Earth Sci Inform (2008) 1:43–48
DOI 10.1007/s12145-008-0007-2

RESEARCH ARTICLE

Developing a SPASE Query Language

T.W. Narock · T. King

Received: 31 July 2007 / Accepted: 13 February 2008 / Published online: 6 March 2008

Abstract The advent of the Virtual Observatory has begun an evolution in the space physics data environment. A number of nascent and discipline specific Virtual Observatories have started to emerge with an emphasis on data search and retrieval. As this new data environment takes shape an emphasis will be placed on interdisciplinary communication in attempts to address large scale and global problems. To this end we formulate the development of a query language to facilitate Virtual Observatory to Virtual Observatory communication. Furthermore, we outline the goals of such a language, how it would work and how existing community efforts can be leveraged to speed the development of this query language.

Keywords Virtual observatory · Data search · Data retrieval · Virtual observatory communication

Communicated by P. Fox

T. Narock (✉)
Goddard Earth Science and Technology Center,
University of Maryland Baltimore County,
Baltimore, MD, USA
e-mail: Thomas.W.Narock@nasa.gov

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

T. King
Institute of Geophysics and Planetary Physics,
University of California,
Los Angeles, CA, USA

Introduction

The astronomical community first coined the term Virtual Observatory (VO) [1,2,3] with their work in the creation of a distributed computing system. This system consisted of simultaneous access to disparate and heterogeneous data sources in addition to advanced visualization and analysis tools. In order to facilitate this effort the community developed the Astronomical Data Query Language (ADQL) (Yasuda et al. 2004). This query standard was intended to provide interoperability among the data servers. It is implemented through the exchange of XML documents that contain a subset of SQL commands. ADQL has proven beneficial to the astronomical community and a number of systems have been developed that utilize it (O'Mullane et al. 2005; Shirasaki et al. 2006).

The VO paradigm has begun to influence space physics and a number of nascent VOs are beginning to emerge within this community. In contrast to the astronomical community, which has one VO serving everyone, the NASA space physics community deemed that multiple sub-discipline VOs would be created with the ultimate goal of unification into a grand data access system. In particular, six VOs were commissioned. Figure 1 shows an illustration of the space physics environment from the Sun to the Earth and points out the regions covered by each VO. Additionally, several VOs are being developed independent of NASA, supported by other funding agencies. Although potentially applicable to all space physics VOs, this work

[1] US National Virtual Observatory, http://www.us-vo.org

[2] Astronomy and Astrophysics in the New Millennium (Decadal Survey), National Academy of Science, http://www.nap.edu/books/0309070317/html/

