A Virtual Observatory Vision based on Publishing and Virtual Data

Roy Williams, Reagan Moore, Robert Hanisch

US National Virtual Observatory

We would like to propose a vision of the Virtual Observatory where the "killer-app" is seen to be generalizing and extending the idea of "publication" from the narrow meaning of peer-reviewed journals. Here, publication ranges from private temporary storage, to group access, to public access, through to data that supports peer-reviewed Journal papers in perpetuity. The publication model is further extended by the possibility of Virtual Data -- where only the method of computation is stored, not necessarily the data itself. Furthermore, virtual data products may depend on other virtual data products, creating an implicit network of on-demand computation. This computation may take huge resources, or it may be all within a laptop.

In this vision, the unifying principle is attaching metadata to data, metadata that contains provenance and curation information, or metadata that contains the computing instructions for instantiating virtual data. These metadata objects can be kept in a (public or private) VO registry. Registries can harvest each other with standard digital library protocols, meaning that a private registry can be incrementally made public, and its contents viewable in academic libraries everywhere. There will be a great variety of query and search services on these registries.

The publication is meaningful when it associated with a curation mechanism. There are multiple approaches, all of which have an appropriate community:

- Self-published material. In essence, a researcher creates a personal digital library, provides a web interface that supports browsing and discovery, and registers the collection into a VO registry.
- Project community library. In this case the publisher of record for the material is done by the sponsoring project.
- VO community library. This contains material that is developed by the VO participants, including software, standard digital data reference sets (hyperatlases), publications, educational material. The material is approved by a VO review board.
- Academic library. This is similar to the VO community library, but now the review board for curation is an academic institution, which imposes its own standards for quality.
- Peer-reviewed, archival data. This kind of data is attached through a permanent link to a paper in an academic journal, a paper that explains what the data means well enough for others to use, and which has been adequately refereed.

The sky surveys and telescope image archives comprise examples of project based digital libraries. They provide provenance, descriptive, administrative, and structural metadata about their holdings, support browsing and queries, and provide a curation mechanism to assert quality. These projects **assert** the quality of their data, but there is no VO-operated review process at this or any higher level.

We assume that astronomical data comes first from a telescope or survey, delivered by one of a number of standard VO protocols -- Cone Search, Simple Image/Spectral Access, Open Sky Query, etc. The data will be manipulated by our customers, using VO-compliant services, into new, derived products. Eventually a paper is written about the investigation, and there may be a

reference to a dataset that is in a VO registry. Through harmonization of the VO and the academic Journal schemes, publication of VO data as part of a peer-reviewed journal will be straightforward.

Datasets and Services

The basic objects here are datasets and services. Either of these can be registered to a VO registry by filling in simple web forms. The dataset may be real or virtual. In the former case, the customer must keep the data online, or contract with a storage service to do so. Alternatively, a journal publisher could take charge of the dataset on behalf of an author, and thereby promise to keep the dataset online. A third possibility would be a "bonded warehouse" that promises to keep that dataset and its attendant inquiry services. On the other hand, if the dataset is virtual, the registered metadata contains the instructions for instantiating the data, and no significant storage services are needed; however, the data will only exist if it's input datasets and transformation services are alive.

Virtual Data

The NSF GriPhyN collaboration introduced the concept of Virtual Data that we will exploit: a data product depends computationally on another product, and it can be either computed and stored directly in a batch fashion, or it can be computed only when requested and the results stored. By merging the batch and on-demand paradigms, we create a flexible and efficient scheme, that can produce popular and static data products quickly through cached copies, yet also produces uncommon or highly volatile data products through the same user-interface, but with computing on demand.

A virtual dataset may be many petabytes in size, but the amount of storage needed may be very little. This is because the only thing that is stored is the *program* that computes the dataset, not the dataset itself. Because the input data was fetched from a well-defined service, and only well-defined transformations have been applied to it, it means that the program for re-creating the data can be well defined.

Virtual data products rely on the continued availability of the services that are involved. In the long run, we need to be freed from the constraint of migrating old service implementations onto new architectures. One solution is to think of virtual data as transient, and compute explicitly any dataset that is to be long-lived. We can also compute and permanently store virtual data, so at that point it is not “virtual “data anymore. In the long term the cost of maintaining and porting computational services exceeds the cost of storing static data products.

