Interopérabilité entre Aladin et d'autres applications 'desktop'

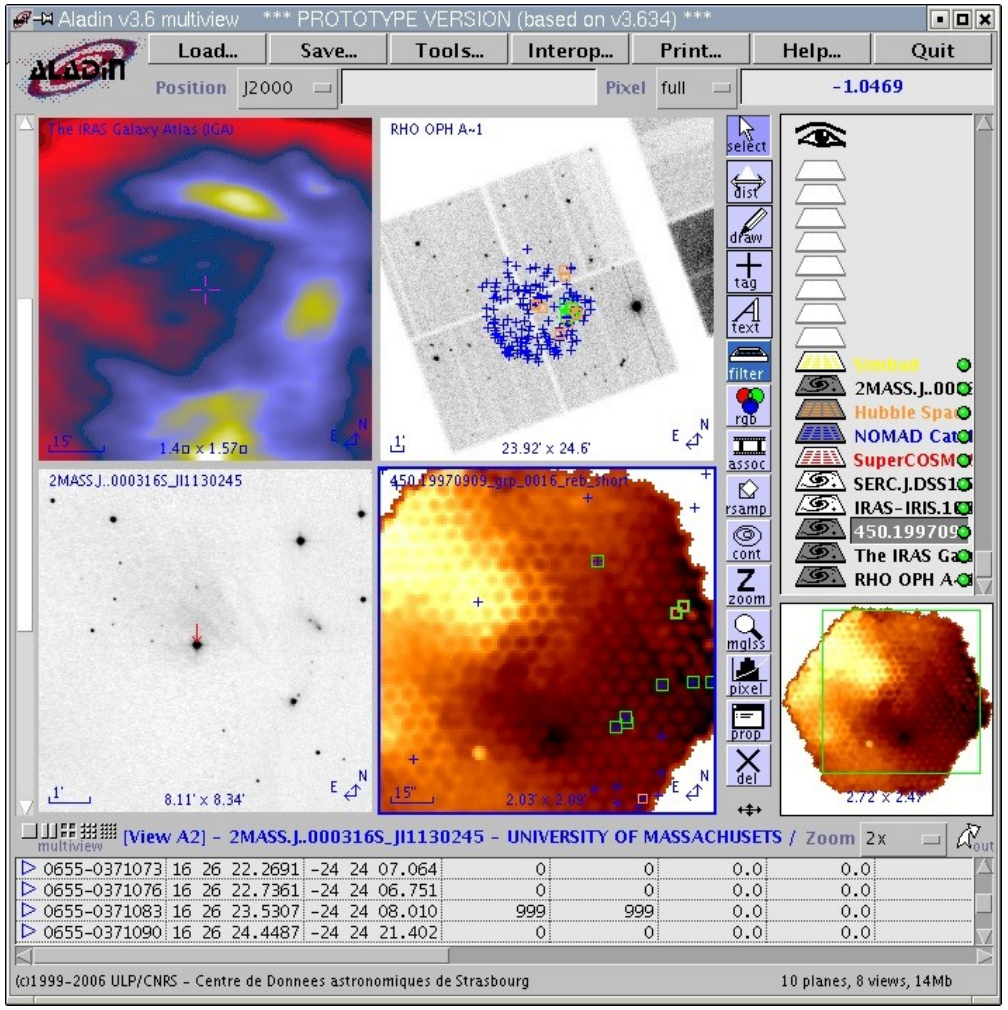
Thomas Boch [CDS]

