterisation en utypes:  
011.8882-25.2883.uty**

## b. Architecture de la solution

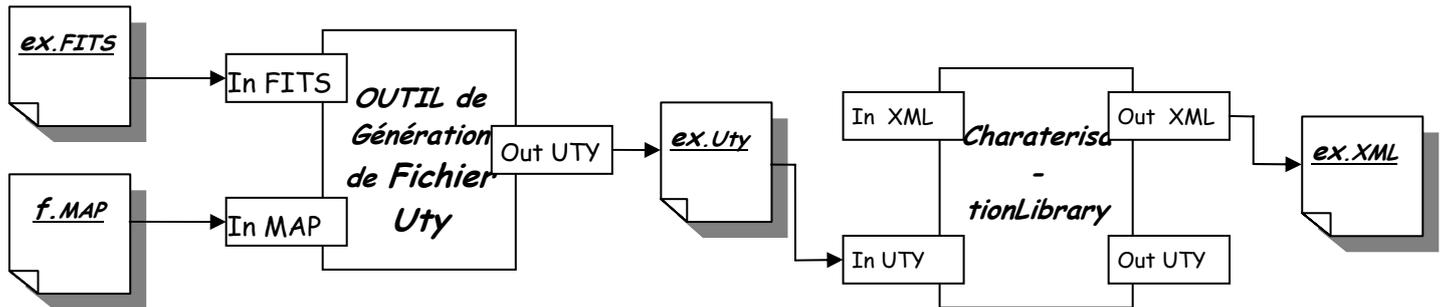
Comme le modèle de Characterisation en **utypes** est équivalent à celui en **XML**, une librairie JAVA a été développée au **CDS (CharacterisationLibrary)** pour assurer une conversion entre les deux modèles.

Characterisation XML  $\leftrightarrow$  Characterisation en utypes.

La simplicité et la maniabilité de la Characterisation en utypes nous ont conduit à adopter ce modèle dans notre solution, le processus de

génération du fichier Characterisation au format XML passe par deux phases :

- A partir d'un fichier **FITS** et d'un fichier de **mapping** (contenant les correspondances possibles entre les keyWords du fichier **FITS** et les **utypes**), on génère le fichier Characterisation en **utypes** correspondant (format **.Uty**).
- Le fichier ainsi obtenu en étape une sert comme une entrée à la librairie **CharacterisationLibrary**, cette dernière se chargeant de la conversion au format **XML**.



**Figure 17: Schéma de la solution adoptée**

(Dans ce qui suit, je détaille la réalisation de chacune de ces étapes en précisant à chaque fois leurs prémisses et leurs contraintes.)

## 5. Réalisation

Pour la mise en œuvre de notre solution, j'ai développé une librairie de classes JAVA : «**FitsToCharacterisation**» qui permet entre autres d'offrir des routines efficaces assurant les deux phases de la génération. Pour plus d'informations concernant cette librairie ainsi que la manière de s'en servir veuillez consulter l'annexe n°: 01.

### a. Phase 1: génération du fichier Uty

Le processus de génération du fichier Uty passe par plusieurs étapes :

#### i) Création du fichier de mapping

Le fichier de mapping fournit à notre outil est de format texte, chacune de ses lignes commence par un « **mot clé** » (de l'entête **FITS**) suivie du séparateur « + » et du « **utype** » correspondant (cf. Figure:18).

La portée de ce fichier se restreint uniquement au fichier **FITS** à partir duquel l'extraction des « **mots clés** » est effectuée. Néanmoins, on peut avoir un fichier commun à une famille d'images **FITS** dont les « **mots clés** » utilisés sont identiques et décrivent les mêmes propriétés.

```

AXIS1NAM + SpatialAxis.AxisName
AXIS1UCD + SpatialAxis.ucd
AXIS1UNI + SpatialAxis.unit
AXIS1CAL + SpatialAxis.calibrationStatus
.....

```

**Figure 18: Extrait de fichier de mapping : 2MASS.map**

Le fichier de mapping à fournir à notre outil est à rédiger (pour l'instant) manuellement dans un éditeur de texte et porte l'extension « .map ».

## ii) Extraction des « mots clés » de l'entête FITS

### *Choix de la librairie*

Pour la manipulation des données **FITS**, il existe plusieurs outils dans différents langages de programmation, entre autres on trouve trois bibliothèques développées en JAVA : tam.aaa.fits, Eap.fits et Jfits.

Ci-dessous un tableau comparatif des principaux services supportés par chacune :

Library	Images	Groups	ASCII Table	Binary Table	Var. Len. Arrays
nom.tam.fits	rw	rw	rw	rw	rw
eap.fits	rw	-	rw	rw	rw
jfits	rw	-	rw	rw	-

**Figure 19: Comparatif de quelques bibliothèques implémentant FITS**

**r** : prise en charge de l'accès en lecture

**w** : prise en charge de l'accès en écriture

- : non supporté

**Images**: support de l'accès à l'image FITS

**Groups** : support des groupes de données aléatoires

**ASCII Tables**: support des tables ASCII.

**Binary Tables**: support des tables binaires.

**Variable Length Arrays**: support de la variable length array au niveau des tables binaires.

On remarque que l'accès en lecture à l'entête **FITS** est une fonctionnalité de base supportée par toutes les bibliothèques citées, ce qui répond bien à nos besoins en terme d'extraction des données. Or, la plupart des outils développés au **CDS** se servent de la bibliothèque **nom.tam.fits**, donc par souci de compactibilité, notre choix s'est porté sur cette dernière.

(Pour plus d'informations concernant l'extraction des données de l'entête **FITS** via cette bibliothèque veuillez consulter l'annexe n°: 02.)

### iii) Traitement des valeurs extraites

Afin d'optimiser le fichier Uty à générer, nous procédons à un filtrage qui permet d'exclure les « **mots clés** » ayant des valeurs non significatives, (comme par exemple : null et Unknown), ensuite les valeurs qui passent ce filtre subissent un éventuel formatage selon la spécification d'écriture des valeurs utypes (voir le document de référence de la Characterisation utype en annexe n° :07).

### iv) Le mapping mots clés/Utypes

Une fois ces 3 étapes achevées on procède au mapping **mots clés /Utypes**, on obtient ainsi une représentation de la forme (**Utype, Valeur**).

### v) Segmentation des utypes en axes

A cette étape deux niveaux de segmentation sont à réaliser, le 1<sup>er</sup> niveau a comme but de ranger pour chaque axe les utypes correspondants et le second permet une séparation selon le type d'axe (spécifique ou générique).

#### *Techniques de la segmentation*

D'après ce que j'ai appris à propos de la représentation des « **mots clés** », dans un entête **FITS** les clefs d'un même axe sont rangées séquentiellement en commençant par la clef qui correspond à son nom. Cette information permet de répondre à notre premier besoin. Pour distinguer le type de l'axe, on se sert de leurs utypes, ainsi, pour un axe donné si ses utypes commencent par le préfix « CharacterisationAxis » alors ce dernier est de type générique (sinon de type spécifique).

### vi) Création du fichier Uty

Afin qu'il soit bien interprété par **CharacterisationLibrary**, notre fichier **Uty** doit obéir à un certain format (cf. Figure 16) :

- Le fichier Uty est un fichier parfile (propriété, valeur).
- Chaque ligne à interpréter doit commencer par le symbole « % »
- La Characterisation de chaque axe doit être faite dans un bloc séparé.
- Chaque bloc doit posséder un entête de la forme :  
CharacterisationAxis indice\_ du\_ bloc
- Le 1<sup>er</sup> utype de chaque bloc doit indiquer le nom de l'axe.

Le format du fichier présenté ci-dessus, s'est affiné au fur et à mesure du développement de **FitsToCharacterisation**. En effet **CharacterisationLibrary** n'était pas encore en version stable, de ce fait, chaque fois qu'une nouvelle version sortait des réunions avec son auteur (Brice Gassmann : ingénieur au CDS) étaient nécessaires afin de faire les tests et de relever les éventuelles incompatibilités entre nos deux bibliothèques.

**Remarque :** Pour des raisons d'optimisation, certaines étapes de cette phase peuvent être fusionnées, ainsi l'énumération précitée est donnée à titre indicatif pour illustrer davantage les différentes étapes.

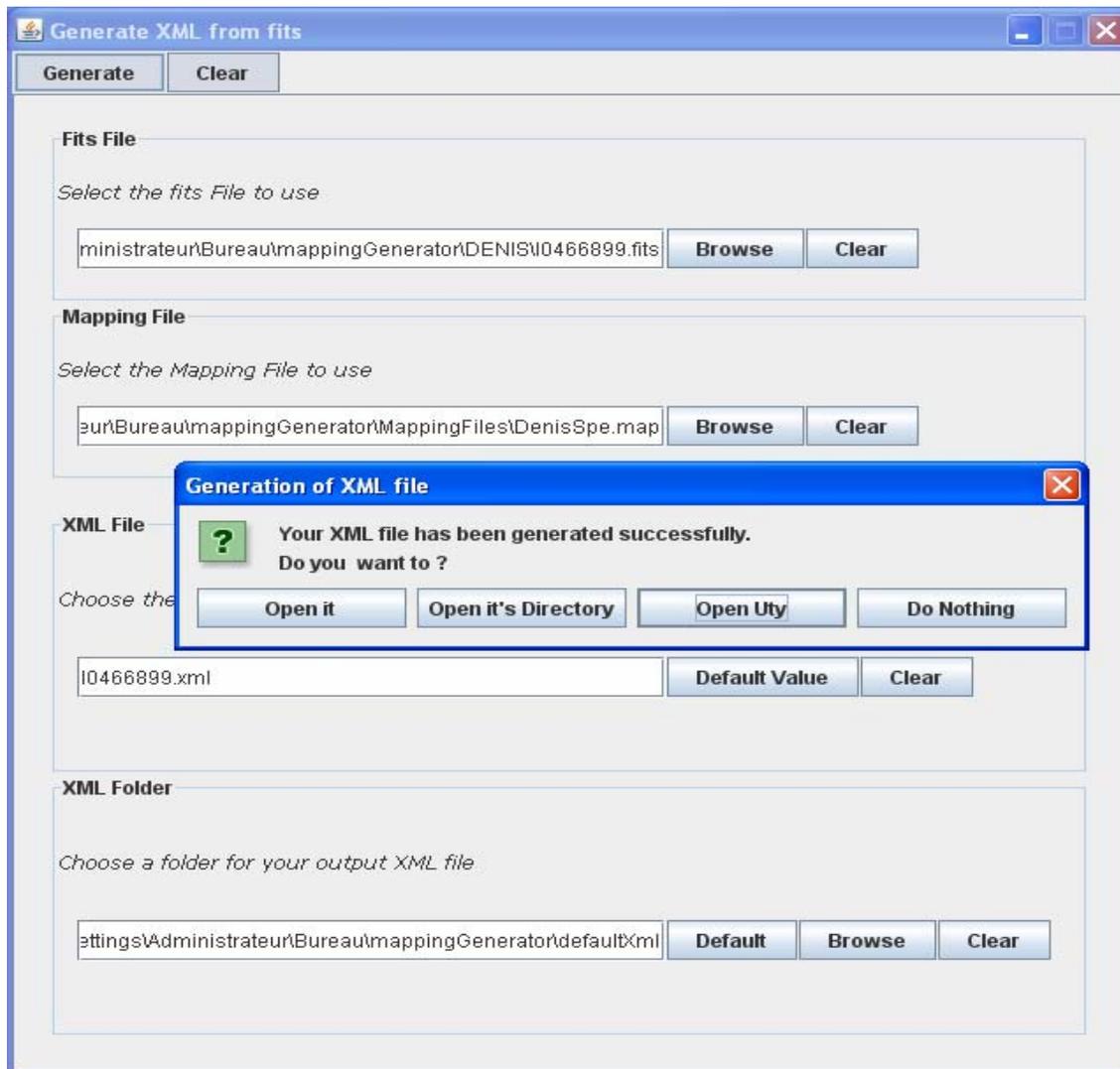
## b. Phase 2: génération du fichier XML

Une fois toutes les étapes de la phase une achevées, le fichier Uty résultant sert comme une entrée à (la librairie) **CharacterisationLibrary** dont le but de sa conversion au format XML.

(Pour plus d'informations concernant cette la génération du fichier Characterisation à l'aide de cette librairie consulter l'annexe n°: 03)

## 6. Illustration

Pour faciliter la manipulation de notre librairie, en plus du mode script j'ai réalisé une application graphique assurant les principales fonctionnalités souhaitées. (cf. figure20 un petit exemple d'utilisation de la librairie en mode fenêtre)



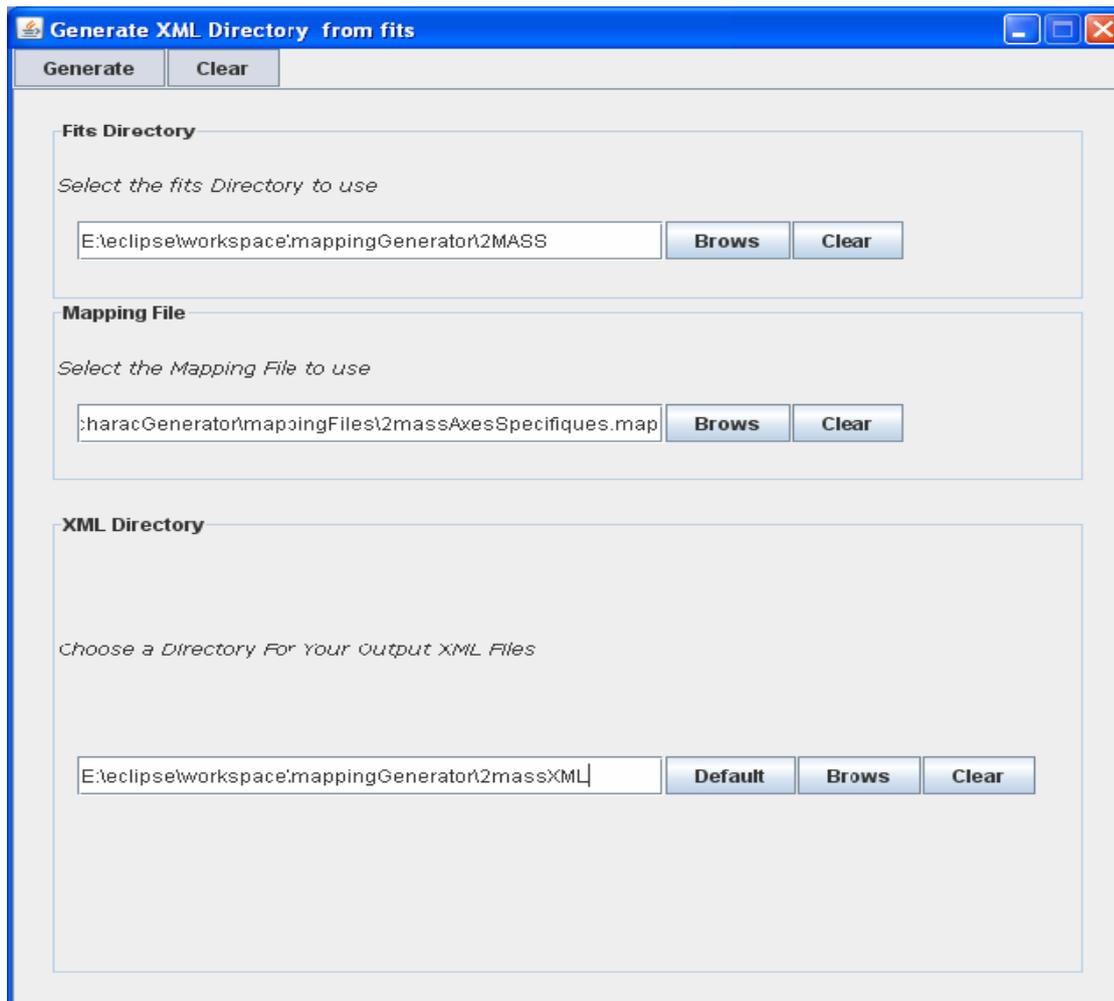
**Figure 20: Interface permettant de générer la Charc. XML**

Cette fenêtre est composée de deux parties principales : la 1<sup>ère</sup> concerne les éléments en entrée à savoir le fichier FITS et le fichier de mapping, la seconde est relative au fichier de sortie.

On commence par le choix des éléments en entrée qui sont obligatoires (fichier **FITS** et le fichier de mapping) et on donne ensuite le nom du fichier **XML** ainsi que le répertoire de sortie dans lequel sera enregistré le fichier généré. Il est possible de ne pas spécifier ces deux derniers paramètres, dans ce cas les éléments en sortie prennent des valeurs par défaut.

Une fois que les champs obligatoires sont remplis, un clic sur le bouton « **Generate** » permet de lancer le traitement. Pour plus d'informations (concernant les fichiers d'entrée et de sortie) relatives à cet exemple consulter l'annexe n°: 06.