[3] International Virtual Observatory Alliance, http://www.ivoa.net

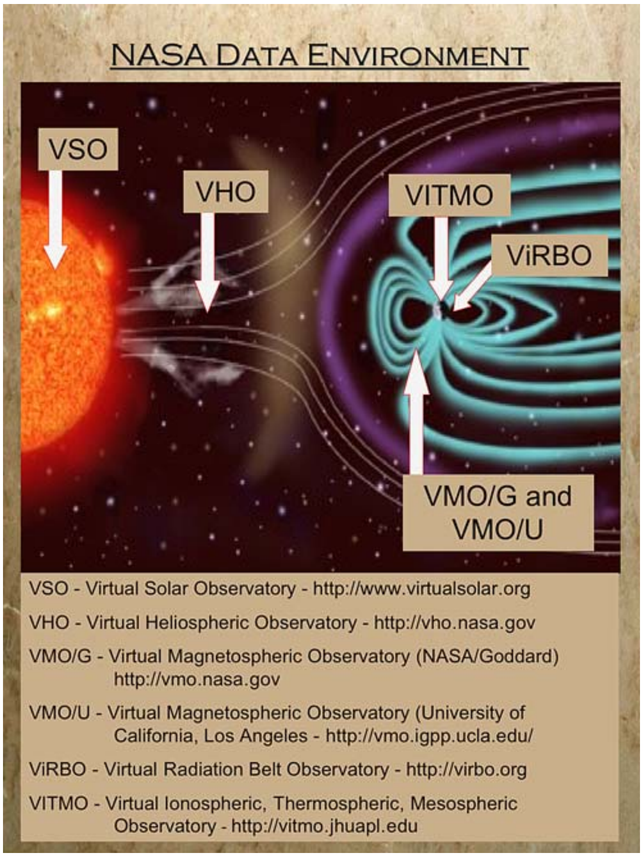

NASA DATA ENVIRONMENT
VSO
VHO
VITMO
ViRBO
VMO/G and VMO/U
VSO - Virtual Solar Observatory - http://www.virtualsolar.org
VHO - Virtual Heliospheric Observatory - http://vho.nasa.gov
VMO/G - Virtual Magnetospheric Observatory (NASA/Goddard) http://vmo.nasa.gov
VMO/U - Virtual Magnetospheric Observatory (University of California, Los Angeles - http://vmo.igpp.ucla.edu/
ViRBO - Virtual Radiation Belt Observatory - http://virbo.org
VITMO - Virtual Ionospheric, Thermospheric, Mesospheric Observatory - http://vitmo.jhuapl.edu

Fig. 1 An illustration of the current NASA VO data environment. The discipline specific VOs are shown in their respective regions from the Sun to the Earth

focuses primarily on the NASA efforts, as they are the primary users of the data model used in this work.

The Space Physics Archive Search and Extract (SPASE) (Harvey et al. 2004, 2007)[4] consortium is an international group of space physics researchers, VO developers and data providers who have taken on the task of creating a comprehensive space physics data model. This data model consists of agreed upon terminology and definitions as well as protocols on how to document a data product for use in the community and use in VOs. It serves as a standard to which the diverse discipline vocabularies can be mapped, thus serving as a lingua franca. The SPASE data model is implementation neutral and the group currently provides an implementation in XML Schema.

Within the NASA VO framework the SPASE data model has been adopted as a metadata standard for exchanging resource descriptions. There exists community consensus amongst the NASA VOs in the use of SPASE and frequent dialogue continues between VO developers and the SPASE consortium. Participation in the consortium is open to all in the community. Updates and additions to the data model are proposed to the consortium and are then debated and potentially ratified. As such, SPASE has become a data model oversight board and its implementation an adhoc standard amongst the NASA VOs. This data model allows VOs to define and classify spacecraft, instruments, data products, and parameters in a clear and consistent manor. Sharing of descriptions is facilitated through unique identifiers assigned to each resource. SPASE affords straightforward comparisons of data products and eases the end user's use and understanding of the data. We see the SPASE data model as the common thread uniting all of the NASA discipline specific VOs and we have devised a method of using SPASE to address the issue of interoperability among the VOs.

Within a given community users may find it sufficient to utilize the programming interface and web services of their VO. However, these often specialized and specific calls make it extremely challenging to communicate with services outside a user’s discipline. Furthermore, this reliance on static interfaces creates a barrier to interdisciplinary communication. Additionally, the lack of standards in interface implementation further complicates matters. Thus, addressing large scale and global space physics problems is at present a formidable challenge.

We conceive a SPASE query language to address this challenge, now, while space physics VOs are in their infancy, so that the goal of VO interoperability can be achieved more readily. This query language addresses the issues of interoperability and shares some features of the ADQL solution. It differs from ADQL in that SPASE descriptions are considerably more structured than their IVOA counterparts and a traditional relational approach is insufficient. Specifically, we propose to adopt key aspects of ADQL and combine them with the existing technologies of Document Object Model (DOM)[5], XQuery[6] and VOTables[7]. The following sections outline our implementation plan and introduce use cases that define our requirements.

Use cases

There are an innumerable number of questions one might like to ask in space physics. Ideally, these questions, and the requirements they impose, could be broken down into several categories. Through use cases we intend to look for commonalities in requirements and define how our query language should be developed. The following are three use

[4] SPASE Group, http://www.spase-group.org

[5] W3C Document Object Model, http://www.w3.org/DOM

[6] W3C Recommendation, http://www.w3.org/TR/xquery

[7] VOTable, http://www.ivoa.net/twiki/bin/view/IVOA/IvoaVOTable

cases that have emerged from discussions with members of the space physics community.

(1). Provide a mechanism for queries difficult to construct with current technologies

Use Case: Answer a query requiring examination of multiple SPASE XML metadata documents without a priori knowledge of which documents.

