VHO
The Virtual Heliospheric Observatory

Design Concept
June 2003

Current VHO team members:

Andrew Davis	California Institute of Technology
George Ho	Applied Physics Laboratory, Johns Hopkins University
Fred Ipavich	University of Maryland
Justin Kasper	Massachusetts Institute of Technology
Davin Larson	University of California, Berkeley
Aaron Roberts	NASA Goddard Space Flight Center
Ruth Skoug	Los Alamos National Laboratory
John Steinberg	Los Alamos National Laboratory
Adam Szabo	NASA Goddard Space Flight Center

Executive Summary

Throughout the past decades, heliospheric physics missions have accumulated vast amounts of data from just about all corners of the interplanetary medium. However, the different data sets, collected by different spacecraft, were not designed to be mined as a single system. The resulting various data products are archived and distributed in widely different formats using radically different services (e.g., NSSDC, PDS, individual spacecraft or PI sites) requiring a considerable level of expertise from the end users. With the implementation of the Living With a Star (LWS) program, an even larger influx of data is expected from sources that will include foreign partners (e.g., Ulysses, Cluster and Solar Orbiter) and planetary missions for which interplanetary observations will be of only secondary objective (e.g., MESSENGER, Beppi-Colombo) increasing the complexity and variety of the available data products. At the same time, new multi-source investigations cutting across discipline boundaries require simple, reliable and rapid access to all of the collected data.

Treating different spacecraft, originally designed as stand-alone platforms, as a system opens up new opportunities in heliospheric research. Proper distribution of the latest, highest quality observations allow instrument cross-calibrations between platforms, reducing the uncertainties in the individual measurements, especially for closely spaced spacecraft (e.g., near-Earth spacecraft, Mercury orbiters). But more importantly, it becomes possible to generate new derived data products that are not possible with single instruments and spacecraft, in effect creating new heliospheric observatories.

Making the various data sets easily accessible is not sufficient by itself to facilitate scientific research. Software tools for data ingest, visualization, and analysis are necessary. The heliospheric community does not currently have a unified software tool library (like SolarSoft) but rely on multiply recreated or informally exchanged programs. Therefore, the development of a unified heliospheric software library would be essential to increase the effective use of heliospheric data.

A simple common data interface, data synchronization and software libraries are not new concepts. However, the recent emergence of new information technology industry standards (e.g., XML, SOAP, P2P) makes it timely to newly revisit these questions. Therefore, we propose that a Virtual Heliospheric Observatory (VHO) be developed that has the following key features:

- A fully distributed system with a minimal middleware where individual nodes can range from data centers to individual PI sites of current and future heliospheric missions.
- A common interface to access all participating heliospheric data either through a browser or through an Application Programming Interface (API).
- The ability to query participating data sources.
- The ability to exchange queries with other VxOs.
- The possibility of the generation and maintenance of synchronized mirror sites.
- The use of industry-standard protocols (e.g., XML, SOAP)
- Extendibility to allow future increases of services (e.g., subsetting, formatting, averaging).

In order to achieve this goal, it is recommended that provisions be made for three different aspects of this proposal: (1) enable current data providers to bring their services to a minimum standard consistent with a distributed query system (e.g., versioning, metadata, service schema); (2) provide for the competitive development and maintenance of the VHO middleware; (3) allow for proposals for the development and deployment of data service synchronization, for a common heliospheric software library, and for other future innovative added value data service developments.

1. Rational for a Distributed Data Environment

With the development of inexpensive, large volume computer storage devices and capable workstations the number of Internet sites serving heliospheric data vastly proliferated over the past decade. As the Living With a Star (LWS) program is implemented, heliospheric missions will become more international in scope resulting in further dispersion of data services. Moreover, the data collection rates have also underwent a 2-3 order of magnitude increase along with the development of more complex data sets so that it is not only difficult to locate the data, but once it is found it is harder to isolate and interpret relevant subsections of it. It is unlikely that this trend will reverse in the close future, nor appears to be any compelling reason to concentrate all heliospheric data holdings at one single repository.

