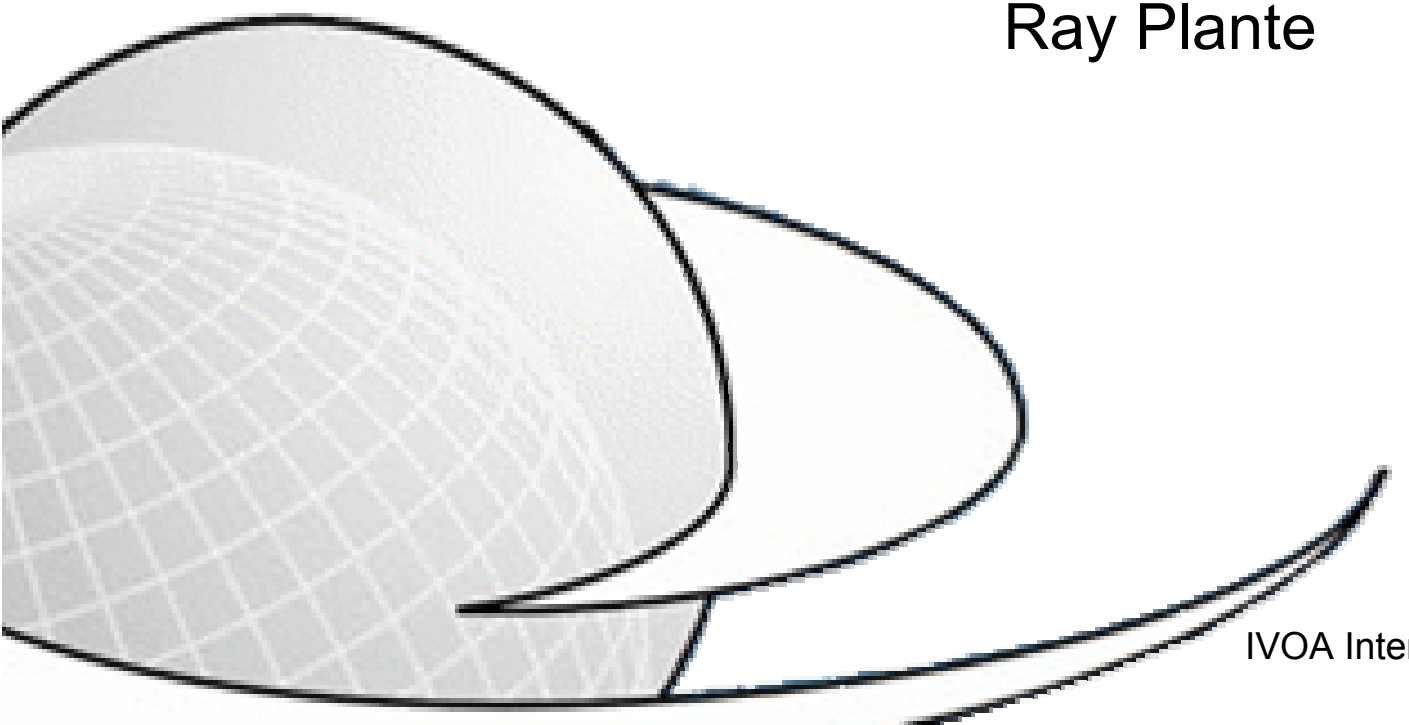
THE INTERNATIONAL VIRTUAL OBSERVATORY ALLIANCE

Ancillary Metadata Access

Graininess III

Ray Plante



17 May 2007

IVOA Interoperability Meeting – Beijing

Towards a solution to the richness controversy

- Registries affect each other via the metadata they put on the harvesting stream.
  - Coarse-grained registry must take some responsibility for fine-grained information they harvest
  - Pressure to support fine-grained metadata if a particular application relies solely on the registry to get that information
- Can we pull this information out of the harvesting stream?
  - If so, how do we get at this information?
  - How can we get it to clients that want to make use of it
- Can we improve the capturing of information by pulling it from the service?
  - If the metadata is generated on-the-fly from the implementation itself…
    - it should be more up-to-date
    - it may be more accurate

Levels of Disruption

- How “disruptive” will a solution be?
  - Does it require updates to the XML Schemas? In the VOResource core? In an extension?
    - A little change: add one or a few metadata
    - A big change: restructuring the data model
  - Does it require an update to RI?
    - Version 2.0?
    - Can solution be built on top of current standards?
  - Does it require a certain level of uptake by publishers?
- Schedule the deployment of the solution according to its level of disruption
  - Possible to make simple fixes right now (?)
  - More transition time for Highly disruptive solutions

  Can’t ask our registry providers to re-implement again in 6 months!
- Does “pretty good” need to be the victim of the “perfect”?

Solution 1: VOSI getRegistration()/getMetadata()

- *Deprecated*
- VOSI defined 2 metadata-related functions:
  - getRegistration() return coarse-grained VOResource record
  - getMetadata() return full (fine-grained) description as VOResource record.
  - Registry calls either method depending on how much information is desired:
- Problems:
  - Does not address controlling what gets onto harvesting stream
  - Does not allow registry to choose which detailed information it wants to pull in.
  - Some resource level perhaps better created via registry input forms.
- Notable advantage:
  - VOSI provides means for retroactively attaching methods to existing services (standard or custom).
  - Low disruption: sits on top of RI/VOResource standards

Solution 2: add table metadata pointer URL

- Small change to VODataService schema:
  - Add an element, <tableMetadataURL>, that can point to a full table & column description in a standard format
    - VODataService model
    - VOTable
  - Requires that we agree to migrate to using pointer rather than in-line descriptions
- Advantages:
  - Low disruption: could apply right away?
  - Smooth transition path without interruption of capabilities.
    - URL can point to a CGI script for dynamic generation at service
    - Proxy services can be pointed to translate for existing services
    - Applicable to non-services
  - Model for application to other kinds of fine-grained metadata?
- Disadvantages:
  - Arbitrary?
  - More general solution might be developed with more careful design

Solution 3: Generalized metadata access interface

- VOSI-like capability for getting detailed metadata
  - Provided by a service
  - getMetadata(*setname*) returns metadata in well-defined (VOResource-based) formats.
    - Capabilities → <capability>
    - Table metadata → <catalog>
  - We restrict VOResource records on the harvesting stream to some core set
  - Registries pull what ancillary metadata from the service they want
    - Publication time
    - Harvest time
  - Registry can return "rich" records to search clients
- Advantages
  - Medium disruption
    - Can build on top of existing standards (?)
    - Will require further design
  - Can be generalized to any *Metadata Store*