

# The architecture of a multi-tiered virtual observatory

Todd King · Jan Merka · Raymond Walker ·  
Steven Joy · Tom Narock

Received: 6 August 2007 / Accepted: 4 February 2008 / Published online: 13 March 2008  
© Springer-Verlag 2008

**Abstract** Virtual observatories are being established in a wide range of disciplines, supported by a variety of agencies. Groups such as the International Virtual Observatory Alliance (IVOA), Planetary Data System (PDS) and the Space Physics Archive Search and Extract (SPASE) consortium are defining metadata standards to aid in archiving and sharing of information resources. The role of the virtual observatories in this resource sharing environment is to locate available resources and help users find the resources they need and then gain access to those resources. There are many different existing resource providers from which virtual observatories must collect descriptions of their resources. These resource providers may have associations with other providers so the topology of information exchange can be complicated. We explore the variety of topologies that can exist and discuss methods of collecting (harvesting) information from providers such as scheduled and on-demand harvesting. We compare the benefits of each approach and look at the issues of management overhead, adaptability and timeliness. We also

explore the benefits of combining searching and harvesting services as part of a comprehensive solution.

**Keywords** Harvesting · Search registry · Topology · Virtual observatory

## Introduction

Virtual observatories are quickly becoming a key component of science research. Virtual observatories with well defined scope are being formed to provide scientists with access to the large and growing body of data. For example, NASA's Heliophysics Virtual Observatory program consists of six complementary virtual observatories organized to provide access to data related to the Sun, the upper atmosphere and the environment in between. The individual domains for the Virtual Observatories are; Solar (VSO), Heliospheric (VHO), Magnetospheric [(VMO located at UCLA and Goddard Space Flight Center (GSFC)], Ionospheric–Thermospheric–Mesospheric (VITMO), and Radiation Belts (ViRBO). It is expected that additional virtual observatories will emerge for other domains. Each virtual observatory functions as a port of entry to resources providing to the user a single, coherent view of available resources which are located in a distributed environment.

How a virtual observatory gathers the information about available resources, a process called “harvesting” is a subject of much discussion in the virtual observatory community. One approach is to have the virtual observatory generate descriptions for each resource. This is the way most virtual observatories get started. While this “bootstrapping” approach is excellent at the beginning since it lets them create the metadata quickly without waiting for data providers to learn the metadata standards, it does not

---

Communicated by P. Fox

T. King (✉) · R. Walker · S. Joy  
Institute of Geophysics and Planetary Physics,  
University of California,  
Los Angeles, CA, USA  
e-mail: tking@igpp.ucla.edu

J. Merka · T. Narock  
Goddard Earth Science and Technology Center,  
University of Maryland Baltimore County,  
Baltimore, MD, USA

J. Merka · T. Narock  
NASA/Goddard Space Flight Center,  
Heliospheric Physics Laboratory,  
Greenbelt, MD, USA

readily scale to an operational environment. As the number of resources increases so does the burden on the virtual observatory. In addition the extent of the metadata is not constant and continually needs to be updated because the data holdings are dynamic. There is also a political aspect to virtual observatories. Organizations and agencies in different countries must adhere to local regulations and expectations, and therefore must maintain control on the descriptions of their resources.

An alternative approach to centralized generation and management of resource descriptions is to distribute the task and respect the natural autonomy inherent in the current network of providers. In this paper we explore methods of harvesting resource descriptions in a distributed environment which function equally well independently of the topology of the provider network. We will also explore how searching can be combined with harvesting in order to bring timely results and provide users with the best data resources available. A number of approaches have been suggested for harvesting metadata. We compare the described methods to those from The Open Archives Initiative (OAI)–Protocol for Metadata Harvesting (OAI-PMH; Lagoze et al. 2004), the International Virtual Observatory Alliance (IVOA) Astronomical Data Query Language (ADQL) (Ohishi and Szalay 2005), the ANSI/NISO Z39.50/ISO-23950 (Z39.50-2003 2003) protocol and Contextual Query Language (LOC 2007). We will conclude with looking at how the virtual observatory architecture compares to the idealized reference model of the Open Archival Information System (CCSDS 2002; ISO 2003) architecture.