L'exemple précité traite le cas d'un seul fichier **FITS** en entrée, mais il existe un autre cas plus intéressant dans lequel on souhaite exercer le traitement sur une famille de fichiers **FITS** possédant un seul fichier de mapping en commun, l'interface ci-dessous est destinée à ce genre de manipulation.



**Figure 21: Interface permettant de générer un répertoire de fichiers XML**

Comme pour le cas d'un seul fichier **FITS**, il est nécessaire de spécifier les deux paramètres d'entrée à savoir le répertoire contenant les fichiers **FITS** et le fichier de mapping correspondant. Pour les éléments en sortie (fichiers **XML** créés), on ne donne cette fois-ci que le répertoire de destination dans lequel ils seront enregistrés.

## 7. Evaluation de la solution

### a. Qualités

La solution implémentée présente plusieurs points forts :

- Fonctionnement en deux modes : script et fenêtre.
- Outre l'utilisation dans **AÏDA** plusieurs implémentations de la librairie développée sont possibles. A titre d'exemple significatif une intégration est en cours dans l'outil **CAMEA** (utilitaire développé au **CDS** permettant la création et la visualisation des fichiers Characterisation en mode graphique).
- Résultats de test encourageants : sur un jeu de test d'une centaine d'images appartenants à deux familles différentes de fichiers **FITS** les fichiers de **Characterisation** obtenus étaient satisfaisants à 100%.
- .....

### b. Critiques

Le point faible qu'on peut reprocher à notre solution est la création manuelle du fichier de mapping, en effet la taille et la complexité syntaxique de ses constituants rendent sa rédaction laborieuse et source d'éventuelles erreurs de saisie.

## Conclusion

A partir d'une image **FITS** et d'un **fichier de mapping**, la génération de la **Characterisation XML** est possible en utilisant la librairie **FitsToCharacterisation**. Une intégration efficace de cette librairie dans **AÏDA** complètera la fonctionnalité manquante en sortie des outils et assurera la validation de leur enchaînement.

## B. Intégration de FitsToCharacterisation dans AIDA et validation du workflow

Le but de l'intégration la librairie **FitsToCharacterisation** dans **AIDA** est de rendre possible la vérification totale d'un workflow. En effet à chaque itération (produisant une ou des images en sortie d'un outil) du processus de validation la librairie est invoquée pour caractériser les images en sortie de chaque outil. La technique d'intégration présentée ci-dessous, est un 1<sup>er</sup> prototype pour la validation de l'intégrité du workflow.

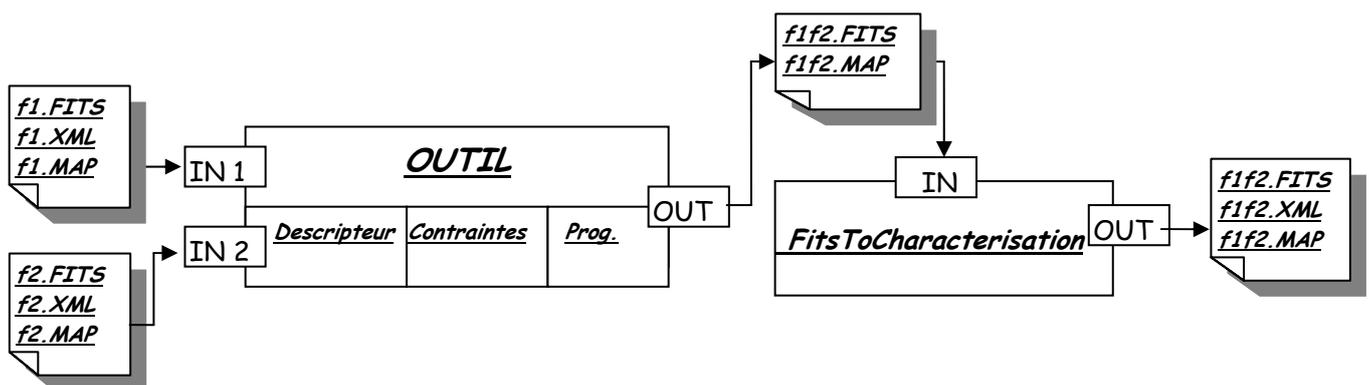
### 1. Intégration de la librairie au niveau des outil du workflow

#### a. Construction d'un fichier de mapping générique

Partant du principe que les outils de traitement d'images intégrés dans **AIDA** préservent les **mots clés** de l'entête **FITS**, (mais modifient éventuellement leurs valeurs), donc, si on concatène les fichiers de mapping associés à chaque entrée (de type image) d'un outil, on obtient un fichier de mapping générique valable pour toutes les images en sortie.

#### b. Génération du fichier Characterisation (en sortie des outils)

En associant aux images en sortie d'un outil un **fichier de mapping générique** (concaténation des fichiers de mapping en entrée de l'outil) leur caractérisation devient possible à l'aide de la librairie **FitsToCharacterisation**. (cf. figure 22)



**Figure 22: Génération du fichier Characterisation en sortie d'un outil**

### 2. Validation d'un workflow

La validation d'un workflow consiste en une chaîne de validation le long de ses constituants, à une étape **i** donnée de ce processus, les fichiers Characterisation générés (en sortie d'un outil) servent pour vérifier les contraintes imposées par les outils en étape **i+1** et ainsi de suite jusqu'au dernier outil du workflow et sa validation totale.

### 3. Critiques de l'intégration

#### Problème :

La manière dont nous avons construit le fichier de mapping en sortie des outils peut mener vers des situations de conflits non contrôlées. En effet l'utilisation arbitraires des **mots clés** dans un entête **FITS** peut conduire à des cas de figure dans lesquels des fichiers de mapping en entrée d'un outil possèdent des mots clés similaires (d'après l'expérience ce cas est très rare mais probable), ce qui entraînent leur apparition plusieurs fois dans le fichier de mapping résultant. En dépit de la rareté de ce cas, il faut prévoir un moyen pour le gérer afin d'éviter toutes sortes de conflits menant vers des résultats erronés.

#### Réflexion sur le problème :

Une 1<sup>ère</sup> réflexion peut porter sur l'analyse des différents cas d'utilisations possibles pour les entrées d'un outil. En effet on peut envisager les situations suivantes :

- **Un seul fichier de mapping en entrée** (cas des images d'une même famille en entrée) : dans ce cas le fichier de mapping en sortie sera le même que celui en entrée c'est-à-dire pas de conflits.
- **Des fichiers de mapping avec des mots clés différents en entrée** : ce cas aussi ne présente pas de conflits dans le fichier de mapping généré en sortie de l'outil.
- **Des fichiers de mapping avec des mots clés redondants en entrée** : on distingue deux cas :
  - ❖ des mots clés redondants associés au même utype : la suppression des lignes qui présentent des répétitions (mot clé, utype) résout le problème.
  - ❖ des mots clés redondants associés à des utypes différents : dans ce cas on peut opérer des modifications au niveau des entêtes FITS en renommant les mots clés présentant des répétitions avant l'exécution de l'outil (les mots clés d'une entête **FITS** n'ont que des significations symboliques, de ce fait leurs modifications n'affectent pas le contenu du fichier **FITS**).

**Remarque** : le planning de stage prévoit le traitement de ce point en parallèle à l'intégration effective de la solution dans **AÏDA** après la soutenance.

#### Conclusion

L'utilisation de la librairie **FitsToCharacterisation** permet de caractériser les images en sortie des outils, la validation de leur enchaînement devient ainsi possible. On peut par conséquent améliorer l'exécution d'un workflow en diminuant les risques d'erreurs à chaque étape.

## C. Outil Interactif de mapping

### 1. Besoin

Le fichier de mapping fourni à l'outil de génération est à rédiger manuellement dans un éditeur de texte, d'une part cela constitue une tâche très lourde pour son auteur à cause de sa taille ainsi que de la longueur des utypes le constituant, et d'autre part cela mène souvent à des erreurs dans le fichier de mapping (erreurs de saisie) et par conséquent dans le fichier Characterisation obtenu.

Une erreur sur une ligne du fichier de mapping implique automatiquement, (au moins) un utype manquant dans le fichier Characterisation de sortie ou voire plus, si l'erreur porte sur le nom de l'axe, alors tout le bloc des utypes le concernant est omis, c'est-à-dire une Characterisation avec un axe en moins.

### 2. Solution

Afin d'éviter toutes sortes d'erreurs dues aux fichiers de mapping (qui pourraient influencer la Characterisation), nous avons décidé d'offrir aux créateurs de ces fichiers un outil graphique et interactif regroupant tous les matériaux nécessaires à l'élaboration de ce fichier.

### 3. Spécifications de l'outil

L'outil à prévoir doit offrir un environnement de travail simple et convivial assurant les fonctionnalités suivantes :

- Une représentation structurée de la liste des **utypes** et des **mots clés** de l'entête FITS (dans l'espace de travail).
- La possibilité d'associer un **utype** à un **mot clé** à l'aide du Drag&Drop (Glisser - Déposer).
- La possibilité de sauvegarder l'état de l'espace de travail dans le but d'une reprise ultérieure.
- Une base de connaissances aidant au choix des utypes.
- ...

### 4. Conception de l'outil

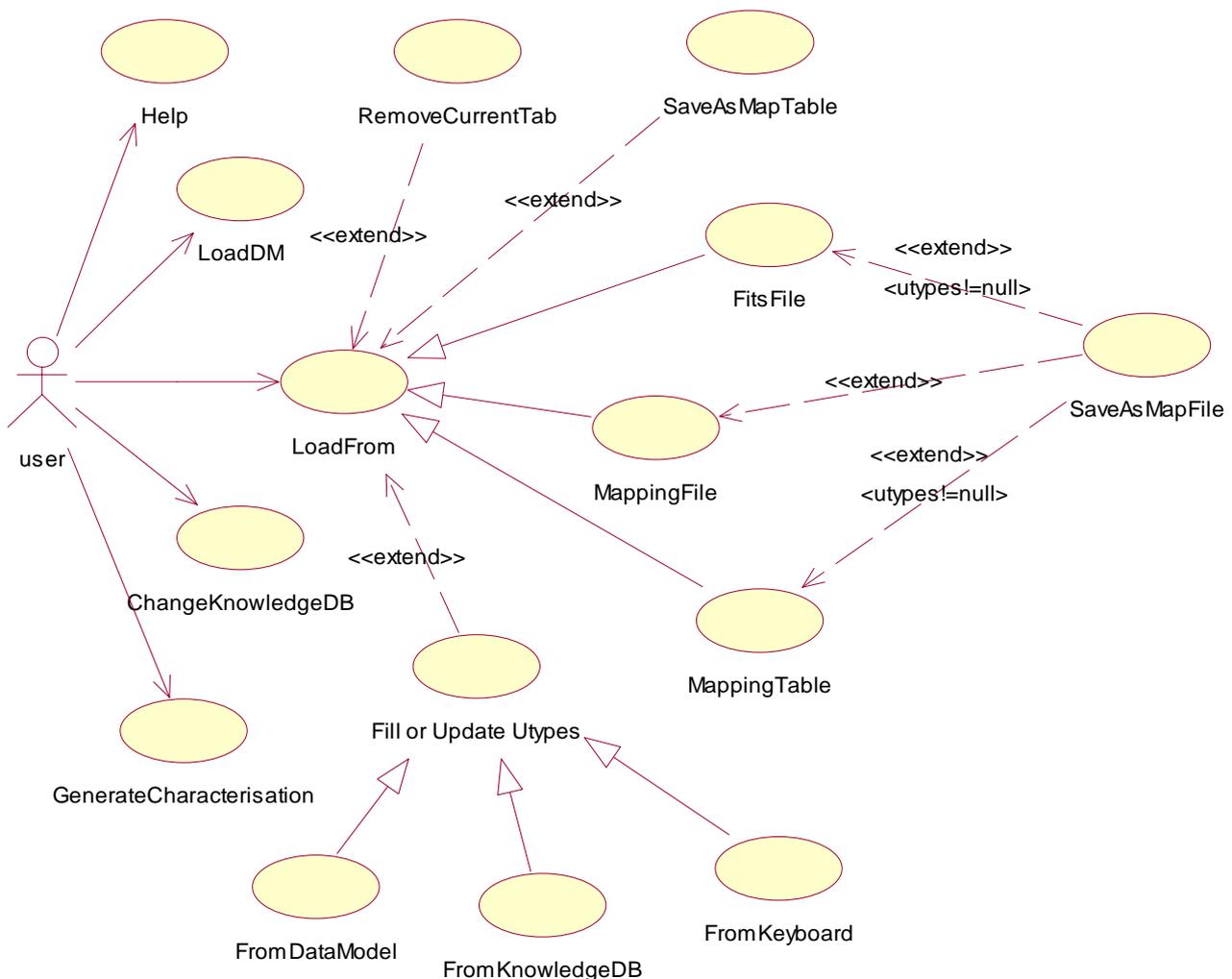
Avant d'entamer la formalisation des besoins requis par l'outil de mapping, je tiens à signaler que le CDS dispose d'un logiciel «SAADA » (développé en JAVA par une autre équipe de l'observatoire) destiné à la génération de bases de données scientifiques d'une façon générique, qui comprend entre autres un service pour la création de fichiers de mapping. Au départ l'enjeu était d'exploiter cette partie ou plutôt de l'extraire de « SAADA » dans le but d'une adaptation aux besoins de notre outil, cette idée nous paraissait très intéressante, car si nous arrivions à séparer le fragment voulu (du reste de SAADA) et à l'intégrer dans notre solution, cela pouvait nous dispenser d'une grande partie de la maintenance et réduire considérablement le temps de développement. Dans ce cadre,

quelques réunions se sont tenues avec le responsable de SAADA (Laurent MICHEL ingénieur de recherche dans l'équipe « **Hautes Energies** ») afin de nous présenter son logiciel.

En analysant ce logiciel, nous avons constaté que ce dernier n'était pas conçu pour qu'il soit réutilisable en parties, en effet le nombre important de dépendances entre ses classes ne permet pas une isolation aisée de la partie mapping sans que cela n'affecte son architecture interne (de classes). Vu la contrainte de temps, nous nous sommes aperçu que refaire de nouveau la partie mapping serait plus facile et plus rapide que la réutilisation d'une partie de SAADA.

### a. Diagramme des cas d'utilisation

Les fonctionnalités requises par notre outil de mapping peuvent se schématiser dans le diagramme des cas d'utilisation suivant :



**Figure 23: Diagramme des cas d'utilisation de l'outil de mapping**

## b. Description de quelques cas d'utilisation

### ▪ LoadFrom

Ce cas d'utilisation représente les supports possibles pour créer un fichier de mapping, à partir d'un :

- ❖ fichier FITS (FitsFile).
- ❖ autre fichier de mapping (MappingFile)
- ❖ espace de travail (MappingTable) : cela rend possible la reprise de la création d'un fichier de mapping non complet.

### ▪ LoadDM

Ce cas sert à recharger la liste des utypes dans l'espace de travail.

### ▪ LoadKnowledgeDB

Ce cas permet de charger une nouvelle base de connaissance.

### ▪ Fill or update Utypes

Ce cas montre la manière dont on peut associer un utype à un mot clé (ou la mise à jour de cette association), cette opération peut se faire à l'aide :

- ❖ du modèle de données : liste des utypes (FromDataModel)
- ❖ de la base de connaissances (FromKnowledgeDB)
- ❖ des touches du clavier : saisie traditionnelle (FromKeyboard)

### ▪ SaveAsMapTable

Ce cas d'utilisation représente le fait qu'on pourrait sauvegarder l'état courant de notre travail dans le but d'une reprise ultérieure.

### ▪ SaveAsMapFile

Ce cas d'utilisation permet de créer un fichier de mapping, l'opération n'est accessible que si au moins une association mot clé - Utype est présente (dans l'espace de travail).

### ▪ GenerateCharacterisation

Ce cas d'utilisation permet d'instancier la librairie FitsToCharacterisation en mode fenêtre.

▪ ...

## c. La base de connaissance

L'intégration d'une base de connaissance à notre outil à pour but de faciliter davantage la création du fichier de mapping. En effet cette dernière devrait à la demande du créateur, offrir une liste des **utypes** qui peuvent correspondre à un **mot clé** donné.

### i) Choix d'implémentation

Pour implémenter la base de connaissances, plusieurs choix étaient possibles à savoir, l'utilisation :

- d'un SGBD (Système de Gestion de Base de Données) tel que MySql ou PostgreSQL.
- des objets java sérialisés
- de XML
- ...

Etant donné que notre base de connaissance ne présente qu'un faible volume de données, l'utilisation d'un SGBD avec ses différents avantages (notamment en terme d'accès et de manipulation de données) était un choix à éviter en faveur d'une solution préservant le fonctionnement autonome de notre outil sans aucune couche logiciel supplémentaire.