Actors: Any Virtual Observatory or registry containing SPASE metadata

Requirements:

1.) a means of transporting query over the network
2.) ability to express intersection and union conditions
3.) a means of expressing which SPASE terms to query and with what values

(2.) Provide a mechanism for science queries based on SPASE terms

Use Case: The "Halloween Storm" of 2003 produced a series of solar eruptions that were some of the most powerful ever seen. Find all the solar images available of these eruptions.

Actors: Virtual Solar Observatory (VSO), Catalogs of solar eruptions (times, locations)

Requirements:

1.) a means of transporting query over the network
2.) ability to express intersection and union conditions
3.) a means of expressing which SPASE terms to query and with what values
4.) a means to finding catalogs if they exist and are not part of VSO

(3.) Provide a mechanism for communicating with data environment web services not affiliated with VOs

Use Case: Find a web service and access it using a standardized messaging schema

Actors: Service lookup mechanism, web service

Requirements:

1.) a means of transporting query over the network
2.) a means of expressing intersection and union conditions
3.) a means of expressing SPASE terms and values

Examination of the above use cases immediately leads to common requirements. First, as this is a distributed system, a mechanism or mechanisms must be defined to address query message transportation. For this we choose Simple Object Access Protocol (SOAP)[8] and Representational State Transfer (REST)[9]. REST is becoming increasingly popular for web development. Additionally as these efforts are web and web service based, REST and SOAP implementations would cover the largest base of users. Further inspection of the use cases reveals a need for union and intersection operations as well as a formal means to express SPASE with values. To meet these requirements we plan to adopt the SQL semantics of ADQL and combine them with expressive power of XQuery. We can exploit the XML DOM model of the SPASE data model and express our queries using XQuery notation. Multiple queries can then be combined using the intersection and union operations of ADQL. The XQuery specification does offer union and intersection operations as well as the "collection" function for operating on a set of documents, however, the "collection" function requires a priori knowledge of which documents to search. One of the core functions of the SPASE query language is the ability to search multiple documents without prior knowledge of which documents while simultaneously answering all or parts of the query. It should be noted, that while the queries are expressed partially in XQuery they need not be executed via an XQuery implementation. As far as the query language is concerned, XQuery is required only for query construction.

Because the query language will be carried out via machine-to-machine communication it will also be beneficial to have a lookup service to differentiate the various capabilities of the space physics data environment. In a number of situations, e.g. use case (2), it is not immediately obvious if a particular capability exists and if so where it resides. Multiple web services will exist independent of VOs. Additionally, even though a service is associated with a VO it may be reachable by a different address, i.e. different endpoints for web services. To this end, we plan to extend a current registry, VOregistry[10], to function as a lookup service for the query language. This registry will contain descriptions of the services available and provide a programming interface to allow query language users to find out whom to send their messages to.

The use cases do not dictate a format for the response messages, however, they do indicate that they may often be tabular. That is, one is often interested in a list of files, data products or time ranges that match a particular query. To facilitate this, we employ VOTables as the container of the response data. The response message will be a VOTable that contains fields housing the users requested information. Each field will correspond to an item in the users SPASE query language "SelectionList". (See next section and Figs. 2, 3 and 4 for details).

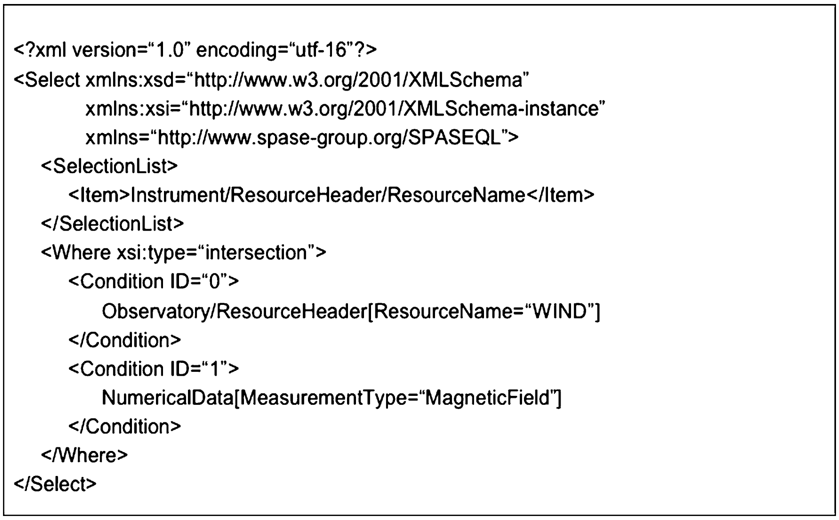
A key aspect of the query language is that it is general enough to express a wealth of queries. This, however, can lead to situations in which a recipient is unable to answer parts or even all of a query. We address this by adding unique identifiers to each of the conditions expressed in a query. This simple, but highly effective, solution allows a

