UWS comments and proposition from usecases by VOParis & CNES

Jonathan Normand, Pierre Le Sidaner
Observatoire de Paris
Jean-Christophe Malapert
CNES

Context

Virtual Observatory Paris Data Centre

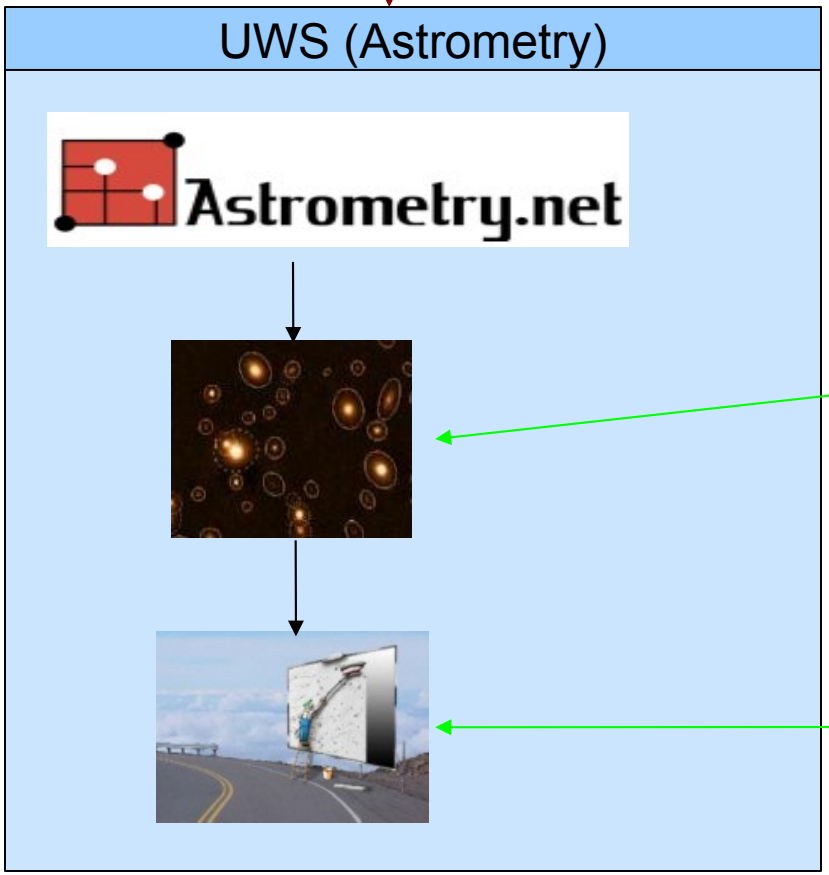
cnes CENTRE NATIONAL D'ÉTUDES SPATIALES

- Services description
- Step by step pb
- Practical example
- Extentions
- Service response

SI TOOLS 2

INFORMATION SYSTEM TOOL

Create jobs / Get Jobs status



take the result of astrometry.net to compare source extraction (sextractor) with catalogues and make polynomial approximation of correction

projection in tan plane using swarp

Context

- Others UWS services at Observatoire de Paris:

  - Astrocheck service to validate astrometry

  - Skybot process in progress

Step by step protocol pb

- **Section 2.1.3 Job**
  - HELD is useless : a job is stopped and can not be automatically executed = suspended. If job is started it's anyway in the queue
  - Aborted and suspended is enough
- **Section 2.1.11 parameter list**
  - To many choices => to complex in use. 3 possibilities to add parameters in rest service, 3 solutions for no reason
  - No way to remove a parameter

=> How clients know behavior of a server

why don't use something conforms to REST philosophy (one url + one verb = one action).

Step by step protocol pb

- **Section 2.2.3.1 Creating a Job**

The paragraph is clear until "One use of this facility might be to have the job placed into a potentially running state by adding PHASE=RUN to the job creation step." => two ways to run a job

Why don't you want to use REST as it Is made for : One URL + one verb = One action.

Running a job is done further on

- **Section 2.2.3.2 Deleting a Job.**
  - Why sending job list after delete? Could be long.
  - Unique sol : DELETE /{job}/{job-id}

    Tunneling API exists for server application to handle delete.
  - One action …

.

Step by step protocol pb

- **Section 2.2.3.2 & 2.2.3.3 Changing duration & destruction time**

As I don't see the real interest of this feature. What the service should respond if not implemented ?

- **Section 2.2.3.5 Starting a Job**

If you **MAY** start a job using … how you must start a job ?

- **Section 2.2.3.6 Aborting a Job**

In REST to abort a job, we just could use an uri as /{jobs}/(job-id)/abort, why adding phase=abort as parameter ?

EXAMPLE of application

UWS document proposes you two way to run a job Astrometry

1) in 3 steps

- Create the job post of /astrometry
  Action is redirect to /astrometry/job-id
- setting parameter to the job, there is **3 options**
  Post on /astrometry/job-id parametrer-name=value
  Post /astrometry/job-id/parameters parameter-name=value
  Put /astrometry/job-id/parameters/parameter-name
  Run the job with a Post
  /astrometry/job-id/phase with parameter phase=run

2) But you can also do 3 phases on one

- post /astrometry with param and phase=run

What is the interest of implementing 3 methods for one action !

You make 3 more time to understand the document
and 6 more time to implement (client + server)
REST is one URL one action not 3.

Virtual Observatory Paris Data Centre

cnes CENTRE NATIONAL D'ÉTUDES SPATIALES

User/developer needs

- Need ROA (Resource Oriented Architecture) design (too many actions for one resource, status not well defined)
- Need sequence diagrams
- Need to upload files
- Need pagination capability for job list
- Need a simple way to describe the REST web service

The Evolution asked for UWS

- To be compliant with ROA methodology

A REST service is based on ROA (Resource Oriented Architecture) :
- Creating a domain model such as OOA/D
- Deriving a resource model from the first step
- Defining unique identification of the resource
- Defining for each resource which HTTP methods are supported and what service(s) they provide
- Defining method status
- Defining the formats of representation exchanged between your server and client applications

Services description

Step by step pb

Practical example

Extentions

Service response

Example :

Resource name	- Status meaning - Resource description
Resource identifier	
Resource Verb : HTTP status : Representation	

See resources model that we propose Or text format

Proposition of complement for UWS

Services description

Step by step pb

Practical example

Extentions

Service response

- Job cannot be deleted when phase != [COMPLETED|ABORTED|ERROR] => need to be stopped before, status ?

- Job cannot be started because its current status is not pending, status ?

- Job cannot be cancelled because its current status is not [pending|queued|executing], status ?

- Cannot update a parameter after a pending phase, status ?

- Connot create parameters after a pending phase, status ?

- Status when a resource is optional ?

Propositions in :

See resources model that we propose Or text format

The Evolution asked for UWS

- File upload capability (multipart/form-data)
to use UWS from desktop, upload configuration files
(application/x-www-form-urlencoded is not enough)

- JDL description by the use of WADL
(allows a larger user community)

- Sequence diagrams are needed to explain
when resources can be handled during the UWS process

- Pagination at job list level to handle long job list