Coordination Action for the integration of Solar System Infrastructures and Science

Project No.: 261618
Call: FP7-INFRA-2010-2

Report summarizing changes to recommendations by the IVOA and similar bodies

Version 1.0

Title:	**Report summarizing changes to recommendations by the IVOA and similar bodies**
Document No.:	CASSIS *Deliverable:* **D3.4**
Date:	10 January 2014
Editor:	**K. Benson**, UCL-MSSL
Contributors:	R.D. Bentley
Distribution:	Project

CAPACITIES

e-infrastructure

Revision History

Version	Date	Released by	Detail
0.1	03/09/2013	R.D. Bentley	First draft
0.2	14/12/2013	K. Benson	Heavy revisions
1.0	10/01/2014	K. Benson	Released version

1. Introduction 1
1.1. IVOA Definition 1
1.2. IVOA References 1
1.3. IVOA Architecture 3
2. Registries 3
2.1. Introduction 3
2.2. Definition of Registries 3
2.3. Support Interface (VOSI) and CatalogService vs. DataCollection 4
2.4. Mirrors, Primary, and Secondary URLS in the Registry 4
3. UCD and UTypes 6
3.1. Introduction 6
3.2. Definition 6
3.3. Solar input to IVOA 6
4. VOTable 7
4.1. Definition 7
4.2. Sample 7
5. Query Interface 9
6. Asynchronous calls to Applications and Query Interface 9
7. References for IVOA Standards 11
7.1. VOTable 11
Description 11
Usage in HELIO and EPN-TAP 11
7.2. Table Access Protocol 11
Description 11
Usage in EUROPLANET 12
Usage in HELIO 12
Sharing of Resources 12
7.3. Universal Worker Service and Parameter Description Language 12
Description 12
Usage in HELIO 12
7.4. UCD and UTypes 13
Description 13
Usage in EUROPLANET 13
Usage in HELIO 14
Sharing of Resources 14
7.5. Registries 14
Description 14
7.6. Simple Access Messaging Protocol (SAMP) 14
Reference 14
Description 15
Usage in EUROPLANET 15
Usage in HELIO 15
Sharing of Resources 15
8. Taverna – Example of Client crossing domains 16
Reference 16
Description 16
Usage 16

Changes to recommendations by the IVOA
Deliverable D3.4

1. Introduction

The International Virtual Observatory Alliance (IVOA) is an important organization that defines standards that should be used when establishing virtual observatories. The primary focus of this document is to describe the issues and interactions with IVOA in regards to the domains related into the CASSIS coordination activities.

The IVOA has produced specifications, standards and techniques that cross boundaries in the goal of producing a seamless Virtual Observatory. IVOA is useful when defining aspects of the interfaces to the HELIO and EUROPLANET services as well as in other areas. In return, the IVOA benefited from CASSIS input in relation to the coordination of HELIO and EUROPLANET projects. The IVOA has had more focus on Astronomy and Astrophysics with HELIO and EUROPLANET project providing needed benefit to the IVOA community in the area of planetary science and *Heliophysics* – the effect of the Sun on the Solar System.

1.1. IVOA Definition

The International Virtual Observatory Alliance was formed in June 2002 with a mission to *"facilitate the international coordination and collaboration necessary for the development and deployment of the tools, systems and organizational structures necessary to enable the international utilization of astronomical archives as an integrated and interoperating virtual observatory."*

The IVOA focuses on the development of standards and encourages their implementation for the benefit of the worldwide astronomical community. Working Groups are constituted with cross-program membership in those areas where key interoperability standards and technologies have to be defined and agreed upon. The Working Groups develop standards using a process modeled after the World Wide Web Consortium, in which Working Drafts progress to Proposed Recommendations and finally to Recommendations. Ultimately, the recommendations may be endorsed by the Virtual Observatory Working Group of Commission 5 (Astronomical Data) of the International Astronomical Union (IAU).

The IVOA also has Interest Groups that discuss experiences using VO technologies and provide feedback to the Working Groups. Ad-hoc and permanent committees deal with specific scientific and procedural topics. Interaction with other scientific disciplines interested in data inter-operability is also pursued through dedicated Liaison Groups.