[8] SOAP, http://www.w3.org/TR/soap/

[9] REST Wiki, http://rest.blueoxen.net/cgi-bin/wiki.pl?FrontPage

[10] VOregistry, http://voregistry.gsfc.nasa.gov

Fig. 2 A SPASE query language example in which multiple SPASE metadata documents need to be searched without a priori knowledge of which documents



```
<?xml version=“1.0” encoding=“utf-16”?>
<Select xmlns:xsd=“http://www.w3.org/2001/XMLSchema”
         xmlns:xsi=“http://www.w3.org/2001/XMLSchema-instance”
         xmlns=“http://www.spase-group.org/SPASEQL”>
   <SelectionList>
      <Item>Instrument/ResourceHeader/ResourceName</Item>
   </SelectionList>
   <Where xsi:type=“intersection”>
      <Condition ID=“0”>
          Observatory/ResourceHeader[ResourceName=“WIND”]
      </Condition>
      <Condition ID=“1”>
          NumericalData[MeasurementType=“MagneticField”]
      </Condition>
   </Where>
</Select>
```

recipient to pinpoint the parts of the query it can and can not address. Thus, the query language serves as the standard for asking questions with the various components of the data environment dictating what is available.

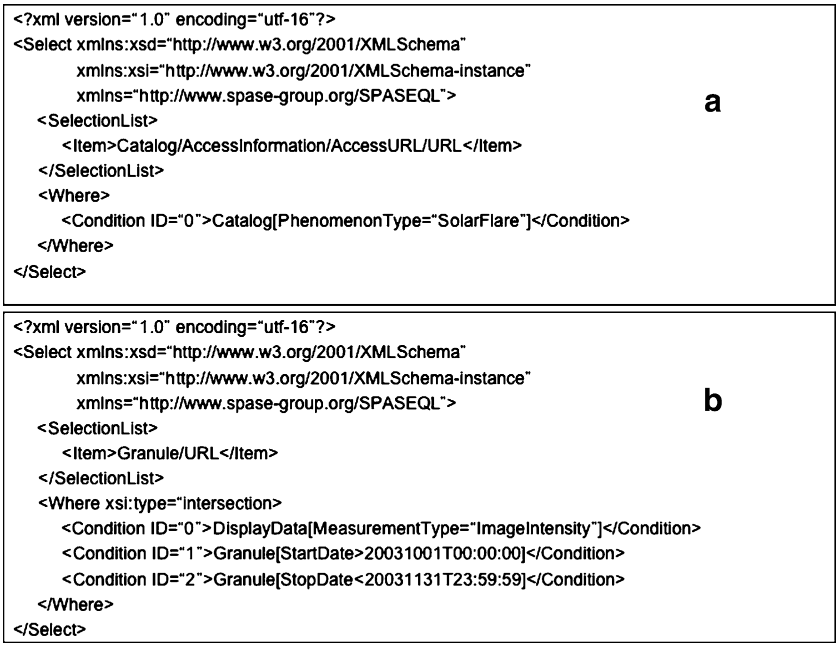
Implementation and examples

As discussed above, SPASE query language messages will be composed through a combination of XQuery statements and ADQL semantics. This combination provides the expressiveness and functionality needed for the space physics community. Figures 2, 3 and 4 illustrate example SPASE query language messages for the three use cases above.