L'utilisation des objets sérialisés est aussi un choix à éliminer car, elle rend la base moins utilisable en dehors de l'outil de mapping. Enfin le choix d'une implémentation XML s'avère la solution la plus adaptée à nos besoins car elle permet une représentation de données sous un format bien structuré et facilement exploitable même dans des environnements hétérogènes.

### ii) Schéma de la base de connaissances

La base de connaissances utilisée est constituée d'un ensemble d'associations de type : **mot clé – utypes** (pour chaque **mot clé** de la base correspond un ou plusieurs **utypes** cf. **figure 24**).

```
<?xml version="1.0" encoding="UTF-8" ?>
-<keywords>
  -<keyword name="AXIS1UNI">
    <utype>SpatialAxis.unit</utype>
    <utype>CharacterisationAxis.unit</utype>
  </keyword>
  -<keyword name="AXIS2NAM">
    <utype>TimeAxis.AxisName</utype>
    <utype>CharacterisationAxis.AxisName</utype>
  </keyword>
  -<keyword name="AXIS2UCD">
    <utype>TimeAxis.ucd</utype>
    <utype>CharacterisationAxis.ucd</utype>
  </keyword>
  .....
</keywords>
```

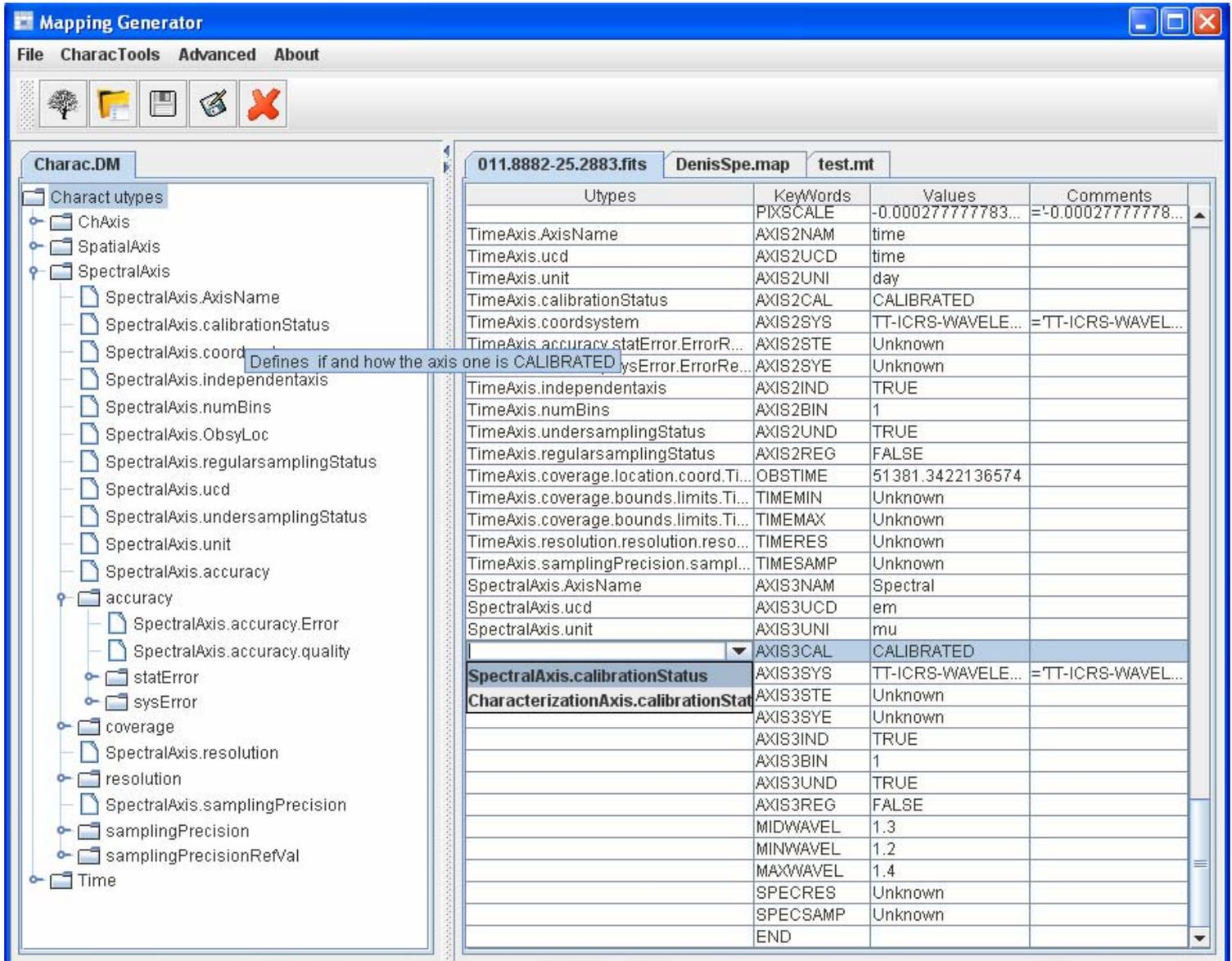
**Figure 24: Extrait d'une base de connaissance XML**

### iii) Alimentation de la base

Les fichiers de mapping constituent la matière première de notre base de connaissances, on peut se servir de chaque fichier créé pour enrichir le contenu de la base.

## 5. Description de l'outil de mapping

### a. Vue d'ensemble



Utypes	KeyWords	Values	Comments
	PIXSCALE	-0.0002777777783...	=-0.000277777778...
TimeAxis.AxisName	AXIS2NAM	time	
TimeAxis.ucd	AXIS2UCD	time	
TimeAxis.unit	AXIS2UNI	day	
TimeAxis.calibrationStatus	AXIS2CAL	CALIBRATED	
TimeAxis.coordsystem	AXIS2SYS	TT-ICRS-WAVELE...	=TT-ICRS-WAVEL...
TimeAxis.accuracy.statError.ErrorR...	AXIS2STE	Unknown	
TimeAxis.accuracy.sysError.ErrorRe...	AXIS2SYE	Unknown	
TimeAxis.independentaxis	AXIS2IND	TRUE	
TimeAxis.numBins	AXIS2BIN	1	
TimeAxis.undersamplingStatus	AXIS2UND	TRUE	
TimeAxis.regularsamplingStatus	AXIS2REG	FALSE	
TimeAxis.coverage.location.coord.Ti...	OBSTIME	51381.3422136574	
TimeAxis.coverage.bounds.limits.Ti...	TIMEMIN	Unknown	
TimeAxis.coverage.bounds.limits.Ti...	TIMEMAX	Unknown	
TimeAxis.resolution.resolution.reso...	TIMERES	Unknown	
TimeAxis.samplingPrecision.sampl...	TIMESAMP	Unknown	
SpectralAxis.AxisName	AXIS3NAM	Spectral	
SpectralAxis.ucd	AXIS3UCD	em	
SpectralAxis.unit	AXIS3UNI	mu	
	AXIS3CAL	CALIBRATED	
<b>SpectralAxis.calibrationStatus</b>	AXIS3SYS	TT-ICRS-WAVELE...	=TT-ICRS-WAVEL...
<b>CharacterizationAxis.calibrationStat</b>	AXIS3STE	Unknown	
	AXIS3SYE	Unknown	
	AXIS3IND	TRUE	
	AXIS3BIN	1	
	AXIS3UND	TRUE	
	AXIS3REG	FALSE	
	MIDWAVEL	1.3	
	MINWAVEL	1.2	
	MAXWAVEL	1.4	
	SPECRES	Unknown	
	SPECSAMP	Unknown	
	END		

**Figure 25: Interface graphique de l'outil de mapping**

L'espace de travail de notre outil est constitué de deux parties principales, la 1<sup>ère</sup> (partie gauche) est destinée à contenir le modèle de données et se présente sous la forme d'un arbre de utypes hiérarchisés en axes et ensuite en propriétés, elle sert ainsi comme une source de

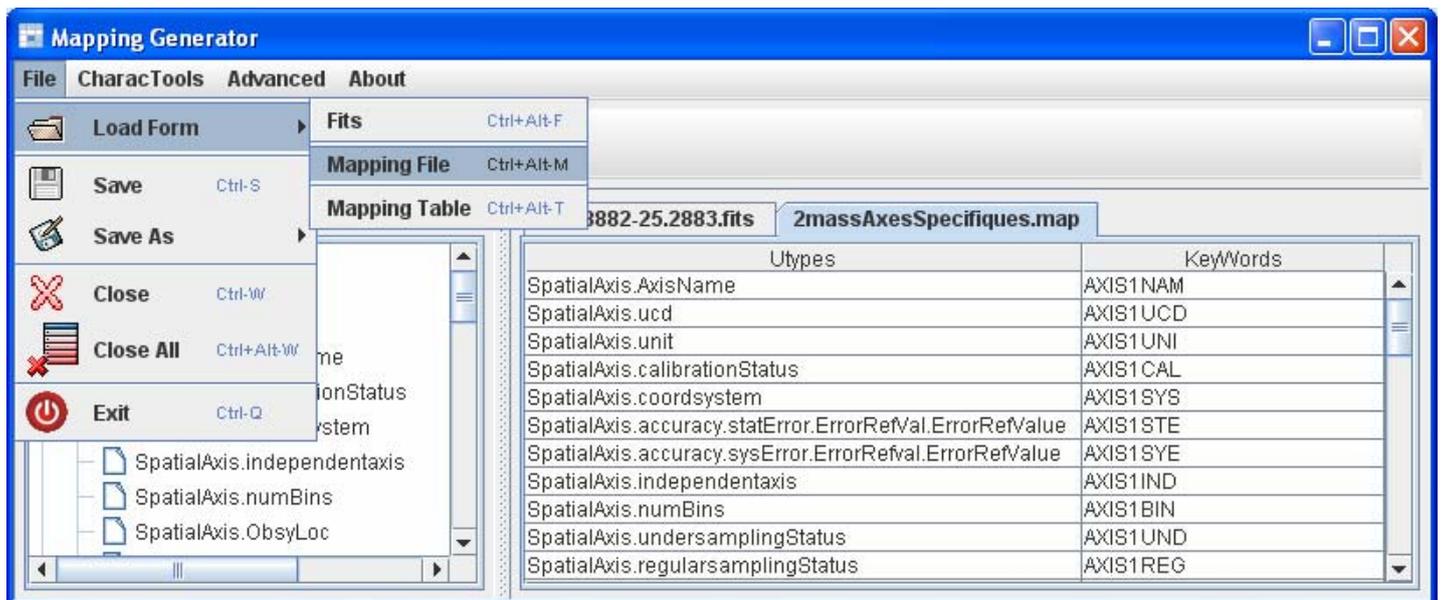
l'opération du Drag & Drop. Au passage de la souris chaque item de l'arbre affiche une bulle d'information décrivant le « **utype** » en cours.

La seconde partie sert à visualiser des fichiers FITS, de mapping ou des « mapping Tables », chaque fichier chargé s'affiche sous la forme d'un tableau -dans un nouvel onglet- qui contient :

- Une 1<sup>ère</sup> colonne « **Utypes** » dont chaque cellule est composée d'un combobox se remplissant à partir de la base de connaissances et qui peut aussi accepter des opérations de Drag & Drop en provenance du Modèle de données.
- Une 2<sup>ème</sup> colonne pour les « **KeyWords** »
- Et deux autres colonnes supplémentaires « **Values** » et « **Comments** » affichables uniquement dans le cas d'un fichier de type FITS ou mt(mapping table).

## b. Quelques éléments de menu

### i) Load From

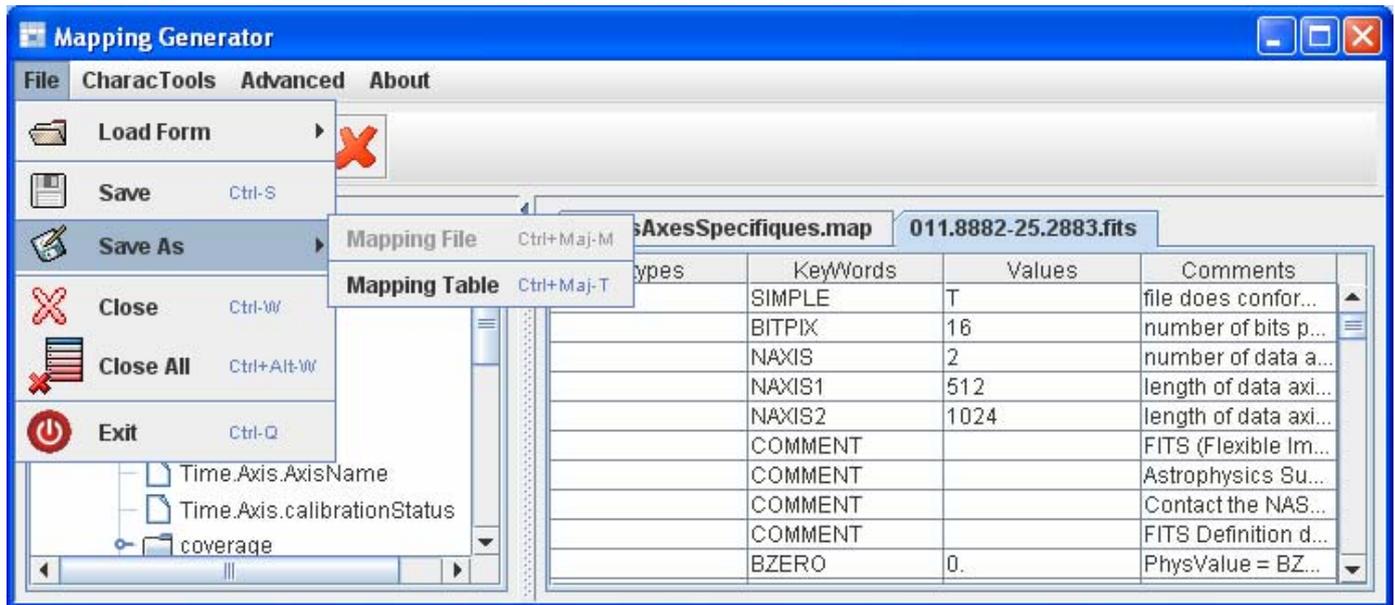


**Figure 26: Items du menu File et de Load From plus affichage d'un fichier de mapping**

La figure 26 illustre les éléments du menu File et en particulier « **Load From** » qui permet de choisir un support pour la création du fichier de mapping parmi **FITS**, **Mapping File** ou **Mapping Table**. Cette figure montre aussi un fichier de mapping chargé à l'aide de :

***File → Load From → Mapping File → 2massAxesSpecifiques.map***

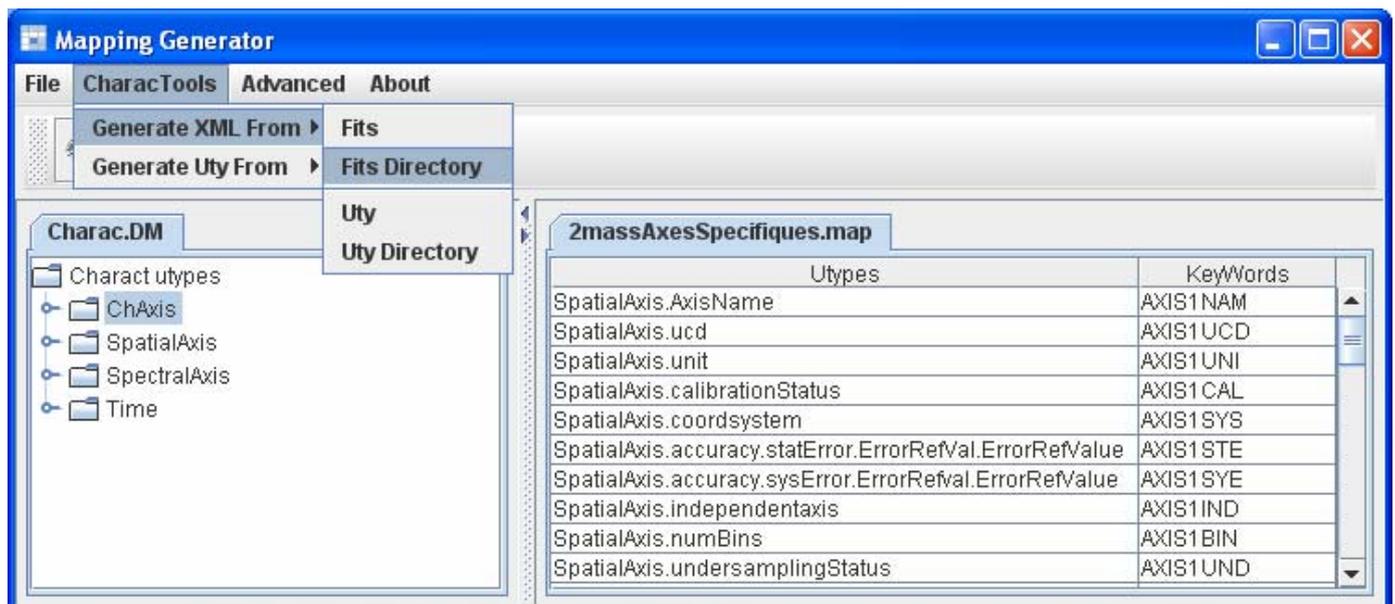
## ii) Save As



**Figure 27: Items du menu File et de Save As**

La figure 27 met en évidence l'item **Save As** du menu **File** qui permet de sauvegarder notre travail sous deux formats : **Mapping File** et **Mapping Table**, n'offrant le cas échéant que la 2<sup>ème</sup> possibilité et cela est due au fait que les utypes de l'onglet courant sont tous vides (cf. partie droite de la figure 27).