1.2. IVOA References

To build the service	
IVOA Architecture	http://www.ivoa.net/documents/Notes/IVOAArchitecture/20101123/IVOAArchitecture-1.0-20101123.pdf
IVOA website	http://www.ivoa.net
IVOA Document	http://www.ivoa.net/documents/index.html
VOTable Specification	http://www.ivoa.net/Documents/VOTable/20091130/
Table Access Protocol	http://www.ivoa.net/Documents/TAP/20100327/
Universal Worker Service	http://www.ivoa.net/documents/UWS/index.html
Universal Column Descriptor	http://www.ivoa.net/Documents/latest/UCD.html
Parameter Description Language	http://www.ivoa.net/documents/PDL/20130718/index.html

Registry Metadata	http://www.ivoa.net/documents/latest/RM.html
Registry Interface	http://www.ivoa.net/documents/RegistryInterface/20091104/
Accomazzi et. al, Describing Astronomical Catalogues and Query Results with XML	http://cds.u-strasbg.fr/doc/astrores.htx
FITS: Flexible Image Transport Specification	http://fits.gsfc.nasa.gov/
EPN-TAP	http://voparis-europlanet.obspm.fr/utilities/EPN_TAP_implementation_0.13.pdf
EPN-DM	http://www.europlanet-idis.fi/documents/public_documents/EPN-DM-v2.0.pdf
Simple Access Messaging Protocol	http://www.ivoa.net/documents/SAMP/20120411/index.html

1.3. IVOA Architecture

The IVOA Architecture Note is a good reference to understand all the services and standards provided by the IVOA.

http://www.ivoa.net/documents/Notes/IVOAArchitecture/20101123/IVOAArchitecture-1.0-20101123.pdf