Figure 2 shows an example of the need to query multiple documents without a priori knowledge of which documents. We are interested in the name of the instrument onboard the WIND spacecraft (observatory) (Acuna et al. 1995) that produced data of type magnetic field. A complication to the query arises from the fact that SPASE observatory, instrument and data set information can be, and regularly are, kept in separate files. Thus, this information can be found in three files, however, we do not know which three files. It is queries of this type that show the limitations of a pure XQuery implementation and lead to the need for ADQL intersection and union constructs.

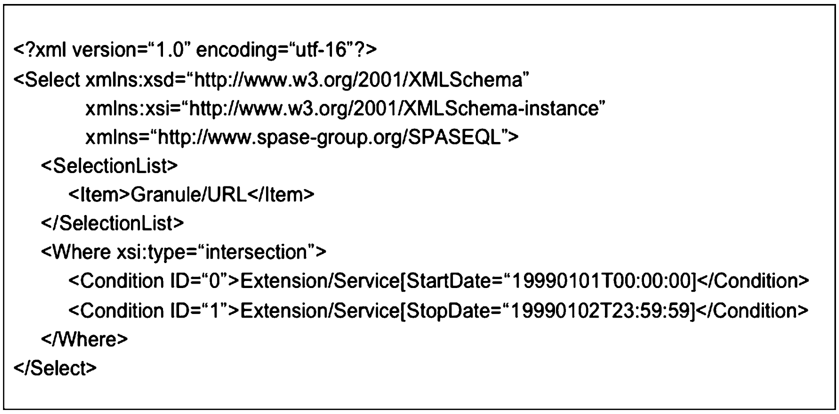
The second example, Fig. 3, illustrates the need for a registry as well as showing how one might query for

Fig. 3 Examples of queries to **a** ask the registry if solar flare catalogs exist and **b** to ask for images during a specified time range



```
<?xml version=“1.0” encoding=“utf-16”?>
<Select xmlns:xsd=“http://www.w3.org/2001/XMLSchema”
        xmlns:xsi=“http://www.w3.org/2001/XMLSchema-instance”
        xmlns=“http://www.spase-group.org/SPASEQL”>
   <SelectionList>
      <Item>Catalog/AccessInformation/AccessURL/URL</Item>
   </SelectionList>
   <Where>
      <Condition ID=“0”>Catalog[PhenomenonType=“SolarFlare”]</Condition>
   </Where>
</Select>
```

```
<?xml version=“1.0” encoding=“utf-16”?>
<Select xmlns:xsd=“http://www.w3.org/2001/XMLSchema”
        xmlns:xsi=“http://www.w3.org/2001/XMLSchema-instance”
        xmlns=“http://www.spase-group.org/SPASEQL”>
   <SelectionList>
      <Item>Granule/URL</Item>
   </SelectionList>
   <Where xsi:type=“intersection>
      <Condition ID=“0”>DisplayData[MeasurementType=“ImageIntensity”]</Condition>
      <Condition ID=“1”>Granule[StartDate>20031001T00:00:00]</Condition>
      <Condition ID=“2”>Granule[StopDate<20031131T23:59:59]</Condition>
   </Where>
</Select>
```



```
<?xml version="1.0" encoding="utf-16"?>
<Select xmlns:xsd="http://www.w3.org/2001/XMLSchema"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xmlns="http://www.spase-group.org/SPASEQL">
    <SelectionList>
        <Item>Granule/URL</Item>
    </SelectionList>
    <Where xsi:type="intersection">
        <Condition ID="0">Extension/Service[StartDate="19990101T00:00:00]</Condition>
        <Condition ID="1">Extension/Service[StopDate="19990102T23:59:59]</Condition>
    </Where>
</Select>
```

Fig. 4 An example of using the SPASE Extension element to interact with a web service

scientific data sets using the query language. It is our presumption that this type of query will be the most prevalent. This type of query allows users to address their primary goal—interoperability to obtain data and answer scientific questions. In particular, this example asks for solar image data during the "Halloween Storm" of 2003 (Veselovsky et al. 2004). Moreover, we are interested in images during solar eruptions. The time intervals of these eruptions are often kept in catalogs for easy lookup. However, to the uninitiated user such catalogs may be difficult to find. They may reside with the VO, in this case the Virtual Solar Observatory, or they may be maintained by a data provider. This uncertainty can lead to such catalogs not being found by the end user. A registry of the data environment capabilities and services will assist in finding and accessing such capabilities.