## iii) CharacTools



**Figure 28: Menu CharacTools**

La figure 28 représente le menu **CharacTools** et ses items, ce menu permet de faire le lien avec la librairie : **FitsToCharacterisation** en leur offrant quelques implémentations graphiques.

## Conclusion

Au début, la création de fichier de mapping se faisait dans un simple éditeur de texte, estimée comme une tâche plutôt embarrassante, prenant un temps considérable et source d'erreurs. Maintenant, à l'aide de notre outil la création s'effectue en quelques clics souris et offre des résultats plus sûrs et plus performants.

## D. Portage du serveur AÏDA

### 1. Besoin

Actuellement le serveur **AÏDA** ne fonctionne que sous **Linux**. L'utilisation d'outils écrits pour **Windows** est donc très délicate car elle nécessite un portage lorsqu'il ne s'agit pas de code **Java**. Etant donné que **APACHE** est le serveur **HTTP** utilisé (par le serveur AÏDA), un autre objectif sera le déploiement du serveur sous **TOMCAT** (qui est un serveur **HTTP** écrit en **JAVA** donc « portable »).

### 2. Analyse de la situation

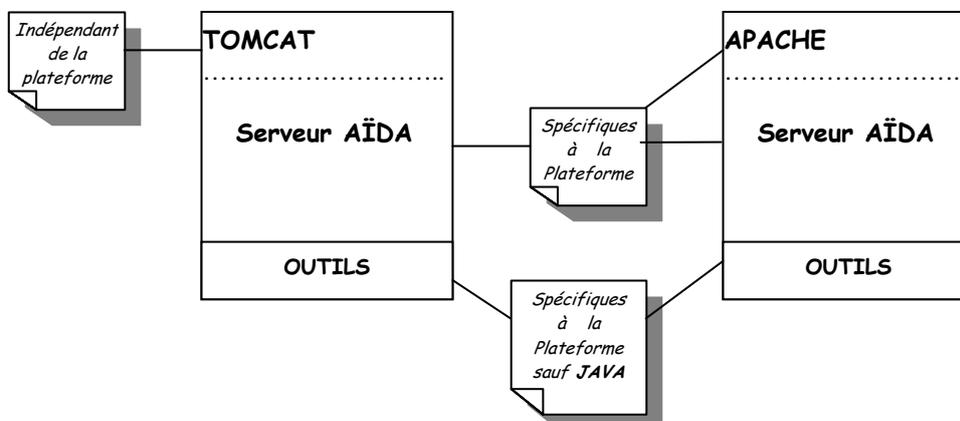
Le serveur **AÏDA** est essentiellement destiné à héberger des outils de traitement d'images pas forcément homogènes. Il se base sur un ensemble de scripts **Perl** et utilise le serveur **HTTP APACHE** pour uniformiser l'accès aux outils via des **CGI (Common Gateway Interface)**, mais possède également différentes interfaces clientes comme **JAVA** et un script d'exécution en **Perl**.

### 3. Démarche du portage

#### a. Portage sous TOMCAT

TOMCAT est essentiellement indiqué pour être un conteneur de servlet/JSP, mais il est doté de beaucoup de possibilités lui permettant de rivaliser avec des serveurs web traditionnels. Une de ces possibilités est le support des **CGI** qui apportent le moyen de lancer un programme externe en réponse à une requête de navigateur, habituellement pour traiter un formulaire web.

Le portage du **serveur AÏDA** sous **TOMCAT** repose sur trois étapes principales, la première vise à configurer le serveur **TOMCAT** pour la prise en charge des scripts **CGI**, la deuxième consiste en quelques paramétrages coté **client AÏDA** et la dernière sert pour le déploiement du serveur sous **TOMCAT** (pour plus de détail cf. annexe n°: 5).



**Figure 29: Comparaison du serveur AÏDA sous TOMCAT et sous APACHE**

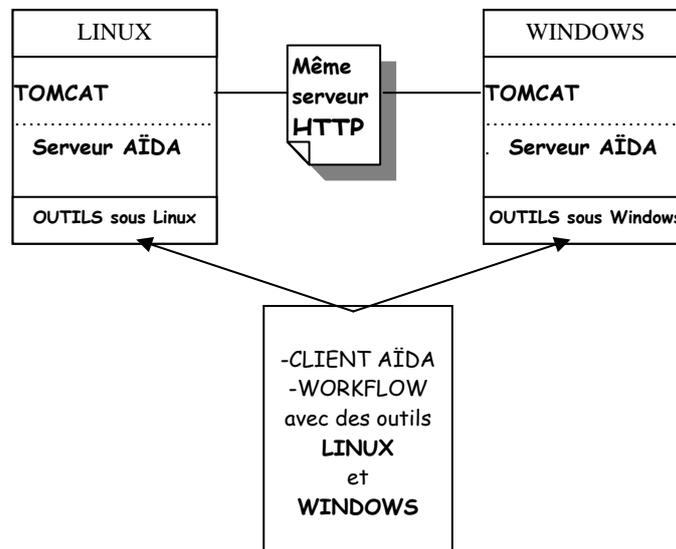
**Remarques** : le serveur TOMCAT est écrit en **JAVA** donc portable, par contre les outils (intégrés dans AÏDA) ne sont pas rendus portables par la même occasion, car par exemple un outil écrit en **C** sous Windows ne s'exécutera pas sous Linux même avec l'utilisation de TOMCAT (cf. figure 30).

### b. Portage sous Windows

Le portage du serveur **AÏDA** sous **Windows** consiste en :

- Une adaptation des scripts **Perl** constituant le noyau du serveur pour qu'ils s'exécutent sous **Windows**.
- Une conversion du script (**shell**) d'installation et de déploiement du serveur sous **Linux** en un **batch** interprétable sous **Windows**.

## 4. Serveur AÏDA après le portage



**Figure 30: serveur AÏDA après le portage**

Après le portage, on dispose de deux **serveurs AÏDA**, un est dédié aux outils Windows et l'autre aux outils Linux, ainsi les requêtes en provenance des clients (vers le serveur AÏDA) se dispatchent selon l'OS des outils vers l'un de ces deux serveurs (cf. figure 31)

### Conclusion

Le portage du serveur **AÏDA** a répondu aux problèmes liés à l'intégration des outils développés sous Windows (portage lorsque il ne s'agit pas du JAVA), **AÏDA** peut ainsi héberger plus d'outils et offrir plus de services.

## Perspectives

Les perspectives à court terme consisteront en l'intégration effective de la librairie **FitsToCharacterisation** dans **AÏDA** et en la validation d'un workflow (objectifs de la 2ème partie du stage). A l'avenir, il serait intéressant que les fournisseurs des outils de traitements d'image prennent en charge le modèle **Characterisation** ainsi si l'outil est utilisable dans **AÏDA** la présence d'un fichier **Characterisation** sera un plus.

Au niveau de l'**outil de mapping**, on peut prévoir des améliorations concernant sa base de connaissances, à deux niveaux :

- **Alimentation de la base** : outre les fichiers de mapping, on peut exploiter d'autres connaissances pour enrichir les données, par exemple le modèle **Characterisation** spécifie pour chaque **utype** le type de sa valeur (booléen, entier, intervalle,...), donc à partir de la valeur d'un **mot clé** on peut avoir une liste exhaustive des utypes possibles.
- **Centralisation de la base** : on peut aussi définir une base de connaissances centrale accessible depuis Internet et fonctionnant à deux sens, **base** → **outil de mapping** : synchronisation de la base de connaissances de l'outil par rapport au référentiel.

**Outil de mapping** → **base centrale** : alimentation de la base centrale à partir d'un fichier de mapping généré au niveau d'un outil.

## Bilan

Ce stage fut riche d'enseignements pour différentes raisons. D'abord il m'a permis d'appliquer les connaissances acquises lors de mon cursus universitaire pour traiter différents problèmes informatiques. Ensuite, j'ai été confronté à la modification d'une application dont je n'étais pas l'auteur. Ceci m'a confirmé l'utilité des documentations techniques et/ou utilisateurs ainsi que des commentaires dans le code source.

En plus des acquis en informatique ce stage m'a apporté une première expérience en recherche. Conformément à ce que j'attendais, l'organisation et la méthode de travail sont assez différentes des autres domaines. En effet, l'application à développer se doit d'être suffisamment évolutive, car chaque rencontre avec l'utilisateur et chaque test peuvent nous obliger à partiellement ou à totalement réorganiser certaines parties de l'application.

A l'issue de la 1<sup>ère</sup> partie du stage (compte tenu qu'il dure encore un mois de plus), l'ensemble des objectifs planifiés ont été atteints avec des taux de satisfaction encourageants. En effet la librairie **FitsToCharacterisation** a répondu au problème de **Characterisation** en sortie des outils d'**AÏDA**. Profitant du fait que l'étendu de la librairie ne se restreint pas au contexte d'**AÏDA**, plusieurs implémentations de **FitsToCharacterisation** sont possibles, à titre d'exemple, une intégration de la librairie est en cours au niveau de **CAMEA** (outil développé au CDS servant à manipuler des fichiers **Characterisation**). L'outil de mapping développé au cours de cette période a renforcé également cette librairie en offrant un moyen plus pratique et sûr pour la création des fichiers de mapping.

Le travail conclu à l'issue de cette période peut se conjuguer avec des outils du **CDS** tel que **CAMEA** et **CharacterisationLibrary** pour former un pack qui fournit un support au standard **Characterisation**.

## Conclusion

Ce stage m'a beaucoup apporté autant professionnellement que personnellement. Il m'a permis d'en savoir plus en informatique, en astronomie et sur mon avenir professionnel. J'ai maintenant une idée beaucoup plus précise et concrète de mon futur emploi.

En outre, les problèmes que j'ai pu rencontrés m'ont permis de progresser et d'acquérir une expérience plus pratique de l'informatique qui me sera utile pour mes futures expériences professionnelles.

J'ai particulièrement apprécié ce premier contact avec le monde de l'astronomie et j'espère renouveler une telle expérience.

## Bibliographie

### Adresses Internet :

- [1] <http://astro.ustrasbg.fr/observatoire/>
- [2] <http://cdsweb.ustrasbg.fr/>
- [3] <http://www.ivoa.net/Documents/latest/CharacterisationDM.html>
- [4] <http://www.ivoa.net/twiki/bin/view/IVOA/IvoaDataModel>
- [5] <http://java.sun.com/docs/books/tutorial/ui/index.html>

## Table des figures

FIGURE 1: PAGE WEB PERMETTANT D'EFFECTUER UNE REQUETE SUR SIMBAD.....	12
FIGURE 2: PAGE WEB PERMETTANT D'EFFECTUER UNE REQUETE SUR VIZIER.....	13
FIGURE 3: EXEMPLE D'UTILISATION D'ALADIN.....	14
FIGURE 4: FONCTIONNEMENT GENERAL D'AÏDA.....	17
FIGURE 5: APPLICATION CLIENTE: LISTE DES OUTILS (G) ET UN OUTIL (D).....	18
FIGURE 6: APPLICATION CLIENTE: WORKFLOW.....	19
FIGURE 7: EXEMPLE DE FICHIER DESCRIPTEUR.....	20
FIGURE 8 : STRUCTURE PRINCIPALE SIMPLE D'UN FICHIER CHARACTERISATION.....	22
FIGURE 9 : EXTRAIT D'UN FICHIER DE CONTRAINTES.....	23
FIGURE 10: CONSTITUTION D'UN OUTIL.....	23
FIGURE 11: ESPACE DE TRAVAIL "WORKFLOW", VALIDATION DES CONTRAINTES.....	24
FIGURE 12: SCHEMA GENERALE DE L'OUTIL DE GENERATION.....	29
FIGURE 13: UN EXTRAIT DE L'ENTETE DU FICHIER FITS : 011.8882-25.2883.FITS.....	30
FIGURE 14: UN EXTRAIT DU FICHIER DE CHARACTERISATION : 011.8882-25.2883.XML ..	31
FIGURE 15: CORRESPONDANCES KEYWORDS/CHARACTERISATION DU FICHIER FITS : 011.8882-25.2883.FITS.....	32
FIGURE 16: UN EXTRAIT DU FICHIER DE CHARACTERISATION EN UTYPES: 011.8882-25.2883.UTY....	33
FIGURE 17: SCHEMA DE LA SOLUTION ADOPTEE.....	34
FIGURE 18: EXTRAIT DE FICHIER DE MAPPING : 2MASS.MAP.....	35
FIGURE 19: COMPARATIF DE QUELQUES LIBRAIRIES IMPLEMENTANT FITS.....	35
FIGURE 20: INTERFACE PERMETTANT DE GENERER LA CHARC. XML.....	37
FIGURE 21: INTERFACE PERMETTANT DE GENERER UN REPERTOIRE DE FICHIERS XML.....	38
FIGURE 22: GENERATION DU FICHIER CHARACTERISATION EN SORTIE D'UN OUTIL.....	40
FIGURE 23: DIAGRAMME DES CAS D'UTILISATION DE L'OUTIL DE MAPPING.....	43
FIGURE 24: EXTRAIT D'UNE BASE DE CONNAISSANCE XML.....	45
FIGURE 25: INTERFACE GRAPHIQUE DE L'OUTIL DE MAPPING.....	46
FIGURE 26: ITEMS DU MENU FILE ET DE LOAD FROM PLUS AFFICHAGE D'UN FICHIER DE MAPPING .....	47
FIGURE 27: ITEMS DU MENU FILE ET DE SAVE AS.....	48
FIGURE 28: MENU CHARACTOOLS.....	48
FIGURE 29: COMPARAISON DU SERVEUR AÏDA SOUS TOMCAT ET SOUS APACHE.....	50
FIGURE 30: SERVEUR AÏDA APRES LE PORTAGE.....	51

## Annexes

### Annexe n° 01 : Description de la librairie FitsToCharacterisation

Cette partie décrit les principales classes et méthodes de notre librairie ainsi que la manière de s'en servir.

#### 1. La classe MoCharacGeneration

A l'exception de la génération du fichier **UTY** cette classe assure toutes les étapes de la 1<sup>ère</sup> phase de génération.

#### 2. La classe CharacFile

Cette classe offre les méthodes nécessaires à la création des fichiers Characterisation aux formats **UTY** et **XML**.

#### Principales méthodes :

- **createUtypeFile(File utypeF, File fitsF, File mapF)**

Cette fonction permet de créer un fichier Characterisation au format **UTY** à partir du fichier **FITS** et du fichier de mapping passés en paramètres, un 3<sup>ème</sup> paramètre est aussi nécessaire pour spécifier le fichier **UTY** de sortie.

- **createUtypeDirectory(File dirUtype, File dirFits, File mapF)**

Cette fonction permet de caractériser tous les fichiers **FITS** contenus dans le répertoire passé en paramètre, outre ce dernier la fonction possède deux autres paramètres, un pour désigner le répertoire des fichiers **UTY** en sortie et l'autre relatif au fichier de mapping commun aux fichiers **FITS**.

- **generateXMLcharacFmFits(File xmlF, File fitsF, File mapF)**

A partir d'un fichier **FITS** et d'un fichier de mapping cette fonction permet de générer le fichier Characterisation correspondant au format **XML**, le fichier de sortie est désigné par le paramètre **xmlF**.

- **createXmlDirectoryFmFits(File dirXml, File dirFits, File mapF)**

La fonction en cours se sert d'un répertoire de fichiers **FITS** et de leur fichier de mapping commun passés en paramètres pour générer les fichiers Characterisation **XML** qui seront contenus dans le répertoire spécifié par l'argument **dirXml**.

- ...

### Annexe n° 02 : Quelques manipulations de la librairie nom.tam.fits

Avant de donner un aperçu de la manière dont j'ai utilisé cette librairie pour extraire les données, il serait judicieux que je fasse un petit retour sur la norme **FITS** pour illustrer de nouveau la structure d'une image selon cette recommandation.

-Chaque fichier **FITS** est composé d'un ou de plusieurs **HDU** (Header Data Unit).