One of the advantages of virtual data is that errors can be repaired easily, and that re-calibration and new insights can be inserted anywhere in the pipeline. Recomputation is not a complex manual process, but rather just touching the new material -- like a giant makefile -- and everything derived from there is automatically rebuilt.

Care must be taken with the use of the Virtual Data paradigm. Experience to date with the HST archive, where only raw data is stored and all calibrated data are produced on demand, shows the necessity of good choices in architecture and implementation.

Furthermore, the virtual data paradigm makes it easy to add new derived products, thus encouraging publication. When a new raw data set is published, all of the data derived from it, and from federating it with other products is "virtually" available immediately. When that data is actually requested, it is computed and delivered within hours -- but also stored, so that it will be available in seconds for the next requestor. Another mistake would be to not cache results and thus force every request to start from scratch.

New derived data products are often the result of weeks or months of painstakingly detailed and careful work, often with processing steps adapted to individual data frames. We run the risk of running into trouble selling the VO concept to the folks who do this kind of work, and we should not oversimplify.

Abstract Virtual Data

The definition of a virtual dataset can be explicit or abstract. Explicit virtual data is an expression of the dependency graph – this dataset depends on these others through this service. Abstract virtual data is an expression of what is wanted, with the assumption that metadata services are available to convert this to an explicit form.

For example, a mosaic of the sky computed from a sky survey may be abstractly expressed with the required sky coverage; the explicit definition resolves the coverage statement to a list of image identifiers that need to be combined. Those images may themselves be virtual, created perhaps by a flat-fielding operation from raw images.

Extensions for abstract virtual data are being proposed to the GriPhyN virtual data system. This means that the virtual data product can be created from the multiple data sets returned by a query, without having to specify each data set independently.

The GriPhyN model is also extended to support a virtual data object that depends on other virtual data products, which must also be computed. Another extension concerns the workflow of the computation, with conditional tests done on the data flow to decide if processing should be continued on each of the data sets returned by a query to a collection.

Note that these extensions mean that the virtual data product is no longer deterministic. As the collection is updated, the set of files that are returned from a query may change. Indeed, this makes it possible to remove artifacts, remove bad data, and add new data to the derived data products. One can also put a time range on the query and restrict the result set to data deposited before a given date, making it possible to create snapshots of the derived data products that would have been created on a specified date.

Habving said this, it may be that the processing software itself has changed, and data formats have changed, and maintaining the capabilities for full backward compatibility could be very expensive!

Publishing Datasets

What is the nature of the data that can be published and "virtualized"? It simply means that it is possible to make metadata records describing the dataset, which include:

A. Curation and provenance of the dataset (VOResource forms), including a VO-formatted global persistent identifier, and

B. Application specific metadata about the dataset, including audit trails to track changes to the processing steps, authenticity and intellectual property information, and,

C. A Virtual Data description of the dataset, either explicitly or abstract.

D. Administrative metadata and access controls, pointing to the location of the dataset, whether the data is in a container, mirrored, virtual, or and

These metadata attributes would be managed by the VO Registry. It would be the responsibility of each registry to provide all of the standard metadata that is needed to provide a context for the dataset. The VO is specifying protocols and mechanisms to publish the metadata, through OAI,

OGSA-DAI, WSDL/SOAP etc. VO will then build a working system that would be used to specify the “reference requirements” that other implementations would need to provide.

Once a collection (here and throughout) has curation data (A), it may be published to a VO registry, and therefore shared by all other VO registries. If the specific metadata (B) is to be browsed and searched, it must be of a VO-adopted format; an example of this is if the dataset is a VOTable, in which case its metadata is the data dictionary of the table -- attributes, types, units, UCDs, etc. Of course a private registry can extend the VOResource as it wishes by including application-specific metadata with the resource metadata.

The virtual data metadata (C) will refer to the IDs of other datasets and services; instantiation can only occur if these IDs can be resolved with an available VO registry, the relevant input datasets found, and the relevant services can run. The virtual data definition may be simple, for example:

- *select * from SDSS1 where alpha < 3*
- *compute page 839 of atlas TM-5-TAN-20 from image source SDSS2*

In the cases above, the transformation is concise, and the data sources (SDSS1, SDSS2) are assumed to be VO dataset IDs that can be found in a registry.