At the same time, current event based studies emphasize rapid access to a wide range of data often within hours of the actual observations necessitating direct and public access to the data production sites and to the tools that the instrument PI teams use. On the other hand, heliospheric physics research started to depend more and more on multi-spacecraft and cross-disciplinary observations – especially in the near-Earth environment – requiring close cooperation between the individual instrument teams and the development of new data products and services that treat the individual instruments and spacecraft as a unified "virtual" observatory.

Fortunately, information technology advances makes it possible to deploy relatively simple and inexpensive data services as amply demonstrated, for example, by the success of the ACE Science Center[1]. Rapidly developing industry standards for distributed web services (e.g., XML, SOAP) encouraged the proliferation of widely available open source software requiring low levels of specialized manpower for deployment and maintenance of data services. Thus it appears that both the need and the necessary building blocks are there for the development of a distributed heliospheric data environment, the "Virtual Heliospheric Observatory", that is held together by a middleware that provides a common view and querying capability of the participating data sites, the minimum amount of "glue" to hold the system together. This proposed data environment is very similar to that outlined by the Virtual Solar Observatory (VSO)[2] [*Gurman et al., 2002*], though with a number of key differences that stem from the peculiarities of heliospheric data services.

[1] http://www.srl.caltech.edu/ACE/ASC/
[2] http://virtualsolar.org/

2. The VHO Architecture

We have envisioned the VHO to be a dynamically evolving data environment where individual elements can be proposed, competed and reviewed separately and new functions and capabilities introduced as the need arises. In this paper we describe what we feel is the smallest core building block that would be necessary for the deployment of a fully functional, albeit with reduced capabilities, VHO. We also outline our current understanding of some additional useful features that would significantly increase the usefulness of the VHO, though are not required for immediate deployment. Hence, one of the key features of the VHO has to be extensibility.

We propose that the core VHO has to be able to perform the following functions:

VHO Core Functionalities

- Provide a common and completely open discovery and access method to all participating data sites via a middleware with the data returning directly to the user bypassing the middleware.
- Provide a minimal query capability to search for data by spacecraft, instrument and/or time of measurement.
- Allow the above functions to be carried out via a simple browser interface or a customizable application programming interface that can have multiple instances.
- Establish the minimum metadata and version requirements (via XML Schema) of data services and provide templates for potential data providers.
- Develop means of registering new and validating existing services.
- Develop a core software library (similar to SolarSoft) that is linked to the data service (metadata that can point to relevant software routines).

Many of these core functionalities are identical with those of the VSO. Specifically, the typical user will be able to send a query through a common browser interface to the VHO middleware (see Figure 1.). The middleware will route the query to the appropriate registered data provider(s) with the query translated to fit the specifics of the particular data service(s). Next, the data service returns the query results (not the matching data set) to the middleware where it is combined and organized before returning it to the user. The query results returned to the use will include hyperlinks that can be used to download the required data directly from the data service sites bypassing the middleware.

Notable differences from the prototype VSO architecture are those that call for minimum requirements on the data providers, the establishment of a common data exchange method between data providers and the developments of the foundations of a heliospheric software library. These stem partly from differences between the typical usage of solar and heliospheric data and partly from differences in the current state of data services. Specifically:

Requirements on Data Providers:

Even though by and large all heliospheric data sets are publicly accessible, the method of access is typically through simple web pages or FTP service. We are aware no heliospheric data provider that provides SQL query capabilities, a common feature of all current VSO participants.