## Materials and methods

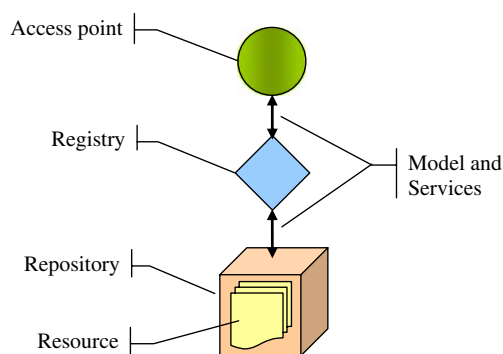
The goal of any virtual observatory is to provide complete and accurate information to the user, irrespective of the underlying topology or organization of the virtual observatory. On the other hand, the virtual observatory needs to harvest information from many providers and there are many different types of providers each with different capabilities. A provider can be a single researcher with a unique resource and simple services (FTP or HTTP), an instrument facility with a wide range of resources and services, or a project, mission or observatory with a large assortment of resources and services. A provider may also be another virtual observatory. A provider can simply offer “as-is” resources to the community. Descriptions of those resources may be generated by the provider or by an intermediary, such as a virtual observatory. Descriptions of resources are harvested and stored in a registry. This registry can exist at the data provider’s site or at a virtual observatory. A registry can be harvested by other registries, so a virtual observatory can be harvested by another virtual

observatory. A generic model for a virtual observatory has the following basic components:

- Resource: An object (document, data, etc.) or service available for use.
- Repository: A facility for storing and maintaining digital information in an accessible form.
- Registry: A collection point for metadata about resources.
- Access Point: An interface to the registries and resources.

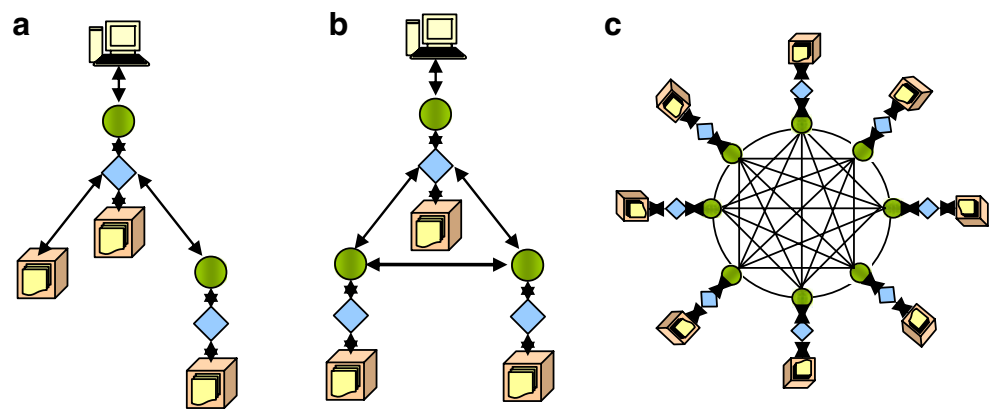
The connection between a repository and a registry, as well as from the registry to the access point is through common data models and services. This is depicted in Fig. 1. An access point can have multiple interfaces, but every access point must have at least one common interface to support the sharing of information with other systems.

Information flow through the system begins at the resource level. For each resource a description is created in the chosen data model. This description may be stored along with the resource in the repository or stored in a separate description repository. A registry is populated by harvesting the resource descriptions from one or more repositories and transforming those descriptions into a searchable collection. The service used to harvest the resource description can include scanning a local file system, accessing an FTP server, downloading a file from a web server, or transferring information stored in an external system. Once the resource descriptions are included in the registry they are accessed through a standard interface at the access point. If this standard interface provides resource descriptions then a registry service can be harvested just like any other repository. This allows the formation of multi-tiered networks of registries like the ones depicted in Fig. 2.