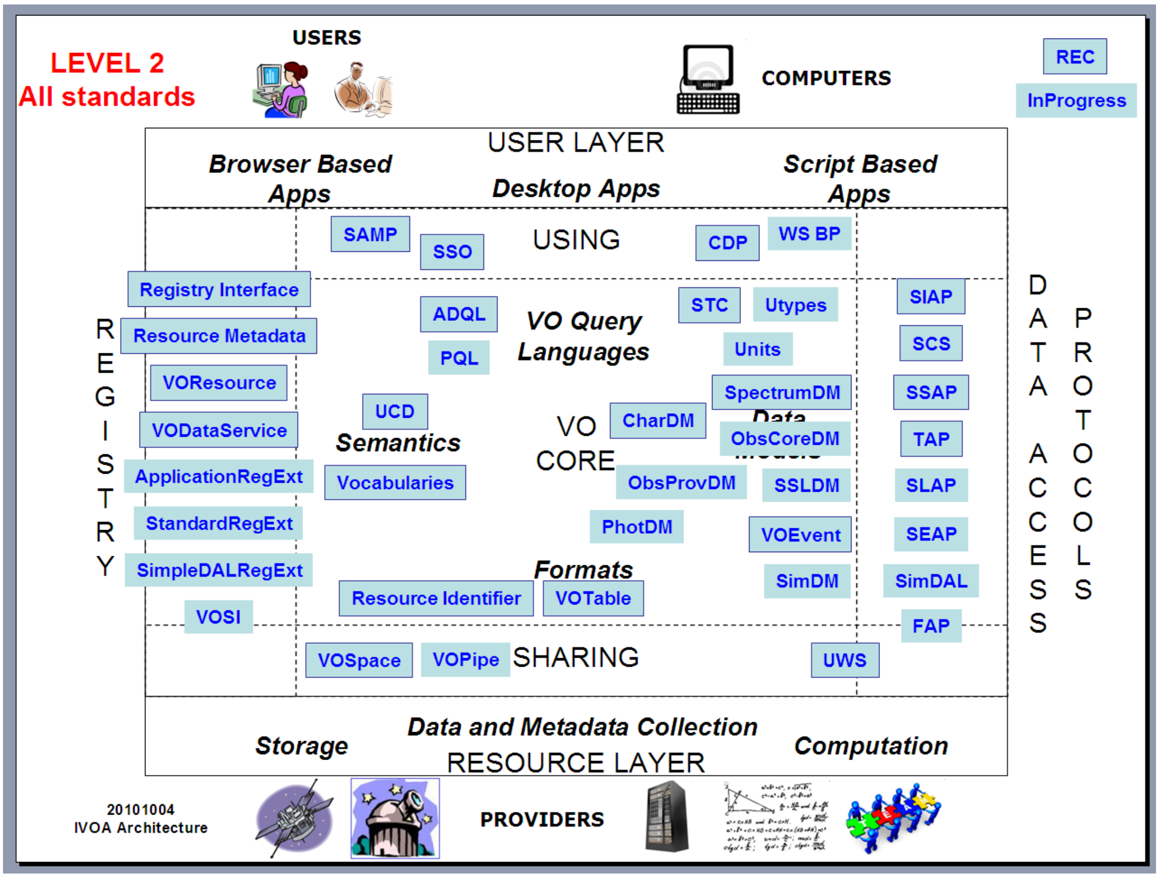

Figure 1: IVOA Architecture at Level 2

2. Registries

2.1. Introduction

Registries are the primary service to access metadata in Virtual Observatories and uses the IVOA standards for the Registry group. Metadata is defined by using a standard set of XML schemas with primary core metadata defined as a 'Resource' and other extension metadata is defined in the IVOA for particular Services and other types of Resources. The following section defines IVOA registry and reviews the feedback with respect to the HELIO implementation for the HELIO project.

2.2. Definition of Registries

The IVOA Registry allows astronomers to locate, get details of, and leverage any resource located anywhere in the IVO space – e.g in any Virtual Observatory. The IVOA defines the protocols and standards used by different registry services to interoperate and thereby realize this goal.

2.3. Support Interface (VOSI) and CatalogService vs. DataCollection

- DataCollection – Metadata that defines data sets and table metadata information.
- CatalogService – Metadata that defines data sets, catalogues, and table metadata, which adds extra metadata on interface location to serve the catalogue service.

Support interfaces known as Virtual Observatory Support Interfaces (VOSI) are installed on the HELIO and EUROPLANET services to extract table metadata and capability information, which is the information about the location of the services. All services use a '/vosi/capability' and '/vosi/tables' support interface to view and harvest metadata. HELIO placed fine granular metadata of tables inside the registry, recent CASSIS meetings have showed clients do not search on such a fine grained metadata and clients prefer to access the '/vosi/tables' for the information. EUROPLANET resources in the Registry submit Space-Time Coordinates metadata, CASSIS interactions have shown this is metadata that is rarely searched and could benefit from removal of the Registration and allow clients to gain access through the VOSI interfaces similar to the table metadata.

The HELIO implementation uses CatalogServices to describe the table metadata with relationships to a CatalogService metadata that could query on the particular catalogue. HELIO framework requires a script to extract needed information from the master table and upload the information to the registry. CASSIS meeting showed that HELIO registrations did not include 'capability' for each CatalogService, but a relationship. It was thought this is the correct approach for the registration, but clients do not use the 'relationship' tags to look up locations of a service and HELIO registrations should include both the 'relationship' and 'capability' metadata.

Issue: The HELIO registries can automatically harvest support interfaces, but the DataCollection does not allow defining a URL to the table metadata.
Solutions, currently '3' is the chosen solution for HELIO:

1.) Have all table metadata currently defined as DataCollection be defined as a CatalogService.
    a. The problem with this solution is all the tables in the dataset will be pointed to the same CatalogService, which is redundant. A Resource to describe the tables is not a CatalogService, but metadata describing the tables. The CatalogService should only need to be defined once.
2.) Have DataCollection metadata allow a support interface URL to be defined; this would allow harvesting a particular table Resource.
    a. HELIO believes this is an ideal solution; IVOA is in discussion about a possible change in the future for the DataCollection metadata to allow this ability. Ongoing discussions are taking place.
3.) Produce a specific solution that will harvest the appropriate metadata and upload the contents into the Registry.
    a. This is the current solution implemented in the HELIO framework. A script is run that looks up the necessary support interface table metadata and uploads the data into the registry.

2.4. Mirrors, Primary, and Secondary URLS in the Registry

Registry metadata allows the ability to define mirrors as added capabilities in the Resource metadata.

```
<capability standardID="ivo://helio-vo.eu/std/FullQuery/v1.0">
    <interface xsi:type="vs:ParamHTTP">
       <accessURL use="full">
              http://msslkz.mssl.ucl.ac.uk/helioics/HelioQueryService
      </accessURL>
      <accessURL use=”full”>
              http://mirror.service/helio-ics/HelioQueryService
       </accessURL>
   </interface>
</capability>
```

Issue: In the HELIO infrastructure the client software expects each of the URLs to be valid; the Registry does not support a flag or status metadata to indicate problems or downtime of a URL. This requires clients of the registry to contact a monitoring service or come up with a different solution for problematic URLs for the services.

