

AstroGrid

Thomas Boch [CDS]



OV-France – Tutoriel Web Services –
1er et 2 février 2006



AstroGrid ??

- Projet britannique
 - But : développement d'une infrastructure *Observatoire Virtuel* pour la communauté astronomique britannique et mondiale
 - Orienté services
 - Participation UK à l'IVOA
- Financé par PPARC et l'UE
 - AstroGrid 1 : 2001-2004
 - AstroGrid 2 : 2004-2007
 - ◆ Project lead : Andy Lawrence
 - ◆ Project scientist : Nic Walton
 - ◆ Project manager : Tony Linde
 - ◆ Technical lead : Keith Noddle



AstroGrid ?? (2)

- Membre du projet Euro-VO
 - Un des 4 instituts participant à VOTech
 - ◆ Impliqué surtout dans le Design Study 3 (New infrastructure)



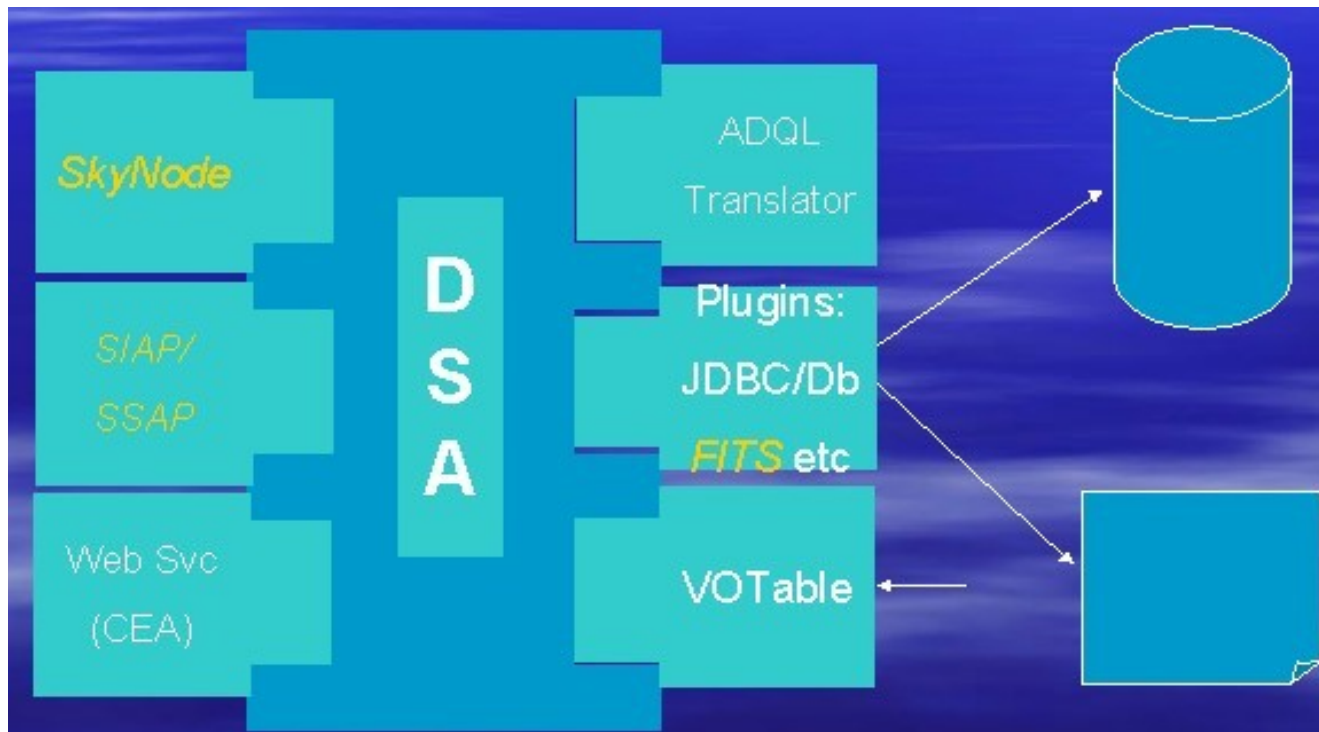
Les composants

- CEA
- Community
- MySpace (FileManager+FileStore)
- JES
- DSA
- Portal
- Registry