**Fig. 1** Schematic of the simplest registry service. A ball represents an access point (API, User Interface, etc). *Diamonds* represent registries which store information about resources and provide search and selection services, *boxes* represent repositories which store individual resources, *arrows* represent communication between each component that is based on well defined models and services

**Fig. 2** Various multi-tier network topologies. Any combination of repositories, registries and access points are allowed. See Fig. 1 for an explanation of each icon. **a** Multiple repositories, single registry. **b** Three fully interconnected nodes. **c** Fully interconnected network of multiple nodes



The process of harvesting centralizes the resource descriptions from all sources at the harvesting registry. For linear networks of registry nodes (Fig. 2a) one node harvests local resources from one or more other nodes. In this scenario centralization may be desirable since the number of sources is small and jurisdiction can be easily arranged. However, if the network of registry nodes is more complicated or a user interacts with multiple access points the task of harvesting becomes more difficult. Figure 2c shows the extreme scenario where every registry node is connected to every other registry node. In this scenario jurisdiction and reconciliation of holdings becomes a complex task. While this scenario may be rare a more common scenario is like the one for the VMO where there are at least two peers which share all resource descriptions. For example we expect that the VMO at UCLA will provide access to the same set of resources that the VMO at GSFC will provide. This is similar to the scenario with international or multi-agency peers. Adding an international peer to the VMO results in the topology depicted in Fig. 2b.

Using the simplest interconnected topology of two nodes, let us explore what occurs during the harvesting of resources from registries. Call the nodes A and B. Initially A and B have different registries. When A harvests from B then those resources available at B will be merged with those at A. When B harvests from A then B will receive all of the resources from A including both those originally at A and those harvested from B. For each harvest cycle this would be repeated and both A and B would incrementally grow creating a “race condition” where resource descriptions are continually repeated. There are three approaches (rules) to eliminate or at least mitigate the race condition.

1. Copy rule: Maintain a distinction between “local” and “remote” harvests. The disadvantage to this approach is that success depends on the topology. In the example topology this will have the desired results since it's easy to distinguish local from remote. However, if the topology of the network connected to one of the nodes consists of other nodes, then the harvesting node must

visit each of the nodes of the topology. This may not be desirable since in more complex topologies a node may include a direct reference to one of the others such as in the topology in Fig. 2b. This situation is prone to a race condition.

2. Uniqueness rule: Rely on some unique attribute of each resource description and add to the local registry only those items that are new. For instance in space physics resource descriptions based on the SPASE (Harvey et al. 2004) data model assign a unique resource identifier to each resource. However, in this scheme synchronization of content becomes a problem. If after a cycle of harvesting a description is updated at the remote node, then a change would not propagate since the SPASE resource ID lacks resource level versioning information. Retaining source node information in the registry and combining the uniqueness rule plus the copy rule of (1) could result in a complete and accurate registry.
3. Visit once rule: During harvesting maintain a list of visited nodes and only visit nodes once then using the uniqueness rule (2) or the copy rule (1) would result in a complete and accurate registry. The benefit of this “visit once” rule is that it functions independently of the topology. Even in the multi-node, fully interconnected topology shown in Fig. 2c a complete and accurate registry is achieved.

### Topology

There are two general approaches to defining the topology of a collection of nodes. One approach is to use an external definition of the topology. For example, we know that A is connected to B and B is connected to C. With this prior knowledge it is possible to optimize the harvesting and to build in an avoidance of any possible race condition. This approach works well if the topology changes slowly and the topology is relatively simple. If the topology is not well defined a priori then there is a continual cycle of

maintenance required. Another approach is to have each node declare its neighbors. The topology is then “discovered” by starting at any node and recursively traversing the network by visiting the node and each neighboring node. Such a network is self-organizing and requires only minimal management that is localized to a node. It is also very flexible and agile since it can be re-organized to respond to the needs of the local node.

We can illustrate this flexibility by the using the VMO as an example. The peer at UCLA has as one of its responsibilities to interface to ground-based providers. Through agreements with ground-based providers it may declare that some are neighbors since they either operate registries or are virtual observatories established by other agencies. The other peer at GSFC has responsibility for many space-based providers and some of those providers may also establish registries. In both cases providers may have other providers so that the actual topology is free form.

The goal of a virtual observatory is provide quality information to its users and this reliance on others for part of what a virtual observatory provides introduces issues of trust. When a node declares another node to be a neighbor there is an implicit level of trust that the neighbor will provide reliable information. When a neighbor has other neighbors then a node indirectly trusts the neighbor of its neighbor. This may not always be reasonable so harvesting methods must include the capability to set the extent of trust. It should be possible to limit how far from a declared neighbor to allow a harvesting action to travel. In a scenario where the extent of trust is just the first neighbor, then it must be possible for a node to respond with only a “local” resource description. This places a requirement on the registry to either track the extent (or distance) to the source node of all resource descriptions or to harvest each neighbor on-demand, returning only those resources up to the desired extent.