The solution feedback to the IVOA is to add a ‘status’ flag on the individual URL. HELIO has extended the registry to support a status flag along with other attributes on the query interface to determine if the Query Interface supports certain extensions, such as SQL.

```
<capability standardID="ivo://helio-vo.eu/std/HELIO-TAP"
xsi:type="htxs:HELIOTAP">    <interface>
     <accessURL asyncservice="false" sqlenabled="true" status="active" use="full">
http://msslkz.mssl.ucl.ac.uk/helio-ics/HelioQueryService </accessURL>
     <accessURL asyncservice="false" sqlenabled="true" status="active" use="full">
http://helio.ukssdc.ac.uk/helio-ics/HelioQueryService </accessURL>
  </interface>
 </capability>
```

Example of HELIO extended registration.

```
<capability standardID="ivo://ivoa.net/std/TAP" xsi:type="tr:TableAccess">
<interface role="std" xsi:type="vs:ParamHTTP">
<accessURL use="full">http://natzgul.oats.inaf.it:8080/epntap</accessURL>
</interface>
<dataModel ivo-id="ivo://ivoa.net/std/ObsCore-1.0">ObsCore-1.0</dataModel>
<language>
<name>ADQL#v2.0</name>
<version ivo-id="ivo://ivoa.net/std/ADQL#v2.0">v2.0</version>
<description>ADQL#v2.0</description>
</language>
<outputFormat ivo-id="ivo://ivoa.net/std/TAPRegExt#output-votable-td"/>
<uploadMethod ivo-id="ivo://ivoa.net/std/TAPRegExt#upload-inline"/>
<uploadMethod ivo-id="ivo://ivoa.net/std/TAPRegExt#upload-http"/>
</capability>
```

Example of standard TAP registration in EUROPLANET registry. Both HELIO and EUROPLANET supports TAP capability.

3. UCD and UTypes

3.1. Introduction

Universal Content Descriptor and UTypes are both used in the HELIO framework. UTypes have no formal document definition in the IVOA, but are heavily used in the IVOA and are defined for the particular standard interfaces. UTypes allow the ability to define information to the data models and lends itself to more detailed descriptions. Universal Content Descriptor (UCD) is a more formal vocabulary in the IVOA defining a restricted set of terms. UCD does not specify units, unlike the ability of a UType.

3.2. Definition

The Unified Content Descriptor (UCD) is a formal vocabulary for astronomical data that is controlled by the International Virtual Observatory Alliance (IVOA). The vocabulary is *restricted* in order to avoid proliferation of terms and synonyms, and *controlled* in order to avoid ambiguities as far as possible. It is intended to be flexible, so that it is understandable to both humans and computers. UCDs describe astronomical quantities, and they are built by combining words from the controlled vocabulary.

UTypes are a core part of the IVOA Architecture (see Fig. 1). Although they currently lack a formal definition within the IVOA (http://www.ivoa.net/documents/Notes/UTypesUsage/), they are employed in a number of situations. The most general usage is as an identifier for a concept defined within an IVOA data model, i.e., a label carrying no more than nominative semantics. A more specific usage is as a pointer (parsable identifier) to a data model concept, semantically equivalent to a URI or XPath in XML. There are also related practices about reuse, inheritance, extensibility, etc.

3.3. Solar input to IVOA

Baptiste Ceconi (OBSPARIS) has submitted talks and a vocabulary for Solar Physics used in HELIO to formally have it accepted to the IVOA.

Anja LeBlanc (UNIMAN) initially constructed UTypes based on the HELIO data model, they have yet to be suggested to the IVOA as UTypes are more directed to our own HELIO data model. HELIO Utypes correspond to our own namespace defined with a 'helio:' prefix. Below is a list of our current primary UTypes.