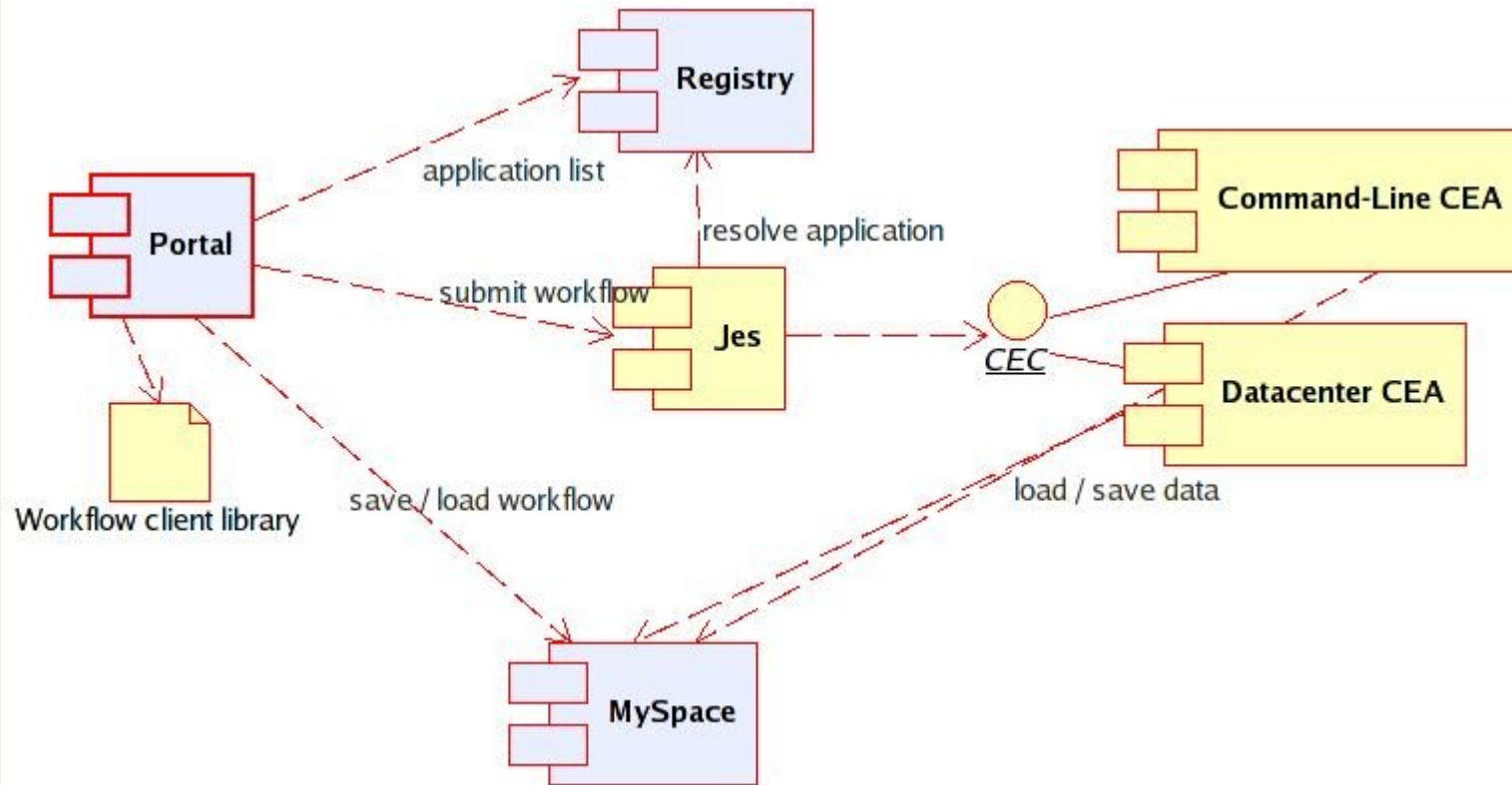


DSA (Data Set Access)