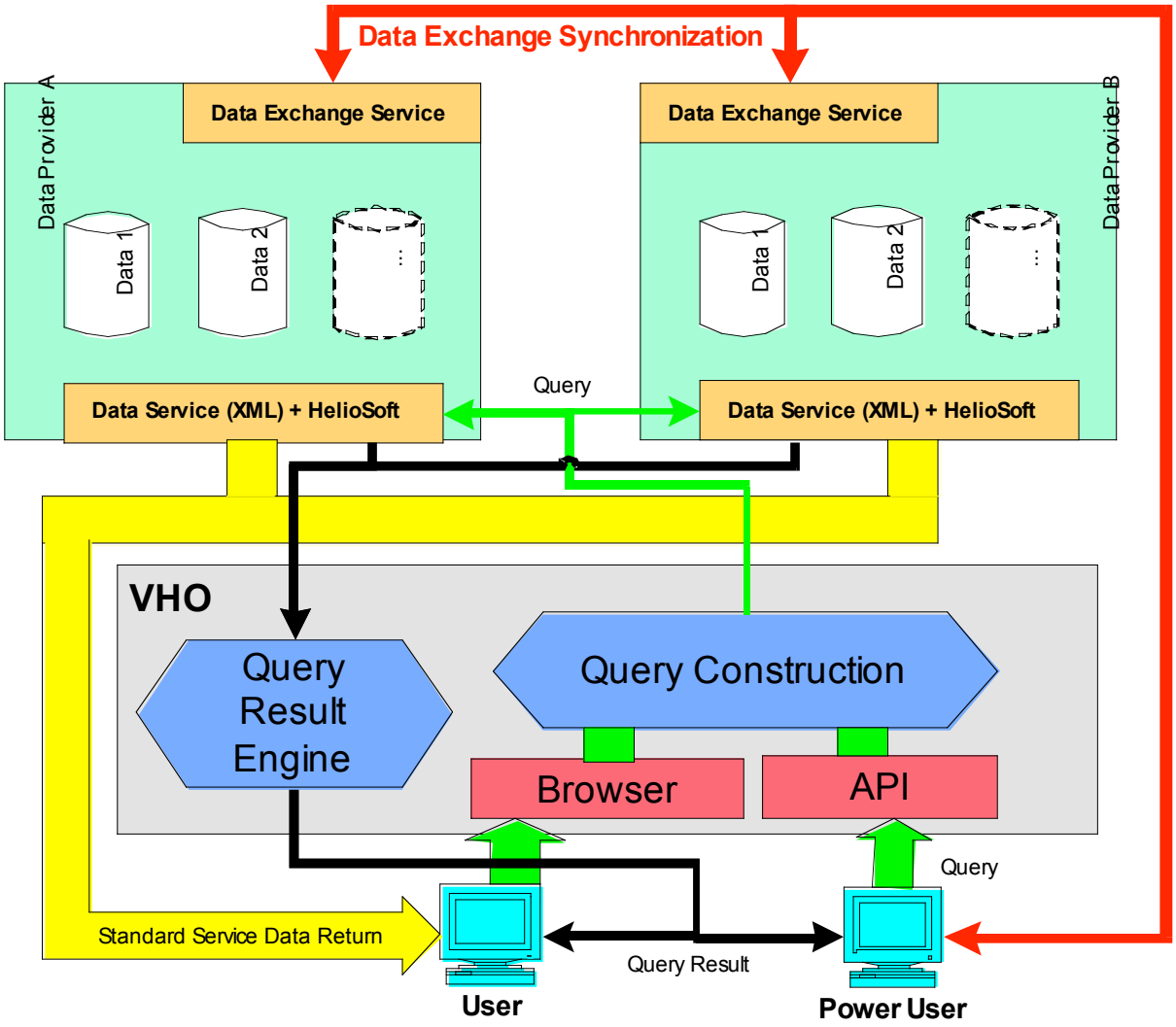

Figure 1. The core VHO architecture. The diagram depicts the VHO middleware in the center that allows users to send queries to participating data providers via either a browser or API interface. The main data transfer takes place without the intervention of the middleware. A solid red line shows the data exchange backbone that allows data inter-calibration and the generation of merged data products. Finally, the first steps are made for the development of a common heliospheric software library, HelioSoft that initially resides with the data providers.

Rather than requiring each data provider to develop SQL database descriptions of their holdings, we recommend that they be given the option to capture the characteristics of their data into a simple XML metadata format that has a few required tags – to allow query processing – but is extensible to capture the unique properties of each data set. This simple metadata standard will be documented in an XML Schema that would minimize the required effort by each data provider to join the observatory.