-Chaque **HDU** comprend deux parties :

1. **Header** (Entête) : contient une série de "card" ASCII qui portent des paires de "**keyWord / value**" et un commentaire de plus en option.
2. **Data** : correspond aux données binaires de l'image.

#### Exemple d'utilisation de la librairie pour extraire les « card » d'un entête FITS

Tout d'abord on crée un objet de type Fits, le constructeur qui sera appelé prendra comme paramètre le fichier FITS.

```
Fits fi=new Fits(fitsFile);
```

L'objet ainsi créé nous servira pour effectuer des opérations de lecture et d'écriture sur l'image.

Ensuite on fait appelle à la méthode **readHDU()** qui nous retourne le prochain **HDU** contenu dans notre flot d'entré(fichier FITS)

```
ImageHDU ih=(ImageHDU)fi.readHDU();
```

Pour extraire la partie HEADER on applique la méthode **getHeader()** à l'objet qu'on vient de récupérer.

```
Header hdr=ih.getHeader();
```

Le HEADER ainsi obtenu est constitué d'une liste des « card », pour les récupérer, il suffit de la parcourir comme suit :

```
Cursor iter =hdr.iterator();  
while (iter.hasNext()) {  
    HeaderCard hc = (HeaderCard) iter.next();  
    String key=hc.getKey();  
    String value=hc.getValue();  
    String comment=hc.getComment();  
    .....  
}
```

**Remarque** : seuls les keyWords communs entre le fichier de mapping et FITS seront extraits.

#### **Annexe n° 03 : Utilisation de la librairie CharacterisationLibrary**

Une fois le fichier Uty généré on fait appel aux services de la librairie CharacterisationLibrary, ainsi la conversion au format XML s'effectue en deux instructions.

Tous d'abord, on crée un objet de type **CharacterisationType** en passant en paramètre le fichier Uty à la fonction **fromUTypetoObject** (de la classe **CharacConversions**).

```
CharacterisationType ct = CharacConversions.fromUTypetoObject(utyFile);
```

L'objet ainsi créé sert comme premier paramètre de la fonction **fromObjectToXML** qui exige un second argument de plus pour désigner le fichier XML de sortie.

```
CharacConversions.fromObjectToXML(ct, xmlFile);
```

#### Annexe n° 04 : Illustration de la génération du fichier Characterisation.

##### ▪ Le fichier FITS : Header du fichier I0466899.fits

KeyWord	Value	Comment
SIMPLE	T	FITS STANDARD
BITPIX	32	
NAXIS	2	2D Image
NAXIS1	768	pixels in alpha
NAXIS2	768	pixels in delta
PCOUNT	0	(S)
GCOUNT	1	(S)
TFIELDS	1	(S)
BSCALE	1.000000000000E-01	
BZERO	0.000000000000E+00	
2MURBIAS	100525225	
2MUBOFF	0	Field median
CTYPE1	RA---TAN	coordinate type of x axis
CTYPE2	DEC--TAN	coordinate type of y axis
CDELTA1	-2.713230035239E-04	
CDELTA2	-2.707240623337E-04	
CRPIX1	384.0000000000	Reference X pixel
CRPIX2	384.0000000000	Reference Y pixel
INSTRUME	CAMI	
EXPOTIME	9	exposure time (in seconds)
ORIGIN	DeNIS	project origin of data
TELESCOP	ESO 1m/LaSilla	telescope used for observations
OBSERVER	lc + gs	astronomer leading the observations
EQUINOX	2000.	equinox used for coordinates
DATE-OBS	27/03/1996	date of observation in the form DD/MM/YY
2MUSOFTR	3.1	RTU Version
2MUTICMP	0	Zero time diff forced
2MUBAND	I	Band
2MUTYPE	NORMAL	
2MUPROCS	SFCX	
2MUCACHE	0c	CPU cache control
2MUHNA00	CAMI TEMP(0)	HSK Name
2MUHPH00	101.2644042969	HSK Value
2MUHBI00	572	HSK Measure
2MUHST00	DEFINED	HSK Status

2MUHNA01	CAMI TEMP(1)	HSK Name
2MUHPH01	21.4289855957	HSK Value
2MUHBI01	6	HSK Measure
2MUHST01	DEFINED	HSK Status
2MUHNA02	CAMI BIAS	HSK Name
2MUHPH02	0.0000000000	HSK Value
2MUHBI02	0	HSK Measure
2MUHST02	UNDEFINED	HSK Status
2MUIDENT	REGULAR	type of field identifier
2MUFIELD	83	position in the strip
2MUNUMBR	466899	running number of the field
CRVAL1	1.900014982379E+02	
CRVAL2	-1.166244239762E+01	
2MUNIGHT	2208	running Nr of project nights
2MUSOFTE	1.0b	(S)
2MUSOFTD	2.6	(S)
2MURDARK	100525176	
2MUSTRIP	3777	running number of the strip
2MUSTEPX	3	nb. of steps along x
2MUSTEPY	3	nb. of steps along y
2MUSOFFX	0.	x offset of the microscanning center
2MUSOFFY	0.	y offset of the microscanning center
2MUSOFTN	2.51	software version of the NRTU
2MUDSTX	3	denominator of the fraction of pixel step x
2MUDSTY	3	denominator of the fraction of pixel step y
2MUNSTX	1	numerator of the fraction of pixels along x
2MUNSTY	7	numerator of the fraction of pixels along y
2MUSTEPO	012345678	steps order of microscanning (hexadecimal)
2MUUTIME	14674.	universal time of observation
2MULST	42068.78906	local sidereal time of observation
2MUAIMR	1.080948949	airmass of observation
2MUFOCUS	-289.	position of secondary mirror
2MUDOME	223.	dome azimuth
2MUDOMN	T	dome mode: manual T, automatic F
2MUTELM	T	telescope tracking: on T, off F
2MURCC	466806	number of image used for positioning
2MUZENIT	22.29798508	telescope zenith
2MUAZIMU	220.9624786	telescope azimuth
MEANFLAT	1.000008535824E+00	
STDVFLAT	3.398779421819E-02	
MDEVSTRI	1.137301477103E+01	
DEVOFDEV	8.314030584858E-01	
MEANBIAS	4.644061710770E-02	
MDEVBIAS	1.219394083348E+02	

STDEVIMA	5.350008010864E+00	
BIAS_OFF	4.573712921143E+01	
SKY_OFF	4.573899841309E+01	
QUAD_OF1	-2.396369818598E-03	
QUAD_SD1	6.379959106445E+00	
QUAD_OF2	-2.950940397568E-04	
QUAD_SD2	6.374459266663E+00	
QUAD_OF3	-3.423852380365E-03	
QUAD_SD3	4.507073879242E+00	
QUAD_OF4	-1.356732915156E-03	
QUAD_SD4	4.138539791107E+00	
2MURFLAT	100525182	
2MURCOMP	100525182	
2MURFLAG	100525224	
2MUSOFTP	7.1.4	
CROTA1	-0.659983277	
CROTA2	-0.659983277	
FWHMX11	2.236000061035E+00	
FWHMX21	2.242000102997E+00	
FWHMX31	2.861999988556E+00	
FWHMX12	2.352999925613E+00	
FWHMX22	2.283999919891E+00	
FWHMX32	2.841000080109E+00	
FWHMX13	2.506000041962E+00	
FWHMX23	2.338999986649E+00	
FWHMX33	2.740999937057E+00	
FWHMY11	2.454999923706E+00	
FWHMY21	2.171999931335E+00	
FWHMY31	2.486999988556E+00	
FWHMY12	2.404999971390E+00	
FWHMY22	2.053999900818E+00	
FWHMY32	2.308000087738E+00	
FWHMY13	2.622999906540E+00	
FWHMY23	2.214999914169E+00	
FWHMY33	2.438999891281E+00	
NAMAXPOS	spatial	
UCDAXPOS	pos	
UNIAXPOS	deg	
INDAXPOS	TRUE	
BINAXPOS	768 768	
SYSAXPOS	FK5	
STEAXPOS	Unknown	
SYEAXPOS	Unknown	
CALAXPOS	CALIBRATED	

UNDAXPOS	FALSE	
REGAXPOS	TRUE	
POSITRA	190.0014982	
POSITDEC	-11.662442398	
LOLEFBOX	189.899461107855- 11.76640043755614	
UPRIGBOX	190.103535367945- 11.55848435768386	
SEEING	Unknown	
SCALEPIX	0.000271323003523900 0.000270724062333700	= '0.000271323003523900 0.000270724062333700'/
NAMAXTIM	time	
UCDAXTIM	time	
UNIAXTIM	day	
INDAXTIM	TRUE	
BINAXTIM	1	
SYSAXTIM	TT-ICRS-WAVELENGTH-TOPO	= 'TT-ICRS-WAVELENGTH-TOPO'/
STEAXTIM	Unknown	
SYEAXTIM	Unknown	
CALAXTIM	CALIBRATED	
UNDAXTIM	TRUE	
REGAXTIM	FALSE	
OBS-TIME	1996-03-27	
OBS-BEGI	Unknown	
OBS-END	Unknown	
TIMERESO	Unknown	
TIMESTEP	Unknown	
NAMAXWL	Spectral	
UCDAXWL	em	
UNIAXWL	mu	
INDAXWL	TRUE	
BINAXWL	1	
SYSAXWL	TT-ICRS-WAVELENGTH-TOPO	= 'TT-ICRS-WAVELENGTH-TOPO'/
STEAXWL	Unknown	
SYEAXWL	Unknown	
CALAXWL	CALIBRATED	
UNDAXWL	TRUE	
REGAXWL	FALSE	
MID-WL	0.8	
MIN-WL	0.7	
MAX-WL	0.9	
WL--RESO	Unknown	
WL--SAMP	Unknown	
END		

- **Le fichier de mapping : DenisSpe.map**

```

WL--SAMP + pectralAxis.samplingPrecision.samplingPrecisionRefVal.SamplingPeriod
NAMAXPOS + SpatialAxis.AxisName
UCDAXPOS + SpatialAxis.ucd
UNIAXPOS + SpatialAxis.unit
INDAXPOS + SpatialAxis.independentaxis
BINAXPOS + SpatialAxis.numBins
SYSAXPOS + SpatialAxis.coordsystem
STEAXPOS + SpatialAxis.accuracy.statError.ErrorRefVal.ErrorRefValue
SYEAXPOS + SpatialAxis.accuracy.sysError.ErrorRefVal.ErrorRefValue
CALAXPOS + SpatialAxis.calibrationStatus
UNDAXPOS + SpatialAxis.undersamplingStatus
REGAXPOS + SpatialAxis.regularsamplingStatus
POSITRA + SpatialAxis.coverage.location.coord.Position2D.Value2.C1
POSITDEC + SpatialAxis.coverage.location.coord.Position2D.Value2.C2
LOLEFBOX + SpatialAxis.coverage.bounds.limits.Coord2VecInterval.LoLimit2Vec
UPRIGBOX + SpatialAxis.coverage.bounds.limits.Coord2VecInterval.HiLimit2Vec
SEEING + SpatialAxis.resolution.resolutionRefVal
SCALEPIX + SpatialAxis.samplingPrecision.samplingPrecisionRefVal.SamplingPeriod
NAMAXTIM+ TimeAxis.AxisName
UCDAXTIM + TimeAxis.ucd
UNIAXTIM + TimeAxis.unit
INDAXTIM + TimeAxis.independentaxis
BINAXTIM + TimeAxis.numBins
SYSAXTIM + TimeAxis.coordsystem
STEAXTIM + TimeAxis.accuracy.statError.ErrorRefVal.ErrorRefValue
SYEAXTIM + TimeAxis.accuracy.sysError.ErrorRefVal.ErrorRefValue
CALAXTIM + TimeAxis.calibrationStatus
UNDAXTIM + TimeAxis.undersamplingStatus
REGAXTIM + TimeAxis.regularsamplingStatus
OBS-TIME + TimeAxis.coverage.location.coord.Time.TimeInstant
OBS-BEGI + TimeAxis.coverage.bounds.limits.CoordScalarInterval.LoLimit
OBS-END + TimeAxis.coverage.bounds.limits.CoordScalarInterval.HiLimit
TIMERESO + TimeAxis.resolution.resolutionRefVal
TIMESTEP + TimeAxis.samplingPrecision.samplingPrecisionRefVal.SamplingPeriod
  
```

- **Le fichier Uty : I0466899.uty**

```

%CharacterisationAxis 1
%SpatialAxis.AxisName spatial
%SpatialAxis.independentaxis TRUE
%SpatialAxis.calibrationStatus CALIBRATED
%SpatialAxis.unit deg
%SpatialAxis.coverage.bounds.limits.Coord2VecInterval.LoLimit2Vec (189.899461107855,-
11.76640043755614)
%SpatialAxis.undersamplingStatus FALSE
%SpatialAxis.coordsystem FK5
%SpatialAxis.ucd pos
%SpatialAxis.numBins (768,768)
%SpatialAxis.regularsamplingStatus TRUE
%SpatialAxis.coverage.bounds.limits.Coord2VecInterval.HiLimit2Vec (190.103535367945,-
11.55848435768386)
%SpatialAxis.samplingPrecision.samplingPrecisionRefVal.SamplingPeriod
(0.000271323003523900,0.000270724062333700)
  
```

```
%CharacterisationAxis 1
%SpatialAxis.AxisName spatial
%SpatialAxis.independentaxis TRUE
%SpatialAxis.calibrationStatus CALIBRATED
%SpatialAxis.unit deg
%SpatialAxis.coverage.bounds.limits.Coord2VecInterval.LoLimit2Vec (189.899461107855,-
11.76640043755614)
%SpatialAxis.undersamplingStatus FALSE
%SpatialAxis.coordsystem FK5
%SpatialAxis.ucd pos
%SpatialAxis.numBins (768,768)
%SpatialAxis.regularsamplingStatus TRUE
%SpatialAxis.coverage.bounds.limits.Coord2VecInterval.HiLimit2Vec (190.103535367945,-
11.55848435768386)
%SpatialAxis.samplingPrecision.samplingPrecisionRefVal.SamplingPeriod
(0.000271323003523900,0.000270724062333700)
%SpatialAxis.coverage.location.coord.Position2D.Value2.C1 190.0014982
%SpatialAxis.coverage.location.coord.Position2D.Value2.C2 -11.662442398

%CharacterisationAxis 2
%TimeAxis.AxisName time
%TimeAxis.undersamplingStatus TRUE
%TimeAxis.coordsystem TT-ICRS-WAVELENGTH-TOPO
%TimeAxis.numBins 1
%TimeAxis.independentaxis TRUE
%TimeAxis.coverage.location.coord.Time.TimeInstant 1996-03-27
%TimeAxis.ucd time
%TimeAxis.calibrationStatus CALIBRATED
%TimeAxis.unit day
%TimeAxis.regularsamplingStatus FALSE

%CharacterisationAxis 3
%SpectralAxis.AxisName Spectral
%SpectralAxis.coverage.bounds.limits.CoordScalarInterval.LoLimit 0.7
%SpectralAxis.undersamplingStatus TRUE
%SpectralAxis.coordsystem TT-ICRS-WAVELENGTH-TOPO
%SpectralAxis.calibrationStatus CALIBRATED
%SpectralAxis.coverage.bounds.limits.CoordScalarInterval.HiLimit 0.9
%SpectralAxis.regularsamplingStatus FALSE
%SpectralAxis.independentaxis TRUE
%SpectralAxis.ucd em
%SpectralAxis.unit mu
%SpectralAxis.coverage.location.coord.Spectral.Value 0.8
%SpectralAxis.numBins 1
```