- Ensemble de servlets Java permettant la publication de données sur le VO
 - Nécessite Tomcat
 - Déployable sous forme de fichiers WAR



Architecture



Glossaire

- Portal : Interface Web permettant l'accès aux services Astrogrid
- Workbench : application cliente fournissant
 - les mêmes fonctionnalités que le portail Web
 - + accès programmatique aux services
- CEA : **C**ommon **E**xecution **A**rchitecture
 - Standard de description des applications (paramètres d'entrée, paramètres de sortie, méthodes d'accès, ...)
 - Permet une description et un accès uniforme aux différentes applications composant un workflow



Glossaire (2)

- JES : **J**ob **E**xecution **S**ystem
 - Moteur d'exécution du workflow
 - contrôle l'exécution du workflow dans sa globalité
- MySpace : espace de stockage distribué
 - Stockage des résultats intermédiaires et finaux
 - Fichiers référencés par leur IVORN (IVO Resource Name),
eg <ivo://uk.ac.le.star/ThomasBoch#OV-France/2mass.xml>
- Registry : annuaire listant (entre autres) les applications CEA disponibles
- CEC : **C**ommon **E**xecution **C**ontroller
 - Contrôle l'exécution d'une tâche précise
 - Avertit JES de la fin d'une tâche



Workflow : mécanisme de persistance

- Structure d'un workflow décrite par un XML Schema
- Chaque description de workflow est sauvegardée dans un fichier XML (dans MySpace)
- Exemple :

```
- <Activity xsi:type="sequence">
  <Activity var="twomassadql" value="" xsi:type="set"/>
  - <Activity xsi:type="script">
    <description>open twomass query template</description>
    - <body>
      ev = astrogrid.ioHelper.getExternalValue("http://www.merlin.ac.uk/astrogrid/cctwomass.q") twomassadql = astrogrid.ioHelper.getContents(ev)
    </body>
  </Activity>
  - <Activity xsi:type="script">
    - <description>
      Insert sky area, 2MASS bands to select on, and colour limit, into query
    </description>
    - <body>
      import groovy.text.Template import groovy.text.SimpleTemplateEngine binding = ["ramin":nra[0], "ramax":nra[1], "raaltmin":nraalt[0],
      "raaltmax":nraalt[1], "decmin":ndec[0], "decmax":ndec[1], "twomassb1":twomass[0], "twomassb2":twomass[1], "twomasslim":twomass[2]] engine =
      new SimpleTemplateEngine() template = engine.createTemplate(twomassadql).make(binding) twomassadql = template.toString()
    </body>
  </Activity>
  - <Activity name="IR query to 2MASS" joinCondition="any" xsi:type="step">
    <description>IR query with colour cut</description>
    - <tool name="roe.ac.uk/DSA_2MASS/ceaApplication" interface="adql">
      - <input>
        - <parameter name="Query" indirect="false">
          <ns1:value>$(twomassadql)</ns1:value>
        </parameter>
        - <parameter name="Format" indirect="false">
          <ns2:value>VOTABLE</ns2:value>
      </input>
    </tool>
  </Activity>
```



Eléments d'un workflow

- Comparable à un langage de programmation
- **Step** : Élément appelant une application CEA
 - Requêtes à des centres de données (ConeSearch, SIA, ADQL, ...)
 - Tâches de traitement (SExtractor, HyperZ, CrossMatcher, ...)
- **Set** : Définit une nouvelle variable ou met à jour la valeur d'une variable existante
- Structures de contrôle
- Scripts Groovy