Data Exchange Between Data Providers:

Increasingly multi-point studies are preferred in heliospheric physics that require precise inter-calibration between different platforms and the generation of merged data products that are based on multi-instrument and/or multi-spacecraft inputs. A number of these merged products are already generated, mainly by near-Earth spacecraft teams who arrived at their ad hoc techniques

historically often by "reinventing the wheel". Therefore, a uniform data exchange and verification method needs to be developed that is open to the heliospheric community to reduce the enormous time and effort currently required to collect and merge different data products. This would be of tremendous benefit not only to the instrument teams – allowing more refined inter-calibration methods – but would likely result in the generation and public distribution of new and imaginative combined data sets resulting in scientific progress. This data exchange mechanism can be encoded in a VHO middleware application programming interface (API), and instances of it run on the data provider sites. This would open up the large volume data exchange backbone between data providers to any interested party, who by deploying locally this VHO API could rapidly become an added value data provider itself encouraging the proliferation of new data services.

Heliospheric Software Library:

Providing open and uniform access to heliospheric data sets by itself is not sufficient to make the VSO widely used. The current heliospheric data sets are stored in a wide variety of formats often including corresponding parameters that are sufficiently different to preclude direct comparison. As a result, data providers are inundated with special data requests that are most often no more than format changes (e.g., CDF or HDF -> ASCII), subsetting or simple recalculations (e.g., thermal speed -> temperature). Most data providers already have considerable local software libraries to accomplish these tasks. However, these useful routines are not publicly available (unlike in the solar community with SolarSoft) and each user is left to develop their own data ingest, visualization and analysis routines. Therefore, we recommend that the basic infrastructure of a heliospheric software library be developed as part of the VHO activities. IN its first manifestation, it could be as simple as existing software routines made public by each data provider with proper metadata that would allow VHO queries to retune not just the requested data, but also point to the appropriate library routines.

3. Possible VHO Augmentations

The core VHO outlined in the previous section is not envisioned as the final product. Rather, it is expected to grow and evolve in time as is demanded by the scientific community. Therefore, we feel that it is very important that a mechanism is established whereby additions to the VHO can be competitively evaluated and funded. It is unlikely that we can envision all possible needs emerging in the future, but some augmentation to the core VHO concept is already apparent:

Connecting the VxOs:

As cross-disciplinary studies become more and more prevalent, especially with the LWS program, it is likely that a typical user would want to have a uniform access to data from more than one discipline. Rather than duplicating the functionalities of the various discipline VxOs, it is desirable to allow them directly communicate to each other passing on queries that are more appropriate to the other and making sense of the returned results thus allowing the user to interact with the whole space physics data set through any of the VxOs. Fortunately, industry standards for connecting SOAP services are starting to emerge (e.g., WSDL) that would make the VxO inter-communication realizable without significant modifications.

Connecting Modeling Centers:

In the heliosphere, where multi-point observations are very sparse, global modeling efforts play an especially important role in developing new scientific understanding. Therefore, it is desirable to connect modeling centers to the VHO so that for the user they appear as one of the data providers. Aside from the same process how any new data provider would join the VHO, modeling centers would need to solve the problem of presenting their often tremendous size result grids in more readily usable forms, such as predictions along a particular spacecraft trajectory and time. This is often a very computationally intensive proposition that can be best eliminated by rerunning the model with along the user requested trajectory. This necessitates that modeling center nodes not only provide data, but also receive input to run on-demand codes (e.g., use observed photospheric magnetic field data as input boundary condition and calculate model predictions along provided spacecraft trajectories). By adding the modeling centers to the data exchange backbone would allow the automatic pickup of data from other data providers, but passing user provided input still needs to be resolved.

Providing Computational Services:

Providing access to the data and tools to ingest and process it is the obvious first step; however, a large fraction of the science users do not want to develop a sufficient depth of understanding of