- **Le fichier XML : I0466899.xml**

```
<?xml version="1.0" encoding="UTF-8"?>
<char:characterisation
xmlns:char="http://www.ivoa.net/xml/Characterisation/Characterisation-v1.11.xsd">
  <char:characterisationAxis>
    <char:axisName>spatial</char:axisName>
    <char:ucd>pos</char:ucd>
    <char:unit>deg</char:unit>
    <char:independentAxis>true</char:independentAxis>
    <char:calibrationStatus>CALIBRATED</char:calibrationStatus>
    <char:numBins2>
      <char:I1>768</char:I1>
      <char:I2>768</char:I2>
    </char:numBins2>
    <char:undersamplingStatus>false</char:undersamplingStatus>
    <char:regularsamplingStatus>true</char:regularsamplingStatus>
    <char:coverage>
      <char:location>
        <char:coord>
          <stc:Position2D xsi:type="stc:posVector2CoordinateType"
xmlns:stc="http://www.ivoa.net/xml/STC/stc-v1.30.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
            <stc:Name1>lon</stc:Name1>
            <stc:Name2>lat</stc:Name2>
            <stc:Value2 xsi:type="stc:double2Type">
              <stc:C1>190.0014982</stc:C1>
              <stc:C2>-11.662442398</stc:C2>
            </stc:Value2>
          </stc:Position2D>
        </char:coord>
      </char:location>
      <char:bounds>
        <char:limits>
          <char:Coord2VecInterval xsi:type="stc:coord2VecIntervalType"
xmlns:stc="http://www.ivoa.net/xml/STC/stc-v1.30.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
            <stc:LoLimit2Vec>
              <stc:C1>189.899461107855</stc:C1>
              <stc:C2>-11.76640043755614</stc:C2>
            </stc:LoLimit2Vec>
            <stc:HiLimit2Vec>
              <stc:C1>190.103535367945</stc:C1>
              <stc:C2>-11.55848435768386</stc:C2>
            </stc:HiLimit2Vec>
          </char:Coord2VecInterval>
        </char:limits>
      </char:bounds>
    </char:coverage>
  </char:characterisationAxis>
</char:characterisationAxis>
```

```

<char:ucd>time</char:ucd>
  <char:unit>day</char:unit>
  <char:independentAxis>true</char:independentAxis>
  <char:calibrationStatus>CALIBRATED</char:calibrationStatus>
  <char:numBins1>1</char:numBins1>
  <char:undersamplingStatus>true</char:undersamplingStatus>
  <char:regularsamplingStatus>>false</char:regularsamplingStatus>
  <char:coverage>
    <char:location>
      <char:coord/>
    </char:location>
  </char:coverage>
</char:characterisationAxis>
<char:characterisationAxis>
  <char:axisName>Spectral</char:axisName>
  <char:ucd>em</char:ucd>
  <char:unit>mu</char:unit>
  <char:independentAxis>true</char:independentAxis>
  <char:calibrationStatus>CALIBRATED</char:calibrationStatus>
  <char:numBins1>1</char:numBins1>
  <char:undersamplingStatus>true</char:undersamplingStatus>
  <char:regularsamplingStatus>>false</char:regularsamplingStatus>
  <char:coverage>
    <char:bounds>
      <char:limits>
        <char:CoordScalarInterval xsi:type="stc:coordScalarIntervalType"
xmlns:stc="http://www.ivoa.net/xml/STC/stc-v1.30.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
          <stc:LoLimit>0.7</stc:LoLimit>
          <stc:HiLimit>0.9</stc:HiLimit>
        </char:CoordScalarInterval>
      </char:limits>
    </char:bounds>
  </char:coverage>
</char:characterisationAxis>
</char:characterisation>

```

### **Annexe n° 05 : Tutorial portage du serveur AIDA sous TOMCAT**

Tomcat est essentiellement indiqué pour être un conteneur de servlet/JSP, mais il est doté de beaucoup de possibilités lui permettant de rivaliser avec des serveurs web traditionnels. Une de ces possibilités est le support des Common Gateway Interface (CGI) qui apportent le moyen de lancer un programme externe en réponse à une requête de navigateur, habituellement pour traiter un formulaire web. CGI est appelé "commun" parce qu'il peut invoquer des programmes dans presque tous les langages de programmation ou de scripting : Perl, Python, awk, scripting de shell Unix et même Java sont tous des options supportées.

Afin de mettre en place le portage du serveur Aida sous TOMCAT, il y a quelques modifications et paramétrages à apporter au serveur TOMCAT ainsi qu'au client AIDA.

### ➤ Modifications coté serveur TOMCAT

**TOMCAT** comprend une servlet **CGI** optionnelle qui permet de lancer d'anciens scripts **CGI**, pour l'activer, il faut suivre les étapes ci-dessous :

1. Renommez le fichier `servlets-cgi.renametojar` (situé dans `CATALINA_HOME/server/lib/`) en `servletscgi.jar`, de façon à ce que la servlet qui traitera les scripts **CGI** soit dans le CLASSPATH de **TOMCAT**.
2. Dans le fichier `CATALINA_BASE/conf/web.xml` de **TOMCAT**, enlevez les commentaires sur la définition de la servlet nommée **cgi**.

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

3. Toujours dans le fichier `web.xml`, décommentez la redirection de servlet pour la servlet `cgi`.

```
<!-- The mapping for the default servlet -->
<servlet-mapping>
  <servlet-name>default</servlet-name>
  <url-pattern>/</url-pattern>
</servlet-mapping>
```

4. Placez les scripts **CGI** soit dans le répertoire `WEB-INF/cgi` directory, soit dans un autre répertoire de votre contexte. Pour notre application Aida, on va opter pour la seconde alternative, ainsi on choisira comme répertoire d'accueil de nos scripts : `cgi-bin` directory, il faut ensuite ajuster le paramètre d'initialisation

cgiPathPrefix du GIServlet de façon à identifier le nouveau répertoire contenant les fichiers. Pour le faire revenez au fichier web.xml, plus particulièrement à la partie qui concerne la définition de la servlet cgi (étape 2) et remplacez la valeur du paramètre cgiPathPrefix (WEB-INF/CGI) par cgi-bin.

Avant modification :

```
<init-param>
  <param-name>cgiPathPrefix</param-name>
  <param-value>WEB-INF/CGI </param-value>
</init-param>
```

Après modification :

```
<init-param>
  <param-name>cgiPathPrefix</param-name>
  <param-value> cgi-bin</param-value>
</init-param>
```

5. Relancez Tomcat, et votre processus CGI devrait être maintenant opérationnel.
6. Déployer sous TOMCAT le serveur Aida de la façon suivante :
  - Context Path /aida(sous windows) ou /~aida(sous Linux)
  - Directory URL windowsPath\aida\public\_html(sous windows)  
linuxPath\~aida\public\_html(sous linux)

#### ➤ **Modifications coté client Aida**

Au fichier de configuration conf.properties ajouter le numéro du port de **TOMCAT** (port par défaut 8080) à l'adresse du serveur aida à utiliser.

```
# server to be used:
server=adresseServeurUtiliser:TomcatPort
```

#### **Exemple**

Ancienne valeur :  
# server to be used:  
server=localhost

Nouvelle valeur :  
# server to be used:  
server=localhost:8080

## Annexe n° 06 : Schéma du modèle de la Characterisation XML

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSPY v5 rel. 4 U (http://www.xmlspy.com) by louys
(observatoire astronomique) -->
<!-- edited with XMLSpy v2005 rel. 3 U (http://www.altova.com) by bonnarel
(CDS) -->
<!-- W3C Schema for Observation Characterization
0.0 First draft by M.Louys and F.Bonnarel 2004 July and August
0.1 Horribly messed up by J. McDowell 2004 Sep
Version 0.2 2004 Sep 22: fixed Bounds mistake
Version 0.3 2005 Mar 31: F.Bonnarel changes to be consistent with
draft sent the same day, and new Mireille Louys' UML diagrams
Version 0.31 2005 APRIL 15 F.bonnarel: corrected bugs about attributes in
Frame
Version 0.4 2005 may F.Bonnarel based on BThomas STC and Qty
Version 0.5 2005 sep F.bonnarel based on A Rots STC
Version 0.9 2005 oct F.Bonnarel based on A Rots STC and consistent with
char Working draft
Version 0.91 2005 oct F.Bonnarel 0.9 + subgroup discussion consistency
modifications
Version 0.92 2005 nov F.Bonnarel 0.9 + document compatible
Version 0.93 2006 may F.Bonnarel Compatibility with new draft
new Bounds, Resolution and Fill factor definition
Version 0.94 2006 Aug 18 F.Bonnarel changes for STC 1.3
Version 0.95 2006 Sep 13 F.Bonnarel Bux fixes for compatibility with STC
1.3 fully validated version (XML spy)
Version 0.96 2006 Sep 30 F.Bonnarel + M.Louys: cleaning, documenting
update of MinMaxOccurs with respect to
the discussion on mandatory status of elements MUST/SHOULD MAY
creation of Specific axis as restrictions. StaError and SysError.
Version 0.97 2006 Oct 05 F.B slight modifications for first document
validation
Version 1.0 2006 Oct 16 Working draft compatible version
2006 Nov. 06: with annotation and documentation files included ML
Version 1.10 2007 Mar 26 FB Harmonisation with Spectrum DM
01 April 2007: moved annotation in types definitions of SpatialAxis, etc... ML
Version 1.11 May 22 2007 FB reusing stc subst group for value, size, error
resolution, pixsize ....
-->
<xsd:schema
targetNamespace="http://www.ivoa.net/xml/Characterisation/Characterisation-
v1.11.xsd" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:stc="http://www.ivoa.net/xml/STC/stc-v1.30.xsd"
xmlns:cha="http://www.ivoa.net/xml/Characterisation/Characterisation-v1.11.xsd"
elementFormDefault="qualified">
<xsd:import namespace="http://www.ivoa.net/xml/STC/stc-v1.30.xsd"
schemaLocation="http://hea-www.harvard.edu/~arots/nvmeta/v1.30/stc-
v1.30.xsd"/>
<xsd:element name="characterizationAxis"
type="cha:CharacterisationAxisType"
substitutionGroup="cha:characterisationAxis">
<xsd:annotation>
<xsd:documentation>
describes an axis and its properties
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="characterisation"
type="cha:CharacterisationType"/>
<!-- http://hea-www.harvard.edu/~arots/nvmeta/v1.30/stc-v1.30.xsd' / -
-->
<!-- Characterization is the root element -->
<xsd:element name="characterization"
type="cha:CharacterisationType" substitutionGroup="cha:characterisation"/>
<xsd:complexType name="CharacterisationType">
<xsd:choice>
<xsd:element name="characterisationAxis"
type="cha:CharacterisationAxisType" maxOccurs="unbounded"/>
</xsd:choice>
</xsd:complexType>
<!-- Characterization, for one axis -->
<xsd:element name="characterisationAxis"
type="cha:CharacterisationAxisType">
<xsd:annotation>
<xsd:documentation>
```