Workflow : structures de contrôle

- **Flow** : ensemble d'opérations exécutées en parallèle
- **Sequence** : ensemble d'opérations exécutées séquentiellement
- **If** : exécution conditionnelle
- **ForLoop** : boucle itérant sur un ensemble
- **ParallelLoop** : boucle dont chaque itération est indépendante des autres --> parallélisme
- **WhileLoop** : boucle avec test à chaque itération



Groovy dans les workflows

AG

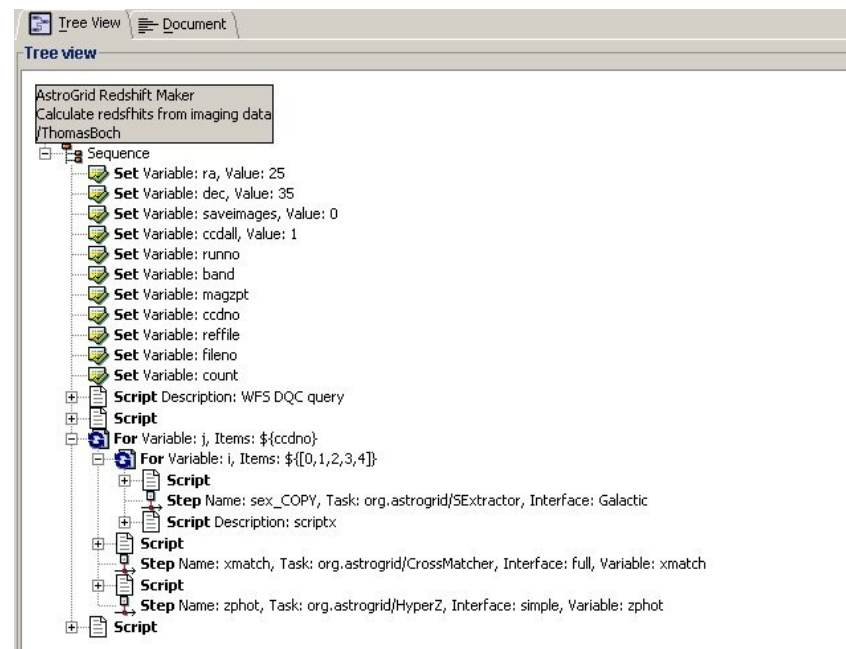
- Langage de script pour Java
 - Super-ensemble de Java
 - Sucre syntaxique pour faciliter certaines opérations
- Utilisé comme “glue” dans les workflows :
 - Post-traitement sur le résultat d'une appli. CEA (eg: extraction de la valeur d'un champ)
 - Adaptation des paramètres d'entrée aux paramètres attendues par l'application
 - Accéder aux données MySpace

```
col = ( 0 ... table.columnCount ).find{ table.getColumnInfo(it).getUCD() == 'VOX:Image_AccessReference' }  
urls = table.columnIterator(col).collect{it}
```



Création/visualisation d'un workflow

- Via le portail Web
 - Visualisation & création
- Via le Workbench
 - Visualisation uniquement pour le moment
 - Création de workflows : **coming soon**