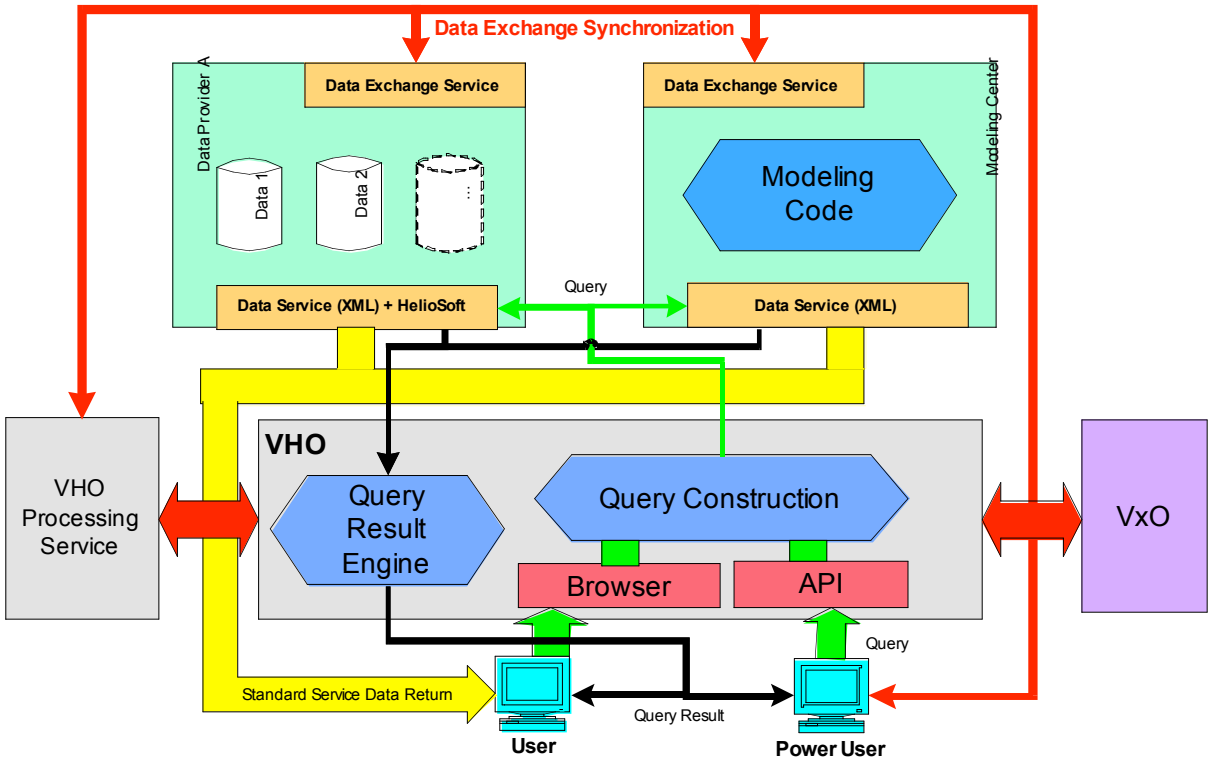

Figure 2. Diagram depicting a possible augmented VHO architecture. New elements are the connection to the other VxOs, the inclusion of modeling centers as possible data provider nodes and added value processing middlewares.

the various data products that would allow them to crop the appropriate subset of data. Therefore, there is a considerable level of interest in the VHO providing not just data discovery a delivery, but also various processing capabilities. In order to maintain the small-box profile and reduce undue network traffic, we believe that it is not desirable to load further processing capabilities into the primary VHO middleware. However, as multiple instances of VHO become a reality and the connection between VxOs established, we see no reason why additional middlewares that specialize in various added value processing cannot be allowed. Like the modeling centers, these processing middlewares would be on the data exchange backbone, allowing them to pick up all necessary primary data, and the primary VHO middleware would divert traffic to them only when their services are required. Thus, the "lego block" architecture of the VxOs is preserved and functionality is added without creating a network bottleneck at the primary VHO middleware.

These augmentation concepts are depicted in Figure 2. where one of the data service sites could be a modeling center. However, since the whole VHO architecture was designed to be fully extensible, other ideas can be incorporated with the same ease.

4. Technical Approach

Motivation:

Current heliospheric mission data sets are publicly available from a diverse group of service providers that are geographically dispersed and are significantly different in capabilities. They range from simple FTP services of flat directory structures to minimal HTTP-based, query enabled sites; from single instrument data service to large, multi-spacecraft data archives. The diversity of heliospheric data services are expected to further increase with time. Therefore, it is desirable to establish a data environment that connects them and allows user access to their holdings via a common interface.