- Time – UCD: time.phase Utype: helio:time.time
- Time_Start – Utype: helio:time.time_start.??
- Time_End – Utype: helio:time.time_end.??
- Time_Peak – Utype: helio:time.time_peak.??
  ?? – A choice can be (sample of time_start): remote_time_start, julian_time_start, insitu_time_start, first_observation_time, predicted_time_start, julian_time_start
- Julian Integer for a Trajectory – Utype: helio:trajectories.julian_date.julian_date_int
- **(julian_date may be int, double, whole, fraction and is also associated with the jultian_time_start, julian_time_end)
- Longitude Hg – Utype: helio:location.long_hg
- Latitutde hg – Utype: helio:location.latitude_hg

HELIO has developed an extensive data model with corresponding list of Utypes – see the page under URL http://www.helio-vo.eu/documents/helio_documents.php

4. VOTable

VOTable is used as the return for all HELIO Query Interfaces and EUROPLANET EPN-TAP services, HELIO and EUROPLANET was able to use the standard ‘as is’. It should be noted that an important feature of the HELIO Query Interface is the use of dates and datetime data types for querying. VOTable by default does not support date or datetime, it must be defined as a column with a specific xtype attribute that clients can use to understand particular date return types from a column.

4.1. Definition

The VOTable format is an XML standard for the interchange of data represented as a set of tables. In this context, a table is an unordered set of rows, each of a uniform structure, as specified in the table description (the table *metadata*). Each row in a table is a sequence of table cells, and each of these contains either a primitive data type, or an array of such primitives. VOTable is derived from the Astrores format (see IVOA References), itself modeled on the FITS Table format (see IVOA References); VOTable was designed to be close to the FITS Binary Table format.

4.2. Sample

Below is a sample VOTable from the HQI showing a datetime return type.


```
<VOTABLE version='1.2' xmlns="http://www.ivoa.net/xml/VOTable/v1.2">
<RESOURCE>
<DESCRIPTION>Helio ICS time based query</DESCRIPTION>
<INFO name="QUERY_STATUS" value="OK"/>
<INFO name="EXECUTED_AT" value="2013-03-04 23:20:58"/>
<INFO name="MAX_RECORD_ALLOWED" value="5000"/>
<INFO name="QUERY_STRING" ></INFO>
<INFO name="QUERY_URL" ></INFO>
<TABLE name="ics-instrument_observatory">
<FIELD arraysize="*" datatype="char" name="name">
<DESCRIPTION>Instrument Name</DESCRIPTION>
</FIELD>
<FIELD arraysize="*" datatype="char" name="observatory_name">
<DESCRIPTION>Observatory Name</DESCRIPTION>
</FIELD>
<FIELD arraysize="*" datatype="char" name="obsinst_key">
<DESCRIPTION>Helio Instrument Name</DESCRIPTION>
</FIELD>
<FIELD arraysize="*" datatype="char" name="experiment_id">
<DESCRIPTION>Experiment ID</DESCRIPTION>
</FIELD>
<FIELD arraysize="*" datatype="char" name="time_start" xtype="iso8601">
<DESCRIPTION>Instrument Start Date</DESCRIPTION>
</FIELD>
<FIELD arraysize="*" datatype="char" name="time_end" xtype="iso8601">
<DESCRIPTION>Instrument End Date</DESCRIPTION>
</FIELD>
<FIELD arraysize="*" datatype="char" name="longname">
<DESCRIPTION>Instrument Full Name</DESCRIPTION>
```

```
</FIELD>
<DATA>
<TABLEDATA>
  <TR>
    <TD>RSS</TD>
    <TD>Voyager-2</TD>
    <TD>VOYAGER_2__RSS</TD>
    <TD>1977-076A-02</TD>
    <TD>1977-08-20T00:00:00</TD>
    <TD>2020-01-01T00:00:00</TD>
    <TD>Radio Science</TD>
  </TR>
  </TABLEDATA>
</DATA>
</TABLE>
</RESOURCE>
</VOTABLE>
```

5. Query Interface

HELIO Query Interface (HQI) defines a specification on how to interact with the HELIO Query Component and exactly what types of Query languages can be submitted and the expected output (VOTable) is returned. The Query Language followed the specifications of the Parameter Query Language (PQL) and the specification defined in the Table Access Protocol (TAP). TAP defines two query languages, a required Astronomical Data Query Language (ADQL) and an optional PQL. ADQL moved from an XML-SQL based structure towards string based SQL used for relational databases with a few extensions for particular queries on coordinates and location. ADQL was added to the HELIO Query Interface at a later stage making the HQI compatible with the IVOA TAP specification. Majority of queries to the HQI are simplistic time-based queries allowing PQL and basic Parameter type searches to be performed. The PQL style queries have an advantage of not requiring a relational database. This is commonly used in the most popular HELIO component 'Data Provider Access Service' (DPAS) which acts as a façade layer to many external services. Clients get an HQI interface on the DPAS to query other data providers.