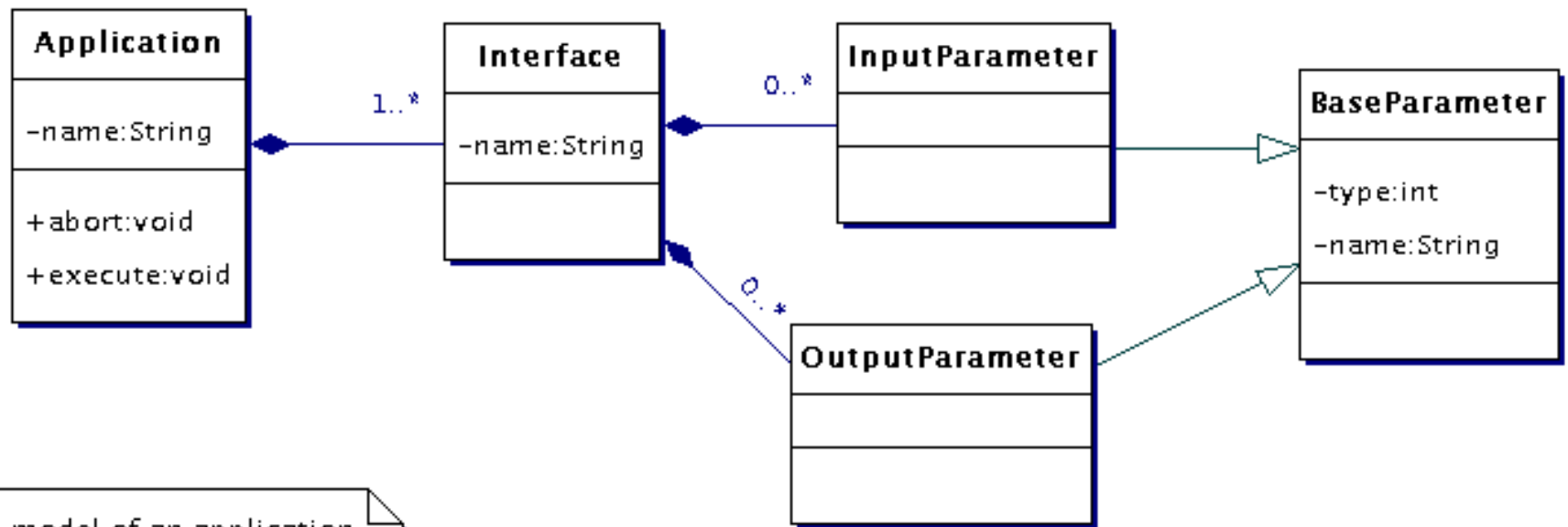


CEA (Common Execution Architecture)

- CEA définit un ensemble d'interfaces et un schéma modélisant la façon dont une application s'exécute dans le monde VO
 - Définit un modèle et une interface uniforme pour une application et ses paramètres
 - Fournit une description de plus haut niveau que WSDL 1.1
 - Fournit des extensions au schéma VO Resource (pour l'enregistrement des applications dans les registries)
 - Supporte les appels asynchrones d'une application
 - Les applications CEA partagent le même contrat WSDL



CEA (2)



Domain model of an application



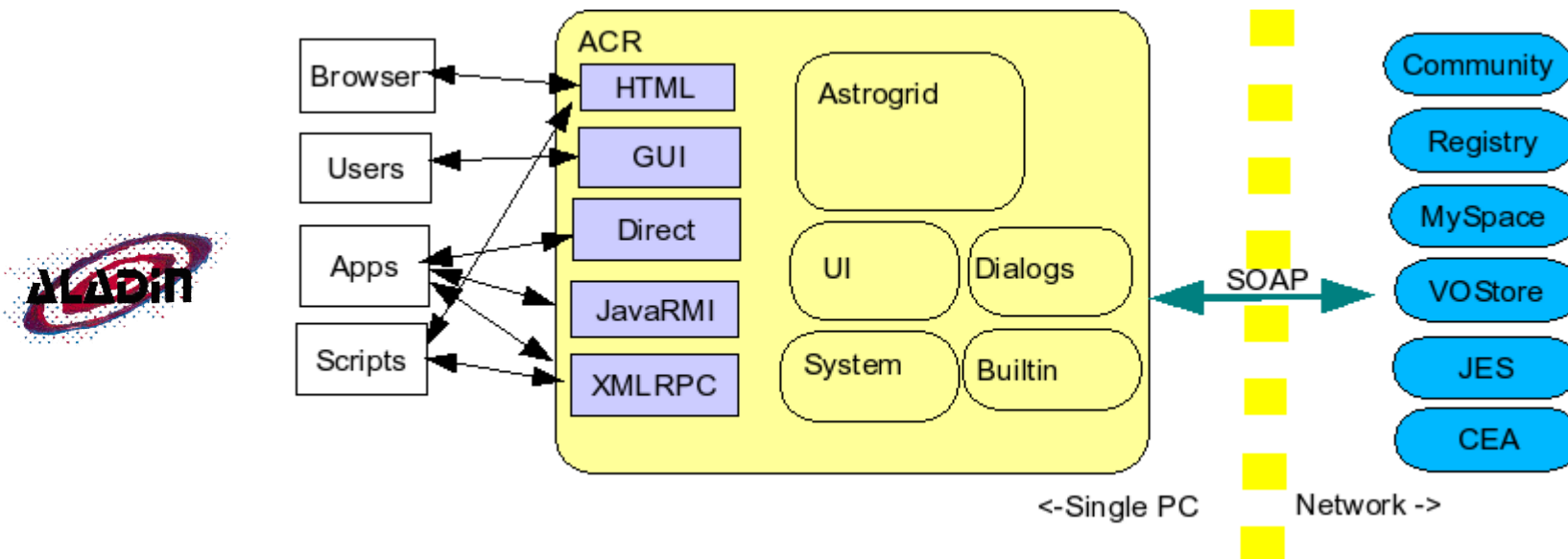
CEA : wrapping d'applications existantes