The heliospheric data sets can be unified by one of three ways: 1. All of the data sets can be collected at a single data repository that would require the transfer of huge data volumes and is unlikely to result in rapid public access to mission data. 2. Combine only the complete metadata of the individual data archives at a single site that would reduce somewhat the volume of data to be transferred but would not significantly reduce the workload on the data providers and, hence not result in the reduction in time required for new data access. 3. Finally, a lightweight middleware containing only information describing the data services provided (in opposed to the actual data served) can be deployed that is requires a very low level of data volume transfer, allows for flexibility and since all of the data remains at the providers' site, will likely improve the speed of putting new data and services on line. This final option is our recommended VHO concept. The VHO does place an increased burden of development on the individual data providers – a significant startup workload as none of the current heliospheric services have SQL database capability – however, after initial startup support, this architecture should lead to more innovative and rapidly implemented solutions. It should be also pointed out that the VHO concept leaves the long-term data archival as a data provider responsibility. The VHO will improve data transfer options, but in no ways would replace the archival responsibilities of NSSDC and PDS.

Overview:

The primary purpose of the VHO data environment is to allow users to search and retrieve heliospheric data using a common interface. Users will interact with the VHO using either their web browsers or an application programming interface (API) where they compose their query. The user query is then submitted to the Query Construction Engine that, relying on the content of the VHO Registry, determines which particular data services are relevant for the current search and converts the user query into a format that is understood by the particular data services to be contacted. The contacted data services will search their holdings and return query results to the VHO middleware where the Query Result Engine combines and reformats the individual results to provide a uniform reporting mechanism for the user. At this point, the user has the option to further refine the query based on the results returned or request the identified data files directly from the data providers bypassing the VHO middleware.

<Provide flow diagram of information flow>

Next we describe the elements of the VHO middleware.

User Interface:

The HTML-based web browser interface provides the user a common way to query and access the holdings of all participating data providers. This interface presumes only minimal computer knowledge on the user's part requiring only a commercial web browser and the filling out of query request forms. This interface hides all of the power and complexity of the data environment, but at the same time allows for the simultaneous search of multiple data services.

The API interface, while more complicated to implement, provides the extra flexibility to develop custom interfaces and direct query construction and data download from the user's software (e.g., from IDL or MathLab). The specifications and examples of this interface will be published on the VHO web site.

Query Construction Engine and VHO Registry:

The Query Construction Engine, a program running on the VHO middleware, determines which participating data providers the query should be routed to and converts the query to a format appropriate for the particular data provider. It can accomplish this task based on data provider information stored in the VHO registry residing in the VHO middleware. The Registry does not contain a comprehensive description of all aspects of the data sets served but only that is required to identify the kind of data served (e.g., *in situ* thermal plasma, or magnetic fields), the general location of the observations (e.g., inner or outer heliosphere, or near Earth) and the time span of the data available. In addition, the Registry contains information on the type of service available at the participating sites (e.g., FTP, HTTP or SOAP service) and details on the conversion between the common VHO query format and the individual data service specifications. The Registry stores this information in XML metadata files, one for each individual data set. The format and requirements of the Registry metadata files are described in an XML Schema that will be openly available. This XML Schema will facilitate the simple addition of future data

services to the VHO data environment. Based on the content of the Registry, the Query Construction Engine will parse the user input and distribute appropriately reformatted queries to all relevant participating data service. Details on the requirements and options on the capabilities of the data services will be described in a later section.

Query Result Engine:

The individual query results returned by the contacted data services will be collected and organized by the Query Result Engine, another program residing in the VHO middleware. Just like the Query Construction Engine, user interaction will be through either a simple browser interface or an individually constructed API. In either case, the user will have the option to further refine the search or to proceed to data download directly from the appropriate data provider removing the VHO middleware from the chain. Since at its first incarnation the VHO will not support added value data processing, the user will be limited to data formats that are available from the particular data provider.

References

Gurman, J. B., R. S. Bogart, K. Tian, F. Hill, S. Wampler, P. Martens, A. Davey and G. Dimitoglou, The Virtual Solar Observatory: Design Proposal, *NASA/GSFC Internal Document*, November, 2002.