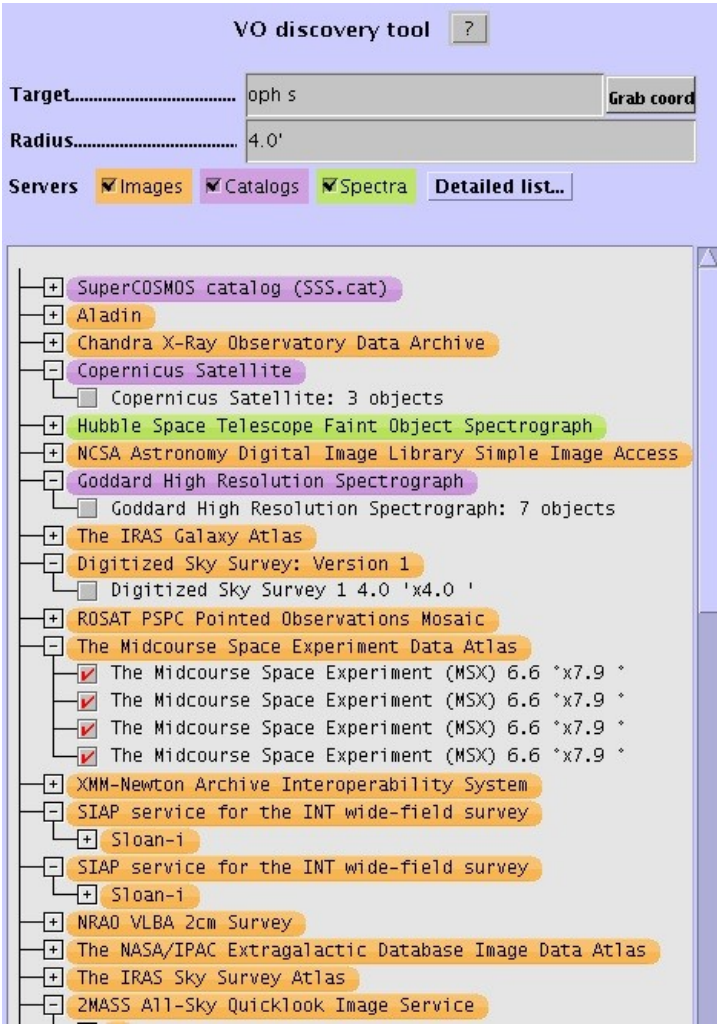
Journée Briques Logicielles –
13 décembre 2006

Plan

- PLASTIC, un protocole d'interopérabilité entre applications
- Mécanisme de plugins dans Aladin
- Interopérabilité entre Aladin et IDL

Aladin ?


Aladin v3.6 multiview *** PROTOTYPE VERSION (based on v3.634) ***
Load... Save... Tools... Interop... Print... Help... Quit
Position J2000
Pixel full
-1.0469
The IRAS Galaxy Atlas (IGA)
RHO OPH A~1
2MASS J..000316S_JI1130245
[View A2] - 2MASS J..000316S_JI1130245 - UNIVERSITY OF MASSACHUSETS / Zoom 2x
0655-0371073 16 26 22.2691 -24 24 07.064
0655-0371076 16 26 22.7361 -24 24 06.751
0655-0371083 16 26 23.5307 -24 24 08.010
0655-0371090 16 26 24.4487 -24 24 21.402
(c)1999-2006 ULP/CNRS - Centre de Donnees astronomiques de Strasbourg
10 planes, 8 views, 14Mb


VO discovery tool
Target
oph s
Grab coord
Radius
4.0'
Servers
Images
Catalogs
Spectra
Detailed list...
SuperCOSMOS catalog (SSS.cat)
Aladin
Chandra X-Ray Observatory Data Archive
Copernicus Satellite
Copernicus Satellite: 3 objects
Hubble Space Telescope Faint Object Spectrograph
NCSA Astronomy Digital Image Library Simple Image Access
Goddard High Resolution Spectrograph
Goddard High Resolution Spectrograph: 7 objects
The IRAS Galaxy Atlas
Digitized Sky Survey: Version 1
Digitized Sky Survey 1 4.0 'x4.0 '
ROSAT PSPC Pointed Observations Mosaic
The Midcourse Space Experiment Data Atlas
The Midcourse Space Experiment (MSX) 6.6 'x7.9 '
The Midcourse Space Experiment (MSX) 6.6 'x7.9 '
The Midcourse Space Experiment (MSX) 6.6 'x7.9 '
The Midcourse Space Experiment (MSX) 6.6 'x7.9 '
XMM-Newton Archive Interoperability System
SIAP service for the INT wide-field survey
Sloan-i
SIAP service for the INT wide-field survey
Sloan-i
NRAO VLBA 2cm Survey
The NASA/IPAC Extragalactic Database Image Data Atlas
The IRAS Sky Survey Atlas
2MASS All-Sky Quicklook Image Service