- Command line Execution Controller
 - Permet de wrapper des applications ***ligne de commande*** comme applications CEA
 - Pré-requis
 - ◆ Pas d'utilisation de GUI pour entrer des paramètres
 - ◆ Pas d'interactivité
 - Paramètres d'entrée, paramètre de sortie et options de l'application décrits via un fichier de configuration (en XML)
- HTTP Common Execution Connector
 - Permet le wrapping de services HTTP
 - Uniquement méthode GET, POST non supporté



Accès programmatique aux services AstroGrid

- ACR : Astronomy Client Runtime
 - API offrant une “façade” permettant un accès “programmatique” aux services Astrogrid
 - Multi-plateformes & multi-langages
- Exemple:



ACR

- ACR permet notamment
 - Au niveau “workflow” :
 - ◆ Soumettre un workflow
 - ◆ Avoir des informations sur un job en cours d'exécution
 - ◆ Obtenir des composants graphiques :
 - Workflow viewer
 - Job monitor
 - Au niveau MySpace
 - ◆ Upload/download de fichiers sur son espace MySpace
 - ◆ Composant graphique de type navigateur
 - Interrogation du registry



ACR (2)

- Exemple de code (accès à MySpace) :

```
// recherche d'une instance d'ACR existante
Finder f = new Finder();
acr = f.find();
// ...
resChooser = (ResourceChooser)acr.getService(ResourceChooser.class);
URI uri = resChooser.fullChooseResource("MySpace Resource Chooser",
                                       true, false, false);
URL u = myspace.getReadContentURL(uri);
```



Liens

- AstroGrid Release 1.3 :
<http://software.astrogrid.org/>
- Portails AstroGrid :
<http://wiki.astrogrid.org/bin/view/Astrogrid/AstroGridPortals>
- CEA Design (note IVOA) :
<http://www.ivoa.net/Documents/Notes/CEA/CEADesignIVOANote-20050513.html>
- Command line Execution controller :
<http://astrogrid.jb.man.ac.uk:8080/astrogrid-cea-commandline/provider/ApplicationConfiguration.html>



Liens (2)

- Documentation workflow Astrogrid :
<http://software.astrogrid.org/components/jes/userguide-architecture.html>
- JES scripting FAQ :
<http://wiki.astrogrid.org/bin/view/Astrogrid/JesScriptingFAQ>
- Différence Portal/Workbench :
<http://software.astrogrid.org/userdocs/index.html>
- ACR :
<http://www.astrogrid.org/maven/docs/HEAD/desktop/multiproject/acr-interface/apidocs/index.html>