Publishing can happen in private as well as public. Even without keeping real, instantiated datasets, customers can have collections of datasets, labeled by VOResource forms, and cross-linked by the Virtual Data derivations. Everything can run from a private registry on a single workstation, but then "opened" to a wider circle when necessary.

Eventually, a dataset that has been created from VO services will be published in a journal. Work is currently underway to harmonize the VO and ADEC , and other Journal publishers identifier schemes.

Therefore we can offer advantages to our customers for using the VO-approved data services. Whenever data is derived from these services, it will be easy to attach metadata, to re-compute it, or to publish it in a journal. The VO offers "soup to nuts" for data computing and – eventually long-term -- publication of the results.

Publishing Services

If a dataset is defined by input datasets and the services that were used to transform them, then the services should be published as well. The VO Registry structure is already in place for this. The service, like the dataset, is defined by VOResource documents, including the WSDL definition if it is a SOAP service, or the parameter definitions for a GET service. There will be a translation layer from the virtual data definition (C) to the exact calling sequence of the relevant services.

Publishing and Federating Database Tables

In the catalog domain, VOTable is the preferred VO format. The ADQL language is an XML fusion of SQL and "region" specifiers, allowing sophisticated grid services such as OpenSkyQuery to be layered on databases. The virtual data description of a VOTable may then consist of the location of the data server and the ADQL query that was used.

Federation is enabled through the *cross-matching* of catalogs, where stellar objects are identified as the same because they have the same (within error) position in the sky. Not just stellar objects, though the situation is indeed complicated when taken as a whole. To first order, objects are identified as “the same” if they are found within a certain angular distance from each other. “Same” means that the emission measured in one bandpass originates in the same physical object as the emission in another bandpass, though the emission may come from different parts of

the object and by different production mechanisms. The interesting thing about federation is not just matching up stellar positions between optical and UV, say, but discerning something about the overall physics of objects through the diversity of their properties in different bandpasses.

The VO will build massive (virtual) derived data products that are cross-matches of existing large catalogs, including SDSS, USNO-B, 2MASS, and DPOSS. Notice the utility of the Virtual Data model here: vast derivative data product can be produced and listed in the registries as "available", but nothing need be computed until a real request comes in; the data can then be cached for reuse.

However, large-scale cross-matches will require substantial quality control and human insight in order to produce usable results. As a first step, yes, you run a cross-match with, say, a 2" angular correlation scale. You find through analysis that many of the associations are spurious (e.g, lead to non-physical spectral characteristics). Or you might find that in one part of the sky you have to accept larger scales because of poorer quality data. So I think what will be cached are derived data products that have had a good deal of thought applied to them.

Publishing and Federating Images

In the VO vision, images are derived from SIAP (Simple Image Access Protocol) services. A set of images can thus be defined (in the virtual data sense) by the SIAP request that produced it.

The application-specific metadata of an image is its "chart" -- the position of the image on the sky (WCS information). Services that federate images through mosaicking will need to select based on overlap of charts -- the output image depends on all those input images whose charts overlap the output chart. Other image datasets may depend in a very straightforward way on the ID of the input; for example the compressed file{n}.jpg always depends on the uncompressed file{n}.raw.

More General Datasets and Theory

There is a desire within the US-VO project to integrate datasets from simulations with observational datasets. These come in a wide variety of formats and semantics: a 3D density image, a set of points with position and velocity, a Fourier transform of a time series, etc. Clearly the curation data (A) can be made for such heterogeneous datasets as easily as for WCS-enabled images and VOTables. However the specific description metadata (B) and virtual data (C) will prove more difficult, since there will be a need for metadata schema as well as the metadata itself. There should be a way for a community to approve a discipline-specific schema and have it become part of the general VOResource so that it can be indexed and browsed.

The real complexity of handling theoretical data and models is that we have to project or filter them through what might be a very complex function in order to get from the parameter space of the simulation to the parameter space of actual data. These functions could be VO services, of course, but they will be highly specialized and may never be in the registry. Rather, they would sit behind a SIAP or OpenSkyQuery service provided by the simulation provider.