EUROPLANET uses Planetary Data Access Protocol (PDAP) and was the result of the International Planetary Data Alliance (IPDA) who created a set of Planetary Data Standard (PDS). The PDAP protocol has both a metadata and a data retrieval layer. EUROPLANET later adopted the use of TAP as a data access query layer. EUROPLANET has been fully adopting the TAP services as a brand EPN-TAP and conforms fully to the ADQL.

IVOA TAP protocol has focused primarily on ADQL and relational database concepts with PQL optional and becoming dormant the last two years in the IVOA. Kevin Benson gave a talk to the IVOA that PQL should still be considered as an easy mechanism for non-relational databases and to still be a part of the TAP protocol. It is foreseen that the TAP specification will continue to use PQL as optional and ADQL as a required language. ADQL was added in the final 6 months of HELIO to allow IVOA client applications using TAP to access the HELIO services. Feedback was given to IVOA in regards to registration of TAP services into the IVOA Registry.

6. Asynchronous calls to Applications and Query Interface

HELIO allows clients to call particular command line applications such as Coordinate Conversion, Flare Plotter, Goes Plotter, and Parker Model. Applications have an interface that conforms to the IVOA Universal Worker Service (UWS) for the submission of jobs. UWS interface was incorporated into an older Astrogrid component called Common Execution Architecture (CEA) and was installed on various services in the HELIO infrastructure for calling the applications. CEA component is an older and unsupported component with non easy-to-use setup, Kevin Benson has been in discussions and Carlo Zwolf from Observatoire of Paris, who has developed a Parameter Description Language (PDL), which conforms to the UWS interface. The PDL has an easy to use GUI setup and has been developed with Grid and Cloud use in mind along with Parameters able to depend on constraints of other input parameters unlike the CEA component. One change performed on the CEA component was to allow the saving to an internal storage mechanism. CEA by default required saving to an external storage i.e. a VOSpace/FTP server though it has the capability of saving the files internally. HELIO changed the mechanism to save internally to the server and return output of results to be a 'http' URL reference to the internal storage. CEA component meets the needs for HELIO for the remainder of the project. PDL has its own job language and is currently as a proposal to the IVOA, at a later stage if PDL becomes a recommendation in the IVOA then a move to PDL is foreseen in the future.

EUROPLANET and HELIO both support Asynchronous Query Interface calls to the datasets with the use of IVOA-TAP.

7. References for IVOA Standards

7.1. VOTable

Reference:
http://www.ivoa.net/Documents/VOTable/20091130/

Description

The VOTable format is an XML standard for the interchange of data represented as a set of tables. In this context, a table is an unordered set of rows, each of a uniform structure, as specified in the table description (the table *metadata*). Each row in a table is a sequence of table cells, and each of these contains either a primitive data type, or an array of such primitives. VOTable is derived from the Astrores format (See IVOA References), itself modeled on the FITS Table format (See IVOA References); VOTable was designed to be close to the FITS Binary Table format.

Astronomers have always been at the forefront of developments in information technology, and funding agencies across the world have recognized this by supporting the Virtual Observatory movement, in the hopes that other sciences and business can follow their lead in making online data both *interoperable* and *scalable*.

VOTable is designed as a flexible storage and exchange format for tabular data, with particular emphasis on astronomical tables.

Interoperability is encouraged through the use of standards (XML). The XML fabric allows applications to easily validate an input document, as well as facilitating transformations through XSLT (eXtensible Style Language Transformation) engines.

Usage in HELIO and EPN-TAP

The primary response from all HELIO Query Interfaces (HQI) and EPN-TAP services is a VOTable format.

7.2. Table Access Protocol

Reference:
http://www.ivoa.net/Documents/TAP/20100327/

Description

Table Access Protocol (TAP) defines a service protocol for accessing general table data, including astronomical catalogs, as well as general relational database tables. Access is provided for both database and table metadata, as well as for actual table data. This version of the protocol includes support for multiple query languages, including queries specified using the Astronomical Data Query Language (ADQL [1]) and the Parameterised Query Language (PQL, under development) within an integrated interface. It also includes support for both synchronous and asynchronous queries. Special support is provided for spatially indexed queries using the spatial extensions in ADQL. A multi-position query capability permits queries against an arbitrarily large list of astronomical targets, providing a simple spatial cross-matching capability. More sophisticated distributed cross-matching capabilities are possible by orchestrating a distributed query across multiple TAP services.