PLASTIC

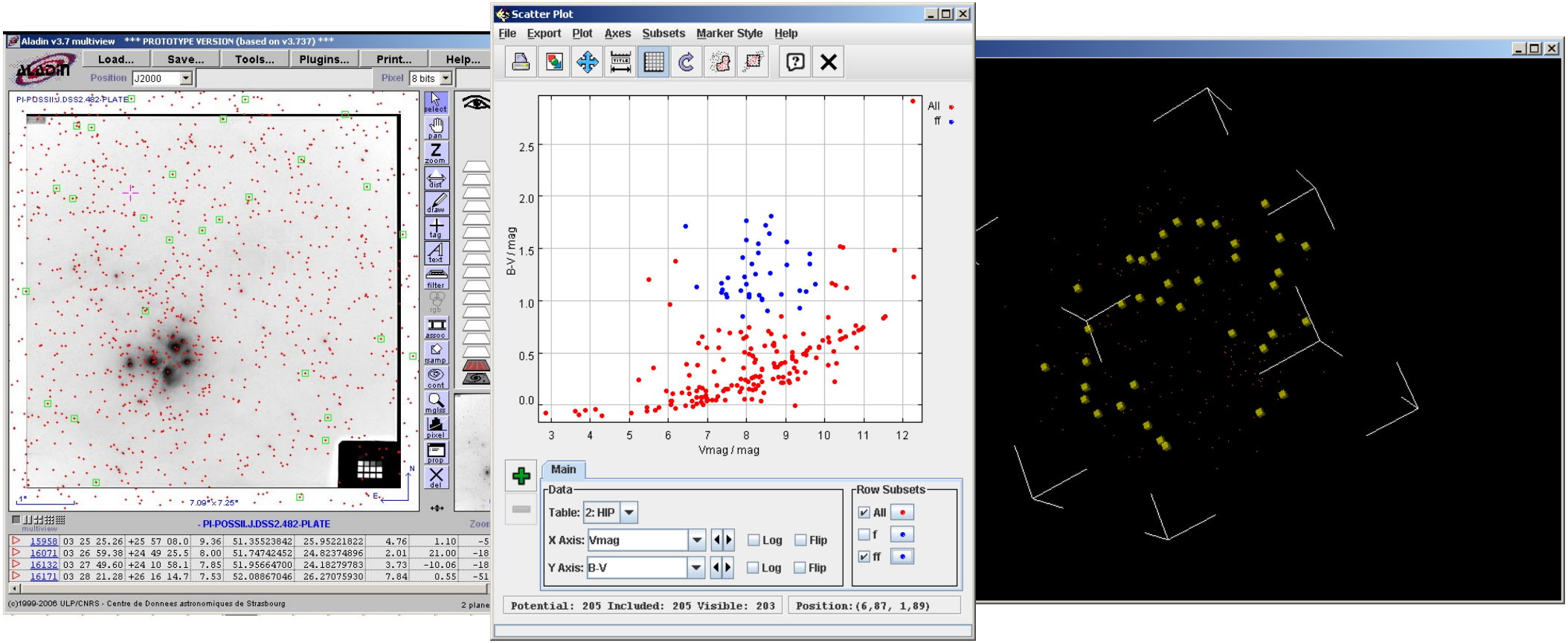
- PLASTIC = PLateform for AStronomical Tool InterConnection
- Un protocole permettant la communication entre applications hétérogènes (sans que celles-ci ne se connaissent a priori)
- Développé au sein du projet européen VOTech
  - Initialement : collaboration entre AstroGrid, CDS (Aladin), Mark Taylor (TOPCAT), INAF (VisIVO)

Motivations

- Etendre facilement les fonctionnalités d'une application ...
- .. sans avoir à réinventer la roue
  - --> combiner les fonctionnalités de différentes applications (“briques”)
- Explorer les données dans des vues liées (“linked views”)
- Cas d'utilisation
  - *Visualiser dans TOPCAT des données sélectionnées depuis AstroScope*
  - *En partant d'un catalogue chargé dans Aladin, tracé d'un diagramme couleur-couleur (dans TOPCAT), sélection des objets les plus rouges, et visualisation dans Aladin de leur position sur le ciel*

Vues liées

- Visualisation d'un même ensemble de données depuis différentes perspectives



Architecture

- Protocole basé sur l'échange de messages
- Architecture “Publish-Subscribe”
- Un “Hub” tourne localement :
  - Gère les inscriptions/désinscriptions des différentes applications
  - Dispatche les messages qu'il reçoit
- Connexion au hub et communication
  - Par Java RMI
  - Par XML-RPC
- Plusieurs implémentations de hub existent
  - Implémentation AstroGrid, également disponible dans le Workbench (impl. “de référence”)
  - PlasKit, développé par Mark Taylor (plus léger)

Architecture (2) – enregistrement d'une nouvelle application

XML-RPC

②

registerXMLRPC(
msg[])

$HOME/.plastic ①

```
plastic.version = 0.4
plastic.xmlrpc.url=http://...
plastic.rmi.port=1099
```

identifiant ③

④ New App registered(VisIVO_ID)

④ New App registered(VisIVO_ID)

Java RMI

Java RMI

Hub PLASTIC

Architecture (3) – envoi d'un message

XML-RPC

loadVOTable(URL)

①

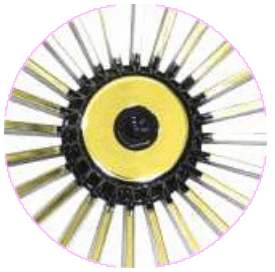
②

loadVOTable(URL)