```
describes an axis and its properties
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<!-- For Americans -->
<xsd:element name="coverage" type="cha:CoverageType"/>
<xsd:element name="resolution" type="cha:ResolutionType">
<xsd:annotation>
<xsd:documentation>
minimum size of an interpretable signal along this axis
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="location" type="cha:LocationType"/>
<xsd:element name="bounds" type="cha:BoundsType"/>
<xsd:element name="support" type="cha:SupportType"/>
<xsd:element name="sensitivity" type="cha:SensitivityType"/>
<xsd:element name="resolutionRefVal"
type="cha:ResolutionRefValType"/>
<xsd:element name="samplingPrecisionRefVal"
type="cha:SamplingPrecisionRefValType"/>
<xsd:complexType name="anyURIType">
<xsd:complexContent>
<xsd:restriction base="xsd:anyType"/>
</xsd:complexContent>
</xsd:complexType>
<!-- COVERAGE can contain Location, Bounds, Support, Sensitivity -->
<!-- coordsystem and unit can be redefined at any level if necessary
otherwise the valid coordsys and
unit are defined on the AxisFrame -->
<xsd:complexType name="CoverageType">
<xsd:annotation>
<xsd:documentation>
How the observation is spanned along this axis
</xsd:documentation>
</xsd:annotation>
<xsd:sequence>
<xsd:element name="unit" type="xsd:anyType"
minOccurs="0">
<xsd:annotation>
<xsd:documentation>redefinition of unit for
coverage</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="coordsystem"
type="stc:astroCoordSystemType" minOccurs="0">
<xsd:annotation>
<xsd:documentation>redefinition of coordsystem</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="location"
type="cha:LocationType">
<xsd:annotation>
<xsd:documentation>Typical
value on one axis</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="bounds"
type="cha:BoundsType" minOccurs="0">
<xsd:annotation>
<xsd:documentation>the limits
of the observation on this axis</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="support"
type="cha:SupportType" minOccurs="0">
<xsd:annotation>
<xsd:documentation>describes
the area where measurements are effectively present and interpretable
</xsd:documentation>
</xsd:annotation>
</xsd:element>
```

<pre>                 &lt;xsd:element name="sensitivity" type="cha:SensitivityType" minOccurs="0"/&gt;                 &lt;xsd:annotation&gt;                     &lt;xsd:documentation&gt;encodes the variability of response along the axis &lt;/xsd:documentation&gt;                 &lt;/xsd:annotation&gt;                 &lt;/xsd:element&gt;             &lt;/xsd:sequence&gt;         &lt;/xsd:complexType&gt;         &lt;xsd:complexType name="ResolutionType"&gt;             &lt;xsd:sequence&gt;                 &lt;xsd:element name="unit" type="xsd:anyType" minOccurs="0"/&gt;                 &lt;xsd:element name="coordsystem" type="stc:astroCoordSystemType" minOccurs="0"/&gt;                 &lt;xsd:element name="resolutionRefVal" type="cha:ResolutionRefValType"/&gt;                 &lt;xsd:element name="resolutionBounds" type="cha:ResolutionBoundsType" minOccurs="0"/&gt;                 &lt;xsd:element name="resolutionSupport" type="cha:ResolutionSupportType" minOccurs="0"/&gt;                 &lt;xsd:element name="resolutionVariability" type="cha:ResolutionVariabilityType" minOccurs="0"/&gt;             &lt;/xsd:sequence&gt;         &lt;/xsd:complexType&gt;         &lt;xsd:complexType name="SamplingPrecisionType"&gt;             &lt;xsd:sequence&gt;                 &lt;xsd:element name="unit" type="xsd:anyType" minOccurs="0"/&gt;                 &lt;xsd:element name="coordsystem" type="stc:astroCoordSystemType" minOccurs="0"/&gt;                 &lt;xsd:element name="samplingPrecisionRefVal" type="cha:SamplingPrecisionRefValType"/&gt;                 &lt;xsd:element name="samplingPrecisionBounds" type="cha:SamplingPrecisionBoundsType" minOccurs="0"/&gt;                 &lt;xsd:element name="samplingPrecisionSupport" type="cha:SamplingPrecisionSupportType" minOccurs="0"/&gt;                 &lt;xsd:element name="samplingPrecisionVariability" type="cha:SamplingPrecisionVariabilityType" minOccurs="0"/&gt;             &lt;/xsd:sequence&gt;         &lt;/xsd:complexType&gt;         &lt;xsd:complexType name="CharacterisationAxisType"&gt;             &lt;xsd:sequence&gt;                 &lt;xsd:element name="axisName" type="xsd:anyType"/&gt;             &lt;/xsd:sequence&gt;         &lt;/xsd:complexType&gt;         &lt;xsd:annotation&gt;             &lt;xsd:documentation&gt;                 A name representing the physical nature of the axis like                 Spatial, Spectral, Time, Velocity or any other free name for an axis             &lt;/xsd:documentation&gt;         &lt;/xsd:annotation&gt;         &lt;/xsd:element&gt;         &lt;xsd:element name="ucd" type="xsd:anyType"&gt;             &lt;xsd:annotation&gt;                 &lt;xsd:documentation&gt;                     The physical meaning of the axis,                     Type: standard semantic tag defined in the UCD                 &lt;/xsd:documentation&gt;             &lt;/xsd:annotation&gt;         &lt;/xsd:element&gt;         &lt;xsd:element name="unit" type="xsd:anyType"&gt;             &lt;xsd:annotation&gt;                 &lt;xsd:documentation&gt;                     default unit for the axis:                     Type: controlled unit vocabulary following the VOTable units                 &lt;/xsd:documentation&gt;             &lt;/xsd:annotation&gt;         &lt;/xsd:element&gt;         &lt;xsd:element name="coordsystem" type="stc:astroCoordSystemType"&gt;             &lt;xsd:annotation&gt; </pre>	<pre> &lt;xsd:documentation&gt;                 Reference coordinate                 system for the axis:             &lt;/xsd:documentation&gt;         &lt;/xsd:annotation&gt;         &lt;/xsd:element&gt;         &lt;xsd:element name="ObsyLoc" type="stc:observatoryLocationType" minOccurs="0"/&gt;             &lt;xsd:annotation&gt;                 &lt;xsd:documentation&gt;                     Observatory                 &lt;/xsd:documentation&gt;             &lt;/xsd:annotation&gt;             location                 Type: a                 STC:observatoryLocationType or an xlink             &lt;/xsd:documentation&gt;         &lt;/xsd:annotation&gt;         &lt;/xsd:element&gt;         &lt;xsd:element name="accuracy" type="cha:AccuracyType" minOccurs="0"/&gt;             &lt;xsd:annotation&gt;                 &lt;xsd:documentation&gt;Global             &lt;/xsd:documentation&gt;         &lt;/xsd:annotation&gt;         accuracy description of the axis             &lt;/xsd:documentation&gt;         &lt;/xsd:annotation&gt;         &lt;/xsd:element&gt;         &lt;xsd:element name="independentAxis" type="xsd:anyType" minOccurs="0"/&gt;             &lt;xsd:annotation&gt;                 &lt;xsd:documentation&gt;                     Indicates wether the axis is independent                     or not from the other axes. The Observable axis                     has a false independentAxis "value"                     Type : boolean                 &lt;/xsd:documentation&gt;             &lt;/xsd:annotation&gt;         &lt;/xsd:element&gt;         &lt;xsd:element name="calibrationStatus" type="xsd:anyType"&gt;             &lt;xsd:annotation&gt;                 &lt;xsd:documentation&gt;                     defines if and how the axis is                     calibrated:                     Possible String values: CALIBRATED, UNCALIBRATED,                     RELATIVE, NORMALIZED                 &lt;/xsd:documentation&gt;             &lt;/xsd:annotation&gt;         &lt;/xsd:element&gt;         &lt;xsd:element ref="cha:numBins"&gt;             &lt;xsd:annotation&gt;                 &lt;xsd:documentation&gt;                     Number of bins for                     this axis.                     Type : array of 1 2                     or 3 integers                 &lt;/xsd:documentation&gt;             &lt;/xsd:annotation&gt;         &lt;/xsd:element&gt;         &lt;xsd:element name="undersamplingStatus" type="xsd:anyType" minOccurs="0"/&gt;             &lt;xsd:annotation&gt;                 &lt;xsd:documentation&gt;                     A flag to mention wether                     the data are undersampled.                     Type: boolean                 &lt;/xsd:documentation&gt;             &lt;/xsd:annotation&gt;         &lt;/xsd:element&gt;         &lt;xsd:element name="regularsamplingStatus" type="xsd:anyType" minOccurs="0"/&gt;             &lt;xsd:annotation&gt;                 &lt;xsd:documentation&gt;                     A flag to mention wether the                     data are regularly sampled.                     Type : boolean                 &lt;/xsd:documentation&gt;             &lt;/xsd:annotation&gt; </pre>
---	---

<pre> &lt;/xsd:element&gt;     &lt;xsd:element name="coverage" type="cha:CoverageType"/&gt;     &lt;xsd:element name="resolution" type="cha:ResolutionType" minOccurs="0"/&gt;     &lt;xsd:element name="samplingPrecision" type="cha:SamplingPrecisionType" minOccurs="0"/&gt;     &lt;/xsd:sequence&gt; &lt;/xsd:complexType&gt; &lt;xsd:complexType name="SpatialAxisType"&gt;     &lt;xsd:annotation&gt;     &lt;xsd:documentation&gt;         Restriction of CharacterisationAxis for Spatial Axis     &lt;/xsd:documentation&gt;     &lt;/xsd:annotation&gt;     &lt;xsd:complexContent&gt;     &lt;xsd:restriction base="cha:CharacterisationAxisType"&gt;     &lt;xsd:sequence&gt;     &lt;xsd:element name="axisName" type="xsd:anyType" fixed="spatial"/&gt;     &lt;xsd:element name="ucd" type="xsd:anyType" fixed="pos"/&gt;     &lt;xsd:element name="unit" type="xsd:anyType"/&gt;     &lt;xsd:element name="coordsystem" type="stc:astroCoordSystemType"/&gt;     &lt;xsd:element name="ObsyLoc" type="stc:observatoryLocationType" minOccurs="0"/&gt;     &lt;xsd:element name="accuracy" type="cha:AccuracyType" minOccurs="0"/&gt;     &lt;xsd:element name="independentAxis" type="xsd:anyType" minOccurs="0"/&gt;     &lt;xsd:element name="calibrationStatus" type="xsd:anyType"/&gt;     &lt;xsd:element ref="cha:numBins"/&gt;     &lt;xsd:element name="undersamplingStatus" type="xsd:anyType" minOccurs="0"/&gt;     &lt;xsd:element name="regularsamplingStatus" type="xsd:anyType" minOccurs="0"/&gt;     &lt;xsd:element name="coverage" type="cha:CoverageType"/&gt;     &lt;xsd:element name="resolution" type="cha:ResolutionType" minOccurs="0"/&gt;     &lt;xsd:element name="samplingPrecision" type="cha:SamplingPrecisionType" minOccurs="0"/&gt;     &lt;/xsd:sequence&gt;     &lt;/xsd:restriction&gt;     &lt;/xsd:complexContent&gt; &lt;/xsd:complexType&gt; &lt;xsd:element name="spectralAxis" type="cha:SpatialAxisType"/&gt; &lt;xsd:complexType name="SpectralAxisType"&gt;     &lt;xsd:annotation&gt;     &lt;xsd:documentation&gt;         Restriction of characterisationAxis for Spectral Axis     &lt;/xsd:documentation&gt;     &lt;/xsd:annotation&gt;     &lt;xsd:complexContent&gt;     &lt;xsd:restriction base="cha:CharacterisationAxisType"&gt;     &lt;xsd:sequence&gt;     &lt;xsd:element name="axisName" type="xsd:anyType" fixed="spectral"/&gt;     &lt;xsd:element name="ucd" type="xsd:anyType" fixed="em"/&gt;     &lt;xsd:element name="unit" type="xsd:anyType"/&gt;     &lt;xsd:element name="coordsystem" type="stc:astroCoordSystemType"/&gt;     &lt;xsd:element name="ObsyLoc" type="stc:observatoryLocationType" minOccurs="0"/&gt;     &lt;xsd:element name="accuracy" type="cha:AccuracyType" minOccurs="0"/&gt; </pre>	<pre> name="independentAxis" type="xsd:anyType" minOccurs="0"/&gt;     &lt;xsd:element name="calibrationStatus" type="xsd:anyType"/&gt;     &lt;xsd:element ref="cha:numBins"/&gt;     &lt;xsd:element name="undersamplingStatus" type="xsd:anyType" minOccurs="0"/&gt;     &lt;xsd:element name="regularsamplingStatus" type="xsd:anyType" minOccurs="0"/&gt;     &lt;xsd:element name="coverage" type="cha:CoverageType"/&gt;     &lt;xsd:element name="resolution" type="cha:ResolutionType" minOccurs="0"/&gt;     &lt;xsd:element name="samplingPrecision" type="cha:SamplingPrecisionType" minOccurs="0"/&gt;     &lt;/xsd:sequence&gt;     &lt;/xsd:restriction&gt;     &lt;/xsd:complexContent&gt; &lt;/xsd:complexType&gt; &lt;xsd:element name="spectralAxis" type="cha:SpectralAxisType"/&gt; &lt;xsd:complexType name="TimeAxisType"&gt;     &lt;xsd:annotation&gt;     &lt;xsd:documentation&gt;         Restriction of CharacterisationAxis for time     &lt;/xsd:documentation&gt;     &lt;/xsd:annotation&gt;     &lt;xsd:complexContent&gt;     &lt;xsd:restriction base="cha:CharacterisationAxisType"&gt;     &lt;xsd:sequence&gt;     &lt;xsd:element name="axisName" type="xsd:anyType" fixed="temporal"/&gt;     &lt;xsd:element name="ucd" type="xsd:anyType" fixed="time"/&gt;     &lt;xsd:element name="unit" type="xsd:anyType"/&gt;     &lt;xsd:element name="coordsystem" type="stc:astroCoordSystemType"/&gt;     &lt;xsd:element name="ObsyLoc" type="stc:observatoryLocationType" minOccurs="0"/&gt;     &lt;xsd:element name="accuracy" type="cha:AccuracyType" minOccurs="0"/&gt;     &lt;xsd:element name="independentAxis" type="xsd:anyType" minOccurs="0"/&gt;     &lt;xsd:element name="calibrationStatus" type="xsd:anyType"/&gt;     &lt;xsd:element ref="cha:numBins"/&gt;     &lt;xsd:element name="undersamplingStatus" type="xsd:anyType" minOccurs="0"/&gt;     &lt;xsd:element name="regularsamplingStatus" type="xsd:anyType" minOccurs="0"/&gt;     &lt;xsd:element name="coverage" type="cha:CoverageType"/&gt;     &lt;xsd:element name="resolution" type="cha:ResolutionType" minOccurs="0"/&gt;     &lt;xsd:element name="samplingPrecision" type="cha:SamplingPrecisionType" minOccurs="0"/&gt;     &lt;/xsd:sequence&gt;     &lt;/xsd:restriction&gt;     &lt;/xsd:complexContent&gt; &lt;/xsd:complexType&gt; &lt;xsd:element name="timeAxis" type="cha:TimeAxisType"/&gt; &lt;xsd:element name="numBins" type="xsd:anyType" abstract="true"/&gt; &lt;xsd:element name="numBins1" type="xsd:int" substitutionGroup="cha:numBins"/&gt; &lt;xsd:element name="numBins2" substitutionGroup="cha:numBins"&gt;     &lt;xsd:complexType&gt;     &lt;xsd:sequence&gt;     &lt;xsd:element name="I1" type="xsd:int"/&gt;     &lt;xsd:element name="I2" type="xsd:int"/&gt;     &lt;/xsd:sequence&gt;     &lt;/xsd:complexType&gt; &lt;/xsd:element&gt; </pre>
---	--

```

<xsd:element name="numBins3" substitutionGroup="cha:numBins">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="I1"
type="xsd:int"/>
      <xsd:element name="I2"
type="xsd:int"/>
      <xsd:element name="I3"
type="xsd:int"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:complexType name="AccuracyType">
  <xsd:sequence>
    <xsd:element name="quality" type="xsd:string"
minOccurs="0"/>
    <xsd:annotation>
      <xsd:documentation>A
combination of flags attesting the data quality.
      </xsd:documentation>
      Type: String
    </xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:element name="statError"
type="cha:StatErrorType" minOccurs="0"/>
  <xsd:element name="sysError"
type="cha:SysErrorType" minOccurs="0"/>
  <xsd:element name="CustError"
type="cha:ErrorType" minOccurs="0"/>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ErrorType">
  <xsd:sequence>
    <xsd:annotation>
      <xsd:documentation>
        Some Error on the axis
      </xsd:documentation>
    </xsd:annotation>
    <xsd:element name="flavor" type="xsd:string">
      <xsd:annotation>
        <xsd:documentation>
          The type of error described : statistical , systematic, global, etc.
          Type: string
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
</xsd:annotation>
  </xsd:element>
  <xsd:element name="ErrorRefVal"
type="cha:ErrorRefValType"/>
  <xsd:element name="ErrorBounds"
type="cha:ErrorBoundsType" minOccurs="0"/>
  <xsd:element name="ErrorVariability"
type="cha:ErrorVariabilityType" minOccurs="0"/>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="StatErrorType">
  <xsd:annotation>
    <xsd:documentation>Statistical Error on the axis
  </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:restriction base="cha:ErrorType">
      <xsd:sequence>
        <xsd:element
name="flavor" type="xsd:string" fixed="statistical"/>
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
  <xsd:annotation>
    <xsd:documentation>The type of error described : here statistical
    Type : string , value="statistical"
  </xsd:documentation>
  </xsd:annotation>
</xsd:complexType>
  </xsd:element>
  <xsd:element
name="ErrorRefVal" type="cha:ErrorRefValType"/>

```

```

<xsd:annotation>
  <xsd:documentation>Typical statistical Error on the
axis</xsd:documentation>
</xsd:annotation>
  </xsd:element>
  <xsd:element
name="ErrorBounds" type="cha:ErrorBoundsType" minOccurs="0"/>
</xsd:annotation>
  <xsd:documentation>Range of the error for mapping along this axis
</xsd:documentation>
</xsd:annotation>
  </xsd:element>
  <xsd:element
name="ErrorVariability" type="cha:ErrorVariabilityType" minOccurs="0"/>
</xsd:annotation>
  <xsd:documentation>
    A detailed description of the local error values
  </xsd:documentation>
  </xsd:annotation>
  </xsd:element>
  </xsd:sequence>
</xsd:restriction>
</xsd:complexType>
<xsd:complexType name="SysErrorType">
  <xsd:annotation>
    <xsd:documentation>
      Systematic Error on the axis
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:restriction base="cha:ErrorType">
      <xsd:sequence>
        <xsd:element name="flavor"
type="xsd:string" fixed="systematic"/>
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
  <xsd:documentation> The type of error described : here, systematic
  Type: string , value: systematic
</xsd:documentation>
</xsd:annotation>
  </xsd:element>
  <xsd:element
name="ErrorRefVal" type="cha:ErrorRefValType"/>
  <xsd:element
name="ErrorBounds" type="cha:ErrorBoundsType" minOccurs="0"/>
  <xsd:element
name="ErrorVariability" type="cha:ErrorVariabilityType" minOccurs="0"/>
</xsd:sequence>
</xsd:restriction>
</xsd:complexType>
<xsd:complexType name="ErrorRefValType">
  <xsd:sequence>
    <xsd:annotation>
      <xsd:documentation>
        Typical statistical error on the axis.
        Types: depend on the dimensionality
        1D Error based on stc:double1Type
        2D Errors based on stc:size2Type,
        double2Type or double1Type (error 2 radius)
        3D Errors based on stc:size3Type,
        double3Type or double1Type (error 3 radius)
      </xsd:documentation>
    </xsd:annotation>
    <xsd:element name="unit" type="xsd:anyType"
minOccurs="0"/>
    <xsd:element name="coordsystem"
type="stc:astroCoordSystemType" minOccurs="0"/>
    <xsd:element ref="stc:CError"/>
    <xsd:element name="documentation"
type="cha:anyURIType" minOccurs="0"/>
  </xsd:sequence>

```

```

</xsd:sequence>
  </xsd:complexType>
  <xsd:element name="ErrorRefValue" type="xsd:anyType"
abstract="true" substitutionGroup="stc:CError"/>
  <!-- <xsd:element name="Error" type="stc:double1Type"
substitutionGroup="cha:ErrorRefValue" nillable="true"/>
  <xsd:element name="CError2" type="xsd:anyType" abstract="true"
substitutionGroup="cha:ErrorRefValue" nillable="true"/>
  <xsd:element name="Error2" type="stc:size2Type"
substitutionGroup="cha:CError2" nillable="true"/>
  <xsd:element name="Error2Matrix" type="stc:double4Type"
substitutionGroup="cha:CError2" nillable="true"/>
  <xsd:element name="Error2Radius" type="stc:double1Type"
substitutionGroup="cha:CError2" nillable="true"/>
  <xsd:element name="CError3" type="xsd:anyType" abstract="true"
substitutionGroup="cha:ErrorRefValue" nillable="true"/>
  <xsd:element name="Error3" type="stc:size3Type"
substitutionGroup="cha:CError3" nillable="true"/>
  <xsd:element name="Error3Matrix" type="stc:double9Type"
substitutionGroup="cha:CError3" nillable="true"/>
  <xsd:element name="Error3Radius" type="stc:double1Type"
substitutionGroup="cha:CError3" nillable="true"/> -->
  <xsd:complexType name="ErrorBoundsType">
    <xsd:annotation>
      <xsd:documentation> Hi and Low values of
statistical errors on this axis.
      Type: stc:coordScalarInterval, 2DVecInterval or
3DVecInterval
    </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element ref="cha:ErrorLimits"/>
      <xsd:element name="documentation"
type="cha:anyURIType" minOccurs="0">
        <xsd:annotation>
          <xsd:documentation> A
description of the error extreme values .
          Type: any URI type
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name="ErrorLimits" type="stc:coordIntervalType"
abstract="true"/>
  <xsd:element name="ErrorLimits1" type="stc:coordScalarIntervalType"
substitutionGroup="cha:ErrorLimits"/>
  <xsd:element name="ErrorLimits2" type="stc:coord2VecIntervalType"
substitutionGroup="cha:ErrorLimits"/>
  <xsd:element name="ErrorLimits3" type="stc:coord3VecIntervalType"
substitutionGroup="cha:ErrorLimits"/>
  <xsd:complexType name="ErrorVariabilityType">
    <xsd:annotation>
      <xsd:documentation>A detailed description of the
local error values. Could be conveyed with
the data as in IVOA Spectrum data model
      Type: any URI type pointing to such a map
    </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element name="ErrorMap"
type="cha:anyURIType">
        <xsd:annotation>
          <xsd:documentation>A
map showing the variability of the error
          Type: any URI type
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
  <xsd:element name="documentation"
type="cha:anyURIType" minOccurs="0">
    <xsd:annotation>
      <xsd:documentation>documentation on the Error variation
      Type: any URI type
    </xsd:documentation>
  </xsd:annotation>

```