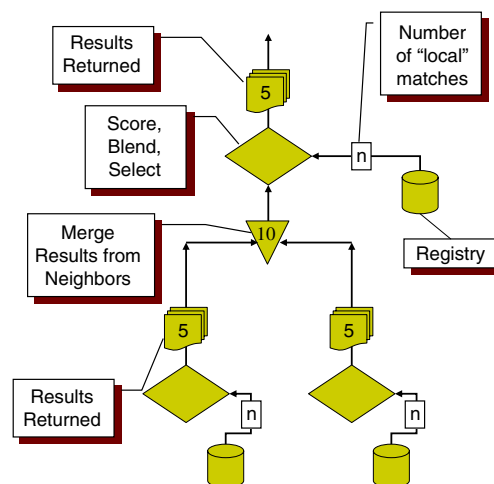
### Harvest on-demand

On-demand harvesting is an alternative to harvesting all resources during a harvest cycle. Instead of collecting resource descriptions from all possible sources during an asynchronous process, a real-time request for resources is passed to each node. As requests are processed at a node, the node will send the request on to its neighbors according to the trust extent constraints. Matching resource descriptions at each node are returned and the results are blended together with local results. Bookkeeping of resource descriptions at each node is greatly simplified since a registry contains only local resource descriptions and pointers to their self-declared neighbors.

If constraints are added to the on-demand harvesting where only those resources which match the constraints are returned then a distributed search is performed. One of the possible constraints could be the number of results to return so that the volume of returned items matches the need. If a relevance scoring technique is used such as the Term Presence-Proximity technique (King et al. 2008) then a federated search, like that depicted in Fig. 3, is possible. As illustrated in Fig. 3 a request is presented to the first node. That node passes the request on to a subsequent node. In the figure its assumed that only the top five matches will be returned. So at each node only the five most relevant matches are selected. As the results are blended at each node, at a subsequent node the relative relevance of each item must be reassessed and the five most relevant resources are passed along. The final results will be the five most relevant resources from the entire network. This approach is scalable to any number of nodes with any possible topology.

A search service for registries can be used for many purposes. One purpose is to perform harvesting whether asynchronous or on-demand. When harvesting the most common scenario is to retrieve resources that were released after a certain date. This allows for incremental synchronization of collection point registries. Another common scenario is to retrieve information for only certain types of resources. For example, suppose there is a registry for observatory information. Such a registry would harvest only resources that describe an observatory.

The main purpose of harvesting is to support user initiated searches. One approach is for the user to provide a set of words and to find all relevant resources. The level



**Fig. 3** A federated search performed on a set of registries. Barrels indicate information stores (registry), arrows the flow direction of information, boxes on each arrow indicate the number of items returned, inverted triangles indicate points where information is merged from multiple sources and diamonds indicate decision making (such as which results to pass on)

of relevance can be adjusted by the user by specifying whether a resource must be relevant to all the words or just some of the words. The user may also be interested only in resources of a certain type. For ease of use and to be concise a user interface typically displays only the top most relevant matches and then allows a user to page through or expand the list. A user may also want to constrain on a specific portion of the resource description. For example, in the SPASE data model the observed region is described by using an enumeration. The ability to constrain on a part of a description must be flexible enough to allow a constraint on any part of the description.

By combining the harvesting and user scenarios we can derive a set of requirements for a search service.

1. The search service will allow locating resources using one or more words as the selection criteria.
2. The search service will allow constraining on any element (node) within the metadata description.
3. The search service will allow constraining the search to a particular resource type.
4. The search service will allow constraining the search to a range of resource release dates.
5. The search service will return a selectable number of most relevant matches.
6. The search service will return the total number of matching resources.

From these requirements we have determined a set of parameters that the search service needs to support:

topLimit: The maximum number of hits to return.

(Req. 5)

resourceType: Constrain the search to a resource type.

(Req. 3)

words: List of words to match. (Req. 1)

match: Indicate whether the system returns matches to all of the words or matches to only some of the words.