②

loadVOTable(URL)

Java RMI



Hub PLASTIC

Java RMI

Messages

- Interopérabilité assurée par la définition d'un ensemble de messages “de base”
  - *Get application name*
  - *Load FITS image from URL(url: String, id: String)*
  - *Load VOTable from URL(url: String, id: String)*
  - *Show objects(id: String, rows: int[])*
  - *Point at coordinates(ra: double, dec: double)*
  - *...*
- Cette liste peut être étendue selon les besoins
  - Le hub ne comprends pas la sémantique des messages
  - Il se contente de dispatcher les messages aux applications concernées

PLASTIC, aller plus loin

- Note IVOA publiée début décembre
- Actuellement, 14 applications PLASTICisées dans divers langages (Java, C++, Tcl, Perl, Ruby)
- Exemples d'implémentation en C++, Perl, Java, Python, Ruby, Tcl
- Démo

Plugins Aladin

- Nombreuses demandes pour obtenir le code source d'Aladin pour ajouter des fonctionnalités manquantes
- Mais développement peu adapté à une distribution de code
- --> Mécanisme de plugins
  - Fournir une “vue” sur certains objets Aladin
  - Permet d'étendre Aladin par l'ajout de nouvelles fonctions non développées en interne

Développer un plugin

- Créer une classe héritant de *AladinPlugin*
  - Définir les 2 méthodes abstraites :
    - menu() --> nom du plugin dans la liste des plugins disponibles
    - exec() --> appelée à l'exécution du plugin
  - Redéfinir éventuellement :
    - description()
    - scriptCommand()
    - scriptHelp()
    - inSeparateThread()
    - execScriptCommand(String[] params)
- La classe AladinData permet l'accès et la manipulation des données présentes dans Aladin
  - Accès aux plans images et catalogues
  - Accès aux valeurs des pixels, fonctions astrométriques
  - Création de nouveaux plans images et catalogues

Installer et utiliser un plugin

- Dans le répertoire “home” (**$HOME** sous *ix, **C:\Document and Settings\<user>** sous Windows), créer un répertoire *.aladin/Plugins*
- Copier les fichiers .class du plugin dans ce répertoire
- (Re)démarrer Aladin
- Le plugin est disponible depuis le menu “Plugins”
- Quelques exemples
  - Photométrie d'ouverture
  - Rotation d'une image
  - Listener de souris
  - Création d'un plan catalogue

Aladin/IDL

- Besoin : *pouvoir utiliser Aladin et ses fonctionnalités depuis des applications IDL*

- IDL Java Bridge (disponible dans IDL v.>=6.0) permet la création d'objets et l'appel de méthodes Java depuis IDL
  - Repose sur JNI (Java Native Interface)

- Wrapping des appels à Aladin dans des fonctions IDL

Fonctions disponibles

- *launch_aladin* – lance Aladin
- *load_table* – chargement dans Aladin d'une table (sous la forme de vecteurs)
- *get_table* – récupère dans des vecteurs IDL les données d'un catalogue chargé dans Aladin
- *load_image* – chargement d'une image
- *get_image* – récupère dans des variables IDL le header et les données d'une image FITS
- *send_script_command* – envoi d'une commande script
- *add_colortable_in_aladin* – ajout d'une nouvelle table de couleurs

Fonctions disponibles (2)

- *set_colortable* – changement de la table de couleurs pour l'image courante
- *set_reticle_pos* – modification de la position du réticule
- *get_reticle_pos* – récupération de la position du réticule
- *get_pixelval_at_reticle_pos* – récupération de la valeur du pixel à la position courante du réticule

Démo

- Exemple d'utilisation d'Aladin dans IDL

Appel de fonctions IDL depuis Aladin

- Java Bridge ne permet pas l'appel de fonctions IDL depuis Java
  - Serait intéressant dans le cadre des plugins Aladin par exemple
- Depuis IDL v6.3 : IDL Export Bridge
  - Librairie Java permettant d'appeler des scripts IDL (dans du code Java)
- A tester ...

Liens

- PLASTIC
  - Home page : http://plastic.sourceforge.net/
  - Note IVOA : **http://ivoa.net/Documents/latest/PlasticDesktopInterop.html**
  - Liste des messages “de base” : http://plastic.sourceforge.net/coremessages.html
- Plugins Aladin
  - **http://aladin.u-strasbg.fr/nph-aladin.pl?frame=plugins**
- Aladin/IDL
  - **http://wiki.eurovotech.org/twiki/bin/view/VOTech/AladinIDL**