```

  </xsd:documentation>
    </xsd:annotation>
  </xsd:element>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="LocationType">
  <xsd:annotation>
    <xsd:documentation>The typical coordinate value
    Type: stc:astroCoordsType
  </xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="unit" type="xsd:anyType"
minOccurs="0">
      <xsd:annotation>
        <xsd:documentation>Redefinition of unit</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="coordsystem"
type="stc:astroCoordSystemType" minOccurs="0">
      <xsd:annotation>
        <xsd:documentation>Redefinition of coordsystem</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="coord"
type="stc:astroCoordsType" maxOccurs="unbounded"/>
    <xsd:element name="documentation"
type="cha:anyURIType" minOccurs="0">
      <xsd:annotation>
        <xsd:documentation>The
typical coordinate value
        Type: stc:astroCoordsType
      </xsd:documentation>
    </xsd:annotation>
  </xsd:element>
</xsd:sequence>
</xsd:complexType>
<!--for some regimes like radio, and some axis of the data (spatial, ...)
bounds may also be expressed in Fourier frequencies We probably need a
FrequencyBounds in addition to Bounds-->
  <!-- For each coordinate, the Bounds has one min and one max. The
CharCooodAreaType
lets you define boxes or open/closed intervals, unit, it's a pseudo restriction on
AstroCoordArea
-->
  <xsd:complexType name="BoundsType">
    <xsd:sequence>
      <xsd:element name="unit" type="xsd:anyType"
minOccurs="0">
        <xsd:annotation>
          <xsd:documentation>for
bounds unit redefinition</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="coordsystem"
type="stc:astroCoordSystemType" minOccurs="0">
        <xsd:annotation>
          <xsd:documentation>for
bounds coosys redefinition</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="Extent"
type="stc:double1Type" minOccurs="0"/>
      <xsd:element name="limits"
type="cha:CharCoordAreaType">
        <xsd:annotation>
          <xsd:documentation>
The actual values defining the bounds.
2 possible types : a cha:CharBox which is a customised
STC:Coordinate with mandatory value and size
or an STC:Interval
        </xsd:documentation>
      </xsd:annotation>
    </xsd:sequence>

```

```

</xsd:annotation>
    </xsd:element>
    <xsd:element name="documentation"
type="cha:anyURIType" minOccurs="0">
        <xsd:annotation>
            <xsd:documentation>
                A document to
                mention how the bounds are defined.
            </xsd:documentation>
        </xsd:annotation>
    </xsd:element>
</xsd:sequence>
</xsd:complexType>
<!-- We extend (instead of aggregate) coordArea to add simple
AreaType.
Really, Support is just an AstroCoordArea,
-->
    <xsd:complexType name="SupportType">
        <xsd:sequence>
            <xsd:element name="unit" type="xsd:anyType"
minOccurs="0"/>
            <xsd:element name="coordsystem"
type="stc:astroCoordSystemType" minOccurs="0"/>
            <xsd:element name="Extent"
type="stc:double1Type" minOccurs="0"/>
            <xsd:element name="Area"
type="stc:astroCoordAreaType">
                <xsd:annotation>
                    <xsd:documentation>
                        Defines the
                        effective covered region
                    </xsd:documentation>
                </xsd:annotation>
            </xsd:element>
            <xsd:element name="AreaType"
type="xsd:anyType">
                <xsd:annotation>
                    <xsd:documentation>
                        Gives the name of the region shape
                        predefined string in {Circle, Polygon, Box, ...}
                    </xsd:documentation>
                </xsd:annotation>
            </xsd:element>
            <xsd:element name="documentation"
type="cha:anyURIType" minOccurs="0">
                <xsd:annotation>
                    <xsd:documentation> Some
                    text about the Support region definition
                </xsd:documentation>
            </xsd:annotation>
            Type: any URI type
        </xsd:documentation>
    </xsd:annotation>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
<!-- URI will be external references (URL) -->
<xsd:complexType name="SensitivityType">
    <xsd:sequence>
        <xsd:element name="unit" type="xsd:anyType"
minOccurs="0"/>
        <xsd:element name="coordsystem"
type="stc:astroCoordSystemType" minOccurs="0"/>
        <xsd:element name="variationMap"
type="cha:anyURIType">
            <xsd:annotation>
                <xsd:documentation>encodes the variability of the response along the
                axis
            </xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:element name="documentation"
type="cha:anyURIType">
        <xsd:annotation>
            <xsd:documentation>
                documents the purpose, type and encoding of sensitivity information
        </xsd:documentation>
    </xsd:annotation>

```

```

Type: URL
    </xsd:documentation>
    </xsd:annotation>
    </xsd:element>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="CharCoordAreaType">
    <xsd:complexContent>
        <xsd:extension base="stc:stcBaseType">
            <xsd:choice>
                <xsd:element
name="CharBox" type="cha:CharCoordinateType"/>
                <xsd:element
ref="cha:Interval"/>
            </xsd:choice>
            <xsd:attribute name="coord_system_id"
type="xsd:IDREF" use="required"/>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="CharCoordinateType" abstract="true">
    <xsd:annotation>
        <xsd:documentation> A coordinate type to define n-
        D bounding boxes.
        To be overridden by any of the following possible
        substitution elements.
    </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
        <xsd:element name="Name" type="xsd:string"/>
        <xsd:sequence>
            <xsd:element ref="stc:CoordValue"/>
            <xsd:element ref="stc:CSize"/>
        </xsd:sequence>
    </xsd:sequence>
</xsd:complexType>
<xsd:element name="value" type="xsd:anyType" abstract="true"
substitutionGroup="stc:CoordValue"/>
<xsd:element name="size" type="xsd:anyType" abstract="true"
substitutionGroup="stc:CSize"/>
<!-- <xsd:element name="Value" type="stc:double1Type"
substitutionGroup="cha:value" nillable="true"/>
    <xsd:element name="Size" type="stc:double1Type"
substitutionGroup="cha:size" nillable="true"/>
    <xsd:element name="value2" type="xsd:anyType" abstract="true"
substitutionGroup="cha:value"/>
    <xsd:element name="Value2" type="stc:double2Type"
substitutionGroup="cha:value2" nillable="true"/>
    <xsd:element name="Curve2" type="stc:curve2Type"
substitutionGroup="cha:value2" nillable="true"/>
    <xsd:element name="size2" type="xsd:anyType" abstract="true"
substitutionGroup="cha:size"/>
    <xsd:element name="Size2" type="stc:size2Type"
substitutionGroup="cha:size2" nillable="true"/>
    <xsd:element name="Size2Matrix" type="stc:double4Type"
substitutionGroup="cha:size2" nillable="true"/>
    <xsd:element name="Size2Radius" type="stc:double1Type"
substitutionGroup="cha:size2" nillable="true"/>
    <xsd:element name="value3" type="xsd:anyType" abstract="true"
substitutionGroup="cha:value"/>
    <xsd:element name="Value3" type="stc:double3Type"
substitutionGroup="cha:value3" nillable="true"/>
    <xsd:element name="Curve3" type="stc:curve3Type"
substitutionGroup="cha:value3" nillable="true"/>
    <xsd:element name="size3" type="xsd:anyType" abstract="true"
substitutionGroup="cha:size"/>
    <xsd:element name="Size3" type="stc:size3Type" substitutionGroup="cha:size3"
nillable="true"/>
    <xsd:element name="Size3Matrix" type="stc:double9Type"
substitutionGroup="cha:size3" nillable="true"/>
    <xsd:element name="Size3Radius" type="stc:double1Type"
substitutionGroup="cha:size3" nillable="true"/> -->
    <xsd:element name="Interval" type="stc:coordIntervalType"

```



<pre> &lt;xsd:documentation&gt;   Typical sampling Period value of the axis   Type : depends on the dimensionality   1D samplingPeriod based on stc:double1Type   2D samplingPeriod based on stc:size2Type, double4Type or double1Type ( sampling period radius)   3D samplingPeriod based on stc:size3Type or double9Type or double1Type (sampling period 3 radius)   &lt;/xsd:documentation&gt;   &lt;/xsd:annotation&gt; &lt;/xsd:element&gt; &lt;xsd:element name="sampleExtent" type="xsd:anyType" substitutionGroup="stc:CPixSize"&gt;   &lt;xsd:annotation&gt;     &lt;xsd:documentation&gt;       Typical sample Extent value of the axis       "Types : depend on the dimensionality"       1D sampleExtent based on stc:double1Type       2D sampleExtent based on stc:size2Type, double4Type or double1Type (sampling extent radius)       3D sampleExtent based on stc:size3Type, double9Type or double1Type (sampling extent 3 radius)     &lt;/xsd:documentation&gt;     &lt;/xsd:annotation&gt;   &lt;/xsd:element&gt;   &lt;!--&lt;xsd:element name="PixSize" type="stc:double1Type" substitutionGroup="cha:samplingPeriod" nillable="true"/&gt;   &lt;xsd:element name="CPixSize2" type="xsd:anyType" abstract="true" substitutionGroup="cha:samplingPeriod" nillable="true"/&gt;   &lt;xsd:element name="PixSize2" type="stc:size2Type" substitutionGroup="cha:CPixSize2" nillable="true"/&gt;   &lt;xsd:element name="PixSize2Matrix" type="stc:double4Type" substitutionGroup="cha:CPixSize2" nillable="true"/&gt;   &lt;xsd:element name="CPixSize3" type="xsd:anyType" abstract="true" substitutionGroup="cha:samplingPeriod" nillable="true"/&gt;   &lt;xsd:element name="PixSize3" type="stc:size3Type" substitutionGroup="cha:CPixSize3" nillable="true"/&gt;   &lt;xsd:element name="PixSize3Matrix" type="stc:double9Type" substitutionGroup="cha:CPixSize3" nillable="true"/&gt; --&gt;   &lt;xsd:complexType name="ResolutionBoundsType"&gt;     &lt;xsd:sequence&gt; &lt;xsd:element name="unit" type="xsd:anyType" minOccurs="0"/&gt;     &lt;xsd:element name="coordsystem" type="stc:astroCoordSystemType" minOccurs="0"/&gt;     &lt;xsd:element ref="cha:resolutionLimits"/&gt;     &lt;xsd:element name="documentation" type="cha:anyURIType" minOccurs="0"&gt;       &lt;xsd:annotation&gt;         &lt;xsd:documentation&gt;           Defines and explains how this resolution has been estimated         &lt;/xsd:documentation&gt;       &lt;/xsd:annotation&gt;     &lt;/xsd:sequence&gt;   &lt;/xsd:complexType&gt;   &lt;xsd:element name="resolutionLimits" type="stc:coordIntervalType" abstract="true"&gt;     &lt;xsd:annotation&gt;       &lt;xsd:documentation&gt; Hi and Low values of resolution on this axis.       Type: stc:coordScalarInterval, or 2Vec or 3Vec     &lt;/xsd:documentation&gt;     &lt;/xsd:annotation&gt;   &lt;/xsd:element&gt;   &lt;xsd:element name="resolutionLimits1" type="stc:coordScalarIntervalType" substitutionGroup="cha:resolutionLimits"/&gt;   &lt;xsd:element name="resolutionLimits2" type="stc:coord2VecIntervalType" substitutionGroup="cha:resolutionLimits"/&gt;   &lt;xsd:element name="resolutionLimits3" type="stc:coord3VecIntervalType" substitutionGroup="cha:resolutionLimits"/&gt;   &lt;xsd:complexType name="SamplingPrecisionSupportType"&gt;     &lt;xsd:sequence&gt;       &lt;xsd:element name="unit" type="xsd:anyType" minOccurs="0"/&gt;       &lt;xsd:element name="coordsystem" type="stc:astroCoordSystemType" minOccurs="0"/&gt;       &lt;xsd:element ref="cha:samplingPeriodLimits" maxOccurs="unbounded"/&gt;       &lt;xsd:element ref="cha:samplingExtentLimits" minOccurs="0" maxOccurs="unbounded"/&gt;       &lt;xsd:element name="documentation" type="cha:anyURIType" minOccurs="0"&gt;         &lt;xsd:annotation&gt;           &lt;xsd:documentation&gt; A place to hook some explanations about "how" the Sampling was done and assessed.           Type: URI         &lt;/xsd:documentation&gt;         &lt;/xsd:annotation&gt;       &lt;/xsd:sequence&gt;     &lt;/xsd:complexType&gt;     &lt;xsd:complexType name="ResolutionVariabilityType"&gt; </pre>	<pre> minOccurs="0"/&gt;   &lt;xsd:element name="coordsystem" type="stc:astroCoordSystemType" minOccurs="0"/&gt;   &lt;xsd:element ref="cha:samplingPeriodLimits"/&gt;   &lt;xsd:element ref="cha:samplingExtentLimits" minOccurs="0"/&gt;   &lt;xsd:element name="documentation" type="cha:anyURIType" minOccurs="0"/&gt;   &lt;/xsd:sequence&gt;   &lt;/xsd:complexType&gt;   &lt;xsd:element name="samplingPeriodLimits" type="stc:coordIntervalType"&gt;     &lt;xsd:annotation&gt;       &lt;xsd:documentation&gt; Hi and Low values of samplingPeriod on this axis.       Type: stc:coordScalarInterval or 2Vec or 3Vec     &lt;/xsd:documentation&gt;     &lt;/xsd:annotation&gt;   &lt;/xsd:element&gt;   &lt;xsd:element name="samplingPeriodLimits1" type="stc:coordScalarIntervalType" substitutionGroup="cha:samplingPeriodLimits"/&gt;   &lt;xsd:element name="samplingPeriodLimits2" type="stc:coord2VecIntervalType" substitutionGroup="cha:samplingPeriodLimits"/&gt;   &lt;xsd:element name="samplingPeriodLimits3" type="stc:coord3VecIntervalType" substitutionGroup="cha:samplingPeriodLimits"/&gt;   &lt;xsd:element name="samplingExtentLimits" type="stc:coordIntervalType" abstract="true"/&gt;   &lt;xsd:element name="samplingExtentLimits1" type="stc:coordIntervalType" substitutionGroup="cha:samplingExtentLimits"/&gt;   &lt;xsd:element name="samplingExtentLimits2" type="stc:coord2VecIntervalType" substitutionGroup="cha:samplingExtentLimits"/&gt;   &lt;xsd:element name="samplingExtentLimits3" type="stc:coord3VecIntervalType" substitutionGroup="cha:samplingExtentLimits"/&gt;   &lt;xsd:complexType name="ResolutionSupportType"&gt;     &lt;xsd:sequence&gt;       &lt;xsd:element name="unit" type="xsd:anyType" minOccurs="0"/&gt;       &lt;xsd:element name="coordsystem" type="stc:astroCoordSystemType" minOccurs="0"/&gt;       &lt;xsd:element ref="cha:resolutionLimits" maxOccurs="unbounded"/&gt;       &lt;xsd:element name="documentation" type="cha:anyURIType" minOccurs="0"&gt;         &lt;xsd:annotation&gt;           &lt;xsd:documentation&gt; Defines and explains how this resolution has been estimated           Type: URL         &lt;/xsd:documentation&gt;         &lt;/xsd:annotation&gt;       &lt;/xsd:sequence&gt;     &lt;/xsd:complexType&gt;     &lt;xsd:complexType name="SamplingPrecisionSupportType"&gt;       &lt;xsd:sequence&gt;         &lt;xsd:element name="unit" type="xsd:anyType" minOccurs="0"/&gt;         &lt;xsd:element name="coordsystem" type="stc:astroCoordSystemType" minOccurs="0"/&gt;         &lt;xsd:element ref="cha:samplingPeriodLimits" maxOccurs="unbounded"/&gt;         &lt;xsd:element ref="cha:samplingExtentLimits" minOccurs="0" maxOccurs="unbounded"/&gt;         &lt;xsd:element name="documentation" type="cha:anyURIType" minOccurs="0"&gt;           &lt;xsd:annotation&gt;             &lt;xsd:documentation&gt; A place to hook some explanations about "how" the Sampling was done and assessed.             Type: URI           &lt;/xsd:documentation&gt;           &lt;/xsd:annotation&gt;         &lt;/xsd:sequence&gt;       &lt;/xsd:complexType&gt;       &lt;xsd:complexType name="ResolutionVariabilityType"&gt; </pre>
--	--

```

<xsd:annotation>
    <xsd:documentation> This map describes the
variability of the resolution along the axis, the varying shape of the point spread
function, or both. Can be attached to the data
    Type: URI
  </xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="unit" type="xsd:anyType"
minOccurs="0"/>
    <xsd:element name="coordsystem"
type="stc:astroCoordSystemType" minOccurs="0"/>
    <xsd:element name="resolutionMap"
type="cha:anyURIType"/>
    <xsd:element name="documentation"
type="cha:anyURIType" minOccurs="0">
      <xsd:annotation>
        <xsd:documentation>
defines and explains how this resolution has been estimated
          Type: URL
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="SamplingPrecisionVariabilityType">
  <xsd:sequence>
    <xsd:element name="unit" type="xsd:anyType"
minOccurs="0"/>
    <xsd:element name="coordsystem"
type="stc:astroCoordSystemType" minOccurs="0"/>
    <xsd:element name="samplingPrecisionMap"
type="cha:anyURIType">
      <xsd:annotation>
        <xsd:documentation>This
map describes the variability of the sampling along the axis, or the varying shape
of the sampling
          fonction, or both. Can be attached
to the data implemented as anyURI type
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  <xsd:element name="documentation"
type="cha:anyURIType" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
</xsd:schema>

```

Ce schéma est disponible sur l'adresse :

<http://www.ivoa.net/xml/Characterisation/Characterisation-v1.11.xsd>

**Annexe n° 07 : Liste des utypes pour le modèle de la Characterisation Version 1.11**