(Inferred from Req. 1 and Req. 5)

xquery: Metadata constraints (xquery=criteria) (Req. 2)

fromReleaseDate: Include items released on or after this date. (Req. 4)

toReleaseDate: Include items released prior to or on this date. (Req. 4)

With these requirements in mind let us look at existing search and harvesting services. The Open Archives Initiative (OAI)–Protocol for Metadata Harvesting (OAI-PMH; Lagoze et al. 2004) supports requesting records published within a specific timeframe. It also supports constraints on metadata prefix. It does not provide word based searches or selection based on metadata content. The International Virtual Observatory Alliance (IVOA) Astronomical Data Query Language (ADQL; Ohishi and Szalay 2005) is based on SQL and does support a form of metadata constraint by

using XPath (W3C 1999) to map a metadata node (an element in a metadata document) to a field reference in the SQL query. An XPath is analogous to a file system path for nodes (tags) within an XML document. The reference to each node is separated by a slash (/) delimiter. Within an XPath it is possible to constrain on the value of a node by using relational constraints and on the attributes of a node. However, ADQL does not support XPath constraints, which limits the use of XPath in ADQL to mapping of a node in an XML schema to a conceptual relational schema. The ADQL does not support the concept of word based searches or a release date.

To support all the requirements for searches discussed above both OAI-PMH and IVO-ADQL would need to be modified. With OAI-PMH the core metadata does not contain enough information to support the most basic word searches. In addition, the most scientifically useful information would presumably be contained in the “metadata” section of an OAI description. The content of the “metadata” section is not standardized. Its content is defined by the provider and can have any schema. This adds complexity since multiple models would have to be adopted (OAI plus at least one other data model). With IVO-ADQL a relational schema would have to be defined to support a word search of resources or a new ADQL specific “function” would need to be added to the specification. Even with the addition of a function the table based model of ADQL is inadequate for the implementation of relevance scoring and facet based selection.

Other search and harvesting services are those specified by the ANSI/NISO Z39.50/ISO-23950 (Z39.50-2003 2003) protocol widely used in library environments where it is combined with the Contextual Query Language (LOC 2007) and an attribute set. The Z39.50 protocol is designed for searching and retrieval of information in a distributed network environment independent of the any underlying information schema. Systems (or services) that wish to use the Z39.50 protocol must agree upon and support standardized attribute sets (schema) which are exchanged as part of the query process. The response to a query makes available result sets which contain records that adhere to a particular schema. The Z39.50 is a statefull protocol in that the result sets are maintained at the server and retrieved through post query transactions. Z39.50 supports proximity testing of terms which allows placing constraints on records based on word location and scope within a document. It does not return a relevance score and does not allow dividing the search space into facets.

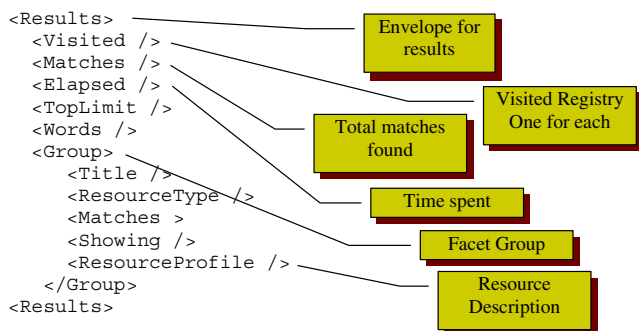
Something new is needed. The VMO has implemented a service which meets the stated requirements. It provides a protocol similar to OAI-PMH, but with increased capabilities and a simpler “language” than ADQL. The protocol is

similar to the Z39.50 protocol, but has a more limited and simpler model. Queries are expressed in a form similar to the Contextual Query Language, but allow elements to be referenced by using a path syntax. Aspects of the query language are described in SPASE Query Language (Narock and King 2008).

The parameters passed to the search service are only half of the picture. When the search service is presented with a request the response is an XML document. An annotated schema of the response is in Fig. 4. This response is processed by the requesting application and can be used for different purposes. For example, in a registry that chains to other registries, the response has sufficient information to blend the responses from multiple sources, reassess the collection of responses (i.e., select a new set of most relevant resources) and send the appropriate response to the query. The response can also be used by a user interface. The query process in a user interface scenario is depicted in Fig. 5. In this “typical” scenario a web form is used to create a REST (Representation State Transfer) query (Fielding 2000). The request is sent to the “local” registry server. The local registry server searches its local registry and in-turn sends the request to other neighbors and blends the results from all sources. The results are returned and ultimately transformed to HTML using an XML style-sheet (XSLT) transformation.