Usage in EUROPLANET

TAP has been implemented in EUROPLANET using the GAVO (German Astronomical Virtual Observatory) framework, which is referred to as EPN-TAP. TAP conforms to a set of specific core parameters that 'MUST' be defined in the TAP services. EPN-TAP using the GAVO framework carries out position-based searches.

Usage in HELIO

HELIO Query Interface originally defined its own interface and conformed to the Parameter Query Language (PQL) as defined in the Table Access Protocol (TAP) with HELIO defining its own extensions. HELIO at a later stage in the project conformed to the total TAP protocol including Astronomical Data Query Language (ADQL) to allow other IVOA clients that query on TAP to be able to use the HELIO services. HELIO defines a set of core parameters for common queries.

Sharing of Resources

The set of specific core parameters are currently being reviewed for the assignment of the common naming convention between HELIO and EUROPLANET. Clients of the services could use parameters, such as STARTTIME, ENDTIME interchangeably. Clients that use the parameters will be able to interchange between the two different domains.

7.3. Universal Worker Service and Parameter Description Language

Reference:
http://www.ivoa.net/documents/UWS/index.html
http://www.ivoa.net/documents/PDL/20130718/index.html

Description

UWS:
The Universal Worker Service (UWS) pattern defines how to manage asynchronous execution of jobs on a service. Any application of the pattern defines a family of related services with a common service contract. Possible uses of the pattern are also described.

The need for the UWS pattern was inspired by AstroGrid's Common Execution Architecture and particularly by discussions with Noel Winstanley. The ideas about statefulness are distilled from debates in the Global Grid Forum in respect of the Open Grid Services Infrastructure that was the forerunner of Web Services Resource Framework. The REST binding came initially from suggestions by Norman Gray.

PDL:
In this language sentences and syntactical elements are organized into a rigorous data model. With no loss of generality, we will represent this data model using XML.
It intends to be a description language for self-descriptive web services exposing the semantic nature of input and output parameters, as well as all possible complex constraints. PDL is a step forward towards web services interoperability.

Usage in HELIO

Universal Worker Service specification defines running an asynchronous jobs as a service. Universal Worker Service does not define the job description language to be used. Parameter Description Language is an upcoming language that can be used as a job description language and defines a data model for running jobs as a web service. At this current time the job description language and conforming to the UWS standards was implemented using a tool called Common Execution Service from the Astrogrid project. Current jobs ran at HELIO are Flare Plotting, Coordinate Transformation Services, and GOES Plotter.

7.4. UCD and UTypes

Reference:
http://www.ivoa.net/documents/REC/UCD/UCD-20050812.html
http://www.ivoa.net/Documents/latest/UCD.html
http://www.ivoa.net/documents/Notes/UTypesUsage/20130213/NOTE-utypes-usage-1.0-20130213.html

Description

The Unified Content Descriptor (UCD) is a formal vocabulary for astronomical data controlled by the International Virtual Observatory Alliance (IVOA). The vocabulary is *restricted* in order to avoid proliferation of terms and synonyms, and *controlled* in order to avoid ambiguities as far as possible. It is intended to be flexible, so that it is understandable to both humans and computers. UCDs describe astronomical quantities, and they are built by combining words from the controlled vocabulary.

UTypes are a core part of the IVOA Architecture (see Fig. 1). Although they currently lack a formal definition within the IVOA (http://www.ivoa.net/documents/Notes/UTypesUsage/), they are employed in a number of situations. The most general usage is as an identifier for a concept defined within an IVOA data model, i.e. a label carrying no more than nominative semantics. A more specific usage is as a pointer (parsable identifier) to a data model concept, semantically equivalent to a URI or XPath in XML. There are also related practices about reuse, inheritance, extensibility, etc.

We note that the IVOA Working Draft on UTypes presents an approach to defining UTypes within a broader context of standardizing data model definition and serialization. UTypes are defined as data model labels that point to their associated data model element — they are a string representation of the logical path through the classes and attributes in a UML representation of a data model from the main data model element to a particular part of the data model. A generating syntax is proposed based on this premise and the resulting usage patterns for UTypes in data model (de-)serialization described. This is, however, work in progress and does not necessarily reflect the existing community of practice, which this document seeks to capture.