Figure 4 illustrates another common need of the space physics community. We would like to interact with the multitude of web services that may exist using common language and a common query interface. To accomplish this we leverage the SPASE Extension element that allows for user defined additions to the data model. We propose to maintain a collection of DOM descriptions for Extension content to support querying over constructs not present in the main data model. As the broad community adopts SPASE they will no doubt utilize the Extension element to meet their individual needs. To facilitate the querying over user defined extensions we will collect and make publicly available DOM descriptions of popular extensions.

Once complete the SPASE query language will be a full and rich query language. To facilitate rapid development and prototyping we have defined several levels of development. At each level the complexity of the queries increases. This approach allows a core set of query language functionality to be released in a short time. As this set of capabilities is dissected by the community, prototyping can continue on remaining levels. Table 1 illustrates the SPASE query language levels of development.

Development and validation

The initial development and prototyping of the query language will take place amongst a collaboration of the Virtual Heliospheric Observatory and the two Virtual Magnetospheric Observatories. These three VOs offer a number of unique data sets and spatial regions to amply test the query language. Following satisfactory initial prototyping the query language will be offered up to the remaining discipline specific VOs. This includes the solar, ionospheric/thermospheric/mesospheric and radiation belt communities who can offer a complete test suite. Additionally, we maintain a close working relationship and participation in the SPASE consortium. As such, the consortium is aware of, and in favor of, such a query language and updates to the SPASE data model can easily be followed and folded into the query language. Our efforts are built on participation and collaboration amongst VOs, SPASE and the user

Table 1 SPASE query language levels of development

Level	Requirements
0	Development of software to construct and send queries
1	Processing of queries with one condition
2	Processing of queries with multiple conditions
3	Ability to query VOregistry
4	Demonstration of VO–VO communication
5	Demonstration of scientific workflow involving registry lookup and multiple VOs and services

community. Once realized, the SPASE query language will unite the discipline specific VOs and services and allow them to thrive as a collective whole.

Acknowledgements This work was supported under NASA Research Opportunities in Space and Earth Sciences grant NNX07AC95G. The authors would like to thank the SPASE consortium and Virtual Observatory members for helpful discussions during the development of this work.

References

Acuna M, Ogilvie KW, Baker DN, Curtis SA, Fairfield DH, Mish WH (1995) The global geospace program and its investigations. Spa Sci Rev 71:5

Harvey CC, Thieman JR, King T, Roberts DA (2004) SPASE—Space Physics Archive Search and Extract, In Proceedings PV-2004 Ensuring the Long Term Preservation and Adding Value to Scientific and Technical Data, Frascati, Italy, ESA/ESRIN WPP-232.

Harvey CC, Gangloff M, King T, Perry CH, Roberts DA, Thieman JR (2007) Recent developments towards a Solar System Virtual Observatory. Ann Geophys 25:1–6, In Press

O'Mullane W, Budavari T, Li N, Malik T, Nieto-Santisteban MA, Szalay AS, Thakar AR (2005) OpenSkyQuery and OpenSkyNode—the VO Framework to Federate Astronomy Archives, Astronomical Data Analysis Software and Systems XIV. In: Shopbell PL, Britton MC, Ebert R (eds) ASP Conference Series, vol. 347

Shirasaki Y, Tanaka M, Kawanomoto S, Honda S, Ohishi M, Mizumoto Y, Yasuda N, Masunaga Y, Ishihara Y, Tsutsumi J, Nakamoto H, Kobayashi Y, Sakamoto M (2006) Japanese Virtual Observatory (JVO) as an advanced astronomical research environment. Advanced Software and Control for Astronomy. In: Lewis H, Bridger A (eds) Proceedings of the SPIE, vol. 6274, pp 62741

Veselovsky IS et al. (2004) Solar and heliospheric phenomena in October–November 2003: causes and effects. Cosm Res 425:435–488

Yasuda N, Mizumoto Y, Ohishi M (2004) Astronomical Data Query Language: Simple Query Protocol for the Virtual Observatory. Astronomical Data Analysis Software and Systems XIII. In: Ochsenbein F, Allen M, Egret D (eds) ASP Conference Series, vol. 314