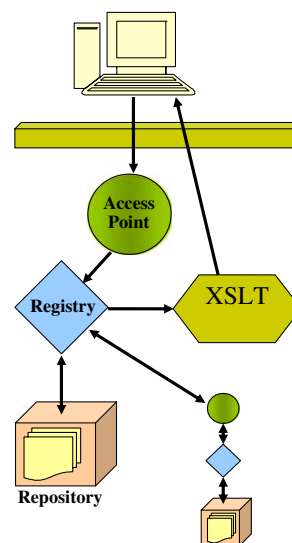
### Archive perspective

A useful reference model for archive systems is the Open Archive Information System (OAIS)/ISO 14721:2003 reference model (OAIS, 2002). From an archive perspective virtual observatories offer an important set of services that connect providers and users (consumers) of resources, this is illustrated in Fig. 6. Virtual observatories also offer services useful in the administration of holdings by giving



**Fig. 4** The annotated schema of a query response. Each response includes information about the visited registries and the total number of matches to the query. Each facet of the response is described in a “group” which provides details on the number of matches in each group and the description of each returned resource

**Fig. 5** A user initiated query process. The user submits a query to the system at an access point which is forwarded to one or more registries. The results are transformed into HTML for viewing by using XML Stylesheet (XSLT). Flow of information is in the direction of the arrows

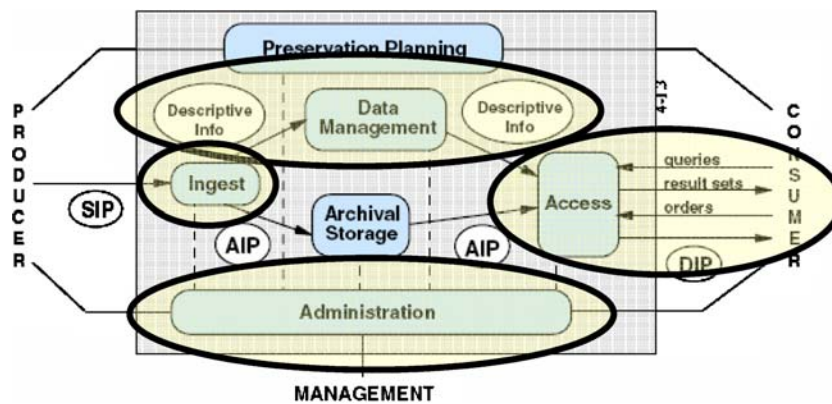


visibility to available resources regardless of where the resources reside. This can help fulfill some of the aspects of preservation planning. One area that a virtual observatory does not fulfill is that of archival storage. From a virtual observatory perspective a permanent archive, represented by the “archival storage” bubble in Fig. 6, is simply another repository with its requirements and policies being set by the archive organization. There can be many different archive organizations, each governed by different policies, that make resources available through a virtual observatory. Virtual observatories also do not specify how resources are moved into and out of an archive which is defined in the Archive Information Package (AIP) of the idealized OAIS reference model.

### Discussion

At present the NASA Heliophysics Virtual Observatories are in their initial development phase so extensive testing of the concepts presented in this paper has not been performed, yet. Even so, the preliminary experience with this architecture has shown it to be robust, efficient and effective. The deployment of a search and registry service across all of NASA’s Heliophysics Virtual Observatories is easier since the virtual observatories have chosen the SPASE data model as the basis for sharing information and interconnecting components. Achieving a federated search capability across many providers is aided by adopting a common data model.

The protocol for search services described in this paper could be extended to other virtual observatory systems which have adopted different data models since the details of the local data description are encapsulated by the “ResourceProfile” tag in the results. Currently there are



**Fig. 6** The reference model for open archive information systems (OAIS). a.k.a. ISO 14721:2003. The portions of the data model performed directly by a virtual observatory are highlighted. A virtual observatory connects consumers to the resources that producers make

available. The non-highlighted areas are typically operated independently from a virtual observatory and are governed according to their own policies and requirements

two other data models in prevalent use. They are the IVOA data model (Hanisch 2007) and the PDS data model (Hughes and Yi 1993). Since most information that describes the resource is contained in the ResourceProfile it would be necessary to support the transformation of descriptions expressed in the IVOA and PDS data models for display to the user. This is a more achievable goal for IVOA since its metadata is expressed in XML, whereas PDS uses a propriety expressive form. Supporting multiple data models will be important as the need to search across all sources of data (heliospheric, planetary and astronomical) emerges. The capability to perform a universal search is possible only after each community establishes standards and services for itself.