Usage in EUROPLANET

As in HELIO, Utypes were defined in the EPN-DM (EUROPLANET Data Model) and through the GAVO framework; responses of the VOTable can be shown by the EUROPLANET services. EURPLANET focused on the schema of the data model, which can be found here: http://www.europlanet-idis.fi/documents/public_documents/EPN-DM-v2.0.pdf

Usage in HELIO

All VOTables returned from HELIO Query Interface will have certain columns defining UCD and Utypes. UCD are a standard controlled vocabulary, whereby UTypes are a type of data model to be used by the community. HELIO defined its own data model of Utypes to better express concepts in the HELIO community and to allow columns to be more specific for clients to better understand and analyze on the responses from the HQI.

Sharing of Resources

HELIO and EUROPLANET data models share common UCD and Utypes. The main particular difference is in the namespaces. Individual names are different, though their semantics remains the same. Both HELIO and EUROPLANET are to agree on a common set of Utypes and UCDs, the advantage of the common naming convention is that clients can query both domains and have common understanding of the responses.

7.5. Registries

Reference:
http://www.ivoa.net/documents/RegistryInterface/20091104/
http://www.ivoa.net/documents/latest/RM.html
http://www.ivoa.net/documents/latest/IDs.html

Description

Metadata:
An essential capability of the Virtual Observatory is a means for describing what data and computational facilities are available where, and once identified, how to use them. The data themselves have associated metadata (e.g., FITS keywords), and similarly we require metadata about data collections and data services so that VO users can easily find information of interest. Furthermore, such metadata are needed in order to manage distributed queries efficiently; if a user is interested in finding x-ray images there is no point in querying the HST archive, for example. In this document we suggest the architecture for resource and service metadata and describe the relationship of this architecture to emerging Web Services standards. We also define an initial set of metadata concepts.

Interface:
In the Virtual Observatory (VO), registries provide a means for discovering useful data and services. To make discovery efficient, a registry typically represents to some extent a centralized warehouse of resource descriptions; however, the source of this information—the resources themselves and the data providers that maintain them—are distributed. Furthermore, there need not be a single registry that serves the entire international VO community. Given the inherent distributed nature of the information used for resource discovery, there is clearly a need for common mechanisms for registry communication and interaction.

7.6. Simple Access Messaging Protocol (SAMP)

Reference

http://www.ivoa.net/documents/SAMP/20120411/index.html

Description

SAMP is a messaging protocol that enables astronomy software tools to interoperate and communicate. IVOA members have recognised that building a monolithic tool that attempts to fulfil all the requirements of all users is impractical, and it is a better use of our limited resources to enable individual tools to work together better. One element of this is defining common file formats for the exchange of data between different applications. Another important component is a messaging system that enables the applications to share data and take advantage of each other's functionality. SAMP supports communication between applications on the desktop and in web browsers, and is also intended to form a framework for more general messaging requirements.

SAMP applications currently registered are located here:
http://wiki.ivoa.net/twiki/bin/view/IVOA/SampSoftware
All the applications can communicate with each other under the messaging protocol.

Usage in EUROPLANET

SAMP is used by the web clients of EUROPLANET to send VOTables to particular VO applications for more analysis in plotting. VOTables that reference FITS Images may also be used to send Images to SAMP enabled applications. Two popular SAMP applications are:
TopCat -- can read VOTables and has special plotting facilities
Aladdin – can read FITS Images for analysis.

Usage in HELIO

HELIO has not implemented SAMP on the current Front End, but is expected to do so in the near future. HELIO developed a plug-in for Taverna Workflow system, and in conjunction with the SpanishVO, who developed a plugin AstroTaverna, are able to send results from HELIO Query Interface to SAMP-enabled applications.

Sharing of Resources

This is a communication protocol that offers many advantages in all of the VO domains. No sharing is needed.

8. Taverna – Example of Client crossing domains

Reference

http://www.taverna.org.uk

Description

Taverna is an open source and domain-independent Workflow Management System – a suite of tools used to design and execute scientific workflows and aid *in silico* experimentation.

Taverna allows using various web services and creating your own personal plugin. Taverna hosts another MyExperiment website allowing workflows to easily be shared between communities.

Usage

HELIO plugin was created to give users an easy access to use the Query Services and access Applications via the UWS. As both EUROPLANET and Helio use IVOA TAP protocols it was later demonstrated that the HELIO plugin was able to query on EPN-TAP services. This allows more complex workflows to be built using both the EUROPLANET and HELIO domains as well as other communities.