## Conclusions

Achieving a coherent view into a system with many independently operating units where the content is constantly evolving presents interesting challenges. One challenge is to conceal from the user the complexity and topology of the underlying system. Virtual observatories have been (and continue to be) established to provide uniform access to available resources which may reside at different nodes in a network of providers. A virtual observatory achieves this by harvesting information about resources from multiple providers and collecting this information into registries. We have examined the various topologies that networks of registries can form and have explored the requirements for an efficient, flexible and agile harvesting approach for virtual observatories. This approach differs from existing harvesting protocols and applies the three-rules of efficient harvesting: copy, uniqueness and visit-once to achieve a self-organizing architecture that requires minimal management. The same techniques for harvesting can be used to perform federated searches if

constraints on the resource attributes and relevance scores are added to the harvesting protocol. We have found that by combining a few simple, but well defined approaches that a full featured, highly adaptable virtual observatory can be created.

The Virtual Magnetospheric Observatory, part of NASA's Heliophysics Virtual Observatories has put the described approach into practice and creates a multi-tiered system which is robust and provides accurate and relevant information to the user. The details of the architecture and features such as scheduled harvesting, on-demand harvesting, and federated searches with relevance scoring have been developed in response to user expectations. We have used these capabilities to build a robust system that requires minimal management and quickly adapts to the sometimes fluid nature of projects and data responsibilities. The scalability of the architecture will be tested over the next few years as virtual observatories arise in the international community and the topology of the registry network expands and becomes more complex. The initial results indicate that we can expect a richer and more efficient environment to conduct scientific research.

**Acknowledgements** This work was supported by the National Aeronautics and Space Administration under Grants No. NNX07AC95G and NNX07AC93G and issued through the Virtual Observatories for Solar and Space Physics Data (S3CVO). The UCLA/IGPP publication number is 6359.

## References

- CCSDS. (2002). Reference model for an open archival information system (OAIS). (CCSDS 650.0-B-1), <http://public.ccsds.org/publications/archive/650x0b1.pdf>
- Fielding RT (2000) Architectural styles and the design of network-based software architectures. Dissertation, University Of California, Irvine, Irvine

- Hanisch R (2007) Resource metadata for the virtual observatory version 1.12, <http://www.ivoa.net/Documents/REC/ResMetadata/RM-20070302.pdf>
- Harvey CC, Thieman JR, King T, Roberts DA (2004) In SPASE—Space Physics Archive Search and Extract (Vol. ESA/ESRIN WPP-232). Paper presented at the PV-2004 Ensuring the Long Term Preservation and Adding Value to Scientific and Technical Data, 05–07 October, Frascati, Italy
- Hughes JS, Yi YP (1993) In the planetary data system data model (pp 183–189). Paper presented at the Twelfth IEEE Symposium on Mass Storage Systems, 26–29 Apr, Monterey, CA. IEEE
- ISO (2003) Space data and information transfer systems—open archival information system. Reference model
- King T, Narock T, Walker R, Merka J, Joy S (2008) A brave new (virtual) world: distributed searches, relevance scoring and facets. *Earth Science Informatics* DOI 10.1007/s12145-008-0002-7
- Lagoze C, Sompel HVd, Nelson M, Warner S (2004) The open archives initiative protocol for metadata harvesting, <http://www.openarchives.org/OAI/openarchivesprotocol.html>
- LOC (2007) CQL: Contextual Query Language (SRU Version 1.2 Specifications): Library of Congress, <http://www.loc.gov/standards/sru/specs/cql.html>
- Narock TW, King TA (2008) Developing a SPASE query language. *East Science Informatics* DOI 10.1007/s12145-008-0003-6
- Ohishi M, Szalay A (2005) IVOA Astronomical Data Query Language, <http://www.ivoa.net/Documents/latest/ADQL.html>
- W3C (1999) XML Path Language (XPath), Version 1.0, from <http://www.w3.org/TR/xpath>
- Z39.50–2003 AN (2003) Information Retrieval (Z39.50): Application Service Definition and Protocol Specification: National Information Standards Organization, <http://www.loc.gov/z3950/agency/Z39-50-2003.pdf>