

Ontology-supported Scientific Data Frameworks: The Virtual Solar-Terrestrial Observatory Experience

Peter Fox¹, Deborah McGuinness^{2,3,4}, Luca Cinquini⁵, Patrick West¹,

Jose Garcia¹, James L. Benedict⁴ and Don Middleton⁵

¹ High Altitude Observatory, Earth Sun Systems Lab, National Center for Atmospheric Research, PO Box 3000, Boulder, CO 80307, {pfox,pwest,jgarcia}@ucar.edu

² Tetherless World Constellation, Rensselaer Polytechnic Institute, 110 8th Street, Troy, NY 12180, {dlm@cs.rpi.edu}

³ Knowledge Systems, Artificial Intelligence Laboratory, Stanford University, 345 Serra Mall, Stanford, CA 94305, {dlm@cs.stanford.edu}

⁴ McGuinness Associates, 20 Peter Coutts Circle, Stanford, CA 94305, {dlm,jbenedict}@mcguinnessassociates.com

⁵ Scientific Computing Division, Computing and Information Systems Lab, National Center for Atmospheric Research, PO Box 3000, Boulder, CO 80307, {luca,don}@ucar.edu

Abstract

We have developed a semantic data framework that supports interdisciplinary virtual observatory projects across the fields of solar physics, space physics and solar-terrestrial physics. This work required a formal, machine understandable representation for concepts, relations and attributes of physical quantities in the domains of interest as well as their underlying data representations. To fulfill this need we developed a set of solar-terrestrial ontologies as formal encodings of the knowledge in the Ontology Web Language – Description Logic (OWL-DL) format.

We present our knowledge representation and reasoning needs motivated by the context of Virtual Observatories, from fields spanning upper atmospheric terrestrial physics to solar physics, whose intent is to provide access to observational datasets. The resulting data framework is built upon semantic web methodologies and technologies and provides virtual access to distributed and heterogeneous sets of data as if all resources appear to be organized, stored and retrieved from a local environment. Our conclusion is that the combination of use case-driven small and modular ontology development, coupled with free and open-source software tools and languages provides sufficient expressiveness and capabilities for an initial production implementation and sets the stage for a more complete semantic-enablement of future frameworks.

Keywords: Ontologies, Semantic Web, Knowledge Representation, Reasoning, Data Frameworks, Virtual Observatories.

1. Introduction

Scientific data is being generated, collected and archived in digital form in high volumes by many research groups, organizations and agencies worldwide. Increasingly efforts such as GEOSS – the global Earth observing system of systems (GEOSS 2005) drive requirements for the search, access and use of often-diverse data holdings. In addition, the need for access to and interoperability between these repositories is growing by an audience with varying education levels, research and/or education interests, and technical skills and capabilities. Increasingly, access to data within a single discipline is being complemented with the need to utilize data from multiple disciplines. Progress in these areas is evident (e.g. the Earth System Grid; Bernholdt et al. 2005) and thus, the promise of the true virtual interconnected heterogeneous distributed international data repository is starting to be realized.

However, many challenges remain including interoperability and integration between data collections. We are exploring ways of technologically enabling scientific virtual observatories - distributed resources that may contain vast amounts of scientific observational data, theoretical models, and analysis programs and results from a broad range of disciplines. The virtual observatory (VO) is a particular paradigm that characterizes the aforementioned characteristics of modern scientific data infrastructure. Our goal is to make these repositories appear as if they are one integrated local resource, while realizing that the information may originate from many entities, using a multitude of instruments (or models) with varying instrument settings in multiple experiments with different goals, and captured in a wide range of formats. Our setting is placed within the realm of interdisciplinary virtual observatories, which introduces further challenges. A typical user is unlikely to be a subject matter expert in the entire collection. Indeed, vocabulary differences across disciplines; varying terminologies, some with standardized conventions and some without, similar terms with different meanings, and multiple terms for the same phenomenon or process are among some of the challenges.

Our approach to developing virtual observatories is to utilize a data framework approach (McGuinness, et. al, 2007c). We have used artificial intelligence technologies, in particular semantic technologies, to create declarative, machine operational encodings of the semantics of the data to facilitate interoperability, smart location and access to data, and semantic integration of data. These capabilities were initially made available in a web portal and we then design semantically enabled web services to find, manipulate, and present scientific data, which is accessible over distributed networks.

Our science domains are solar physics, space physics, and solar-terrestrial physics. We have many data collections, spanning disciplines, and growing in volume and complexity. Major communities include those interested in solar images from the Mauna Loa Solar Observatory (MLSO¹), and the NSF-funded Coupled Energetics and Dynamics of Atmospheric Regions (CEDAR²). These collections provided a good focus for virtual observatory work since the datasets are of

¹ <http://mlso.hao.ucar.edu>

² <http://cedarweb.hao.ucar.edu>

significant scientific value to a set of researchers and capture many, if not all, of the challenges inherent in complex, diverse scientific data. The result is the Virtual Solar-Terrestrial Observatory (VSTO) which we view as representative of multi-disciplinary virtual observatories in general and thus claim that our results can be applied in other multi-disciplinary VO efforts (Fox et al. 2006, McGuinness et al. 2006, McGuinness et al. 2007, Fox et al. 2007). We will note such generalizations as well as applications of our present work into other discipline and application areas as appropriate.

In section 2 we present our semantic web methodology. In section 3 we present the motivating use cases, which lead to the specific knowledge representation technical requirements and how they were balanced with our ability to implement them and satisfy user requirements. In section 4, we present and discuss the encodings of the classes, relations and properties in ontologies. This includes the tools and infrastructure we utilized as well as some unifying concepts that were enabled as a result of our knowledge representation. In section 5 we discuss how the knowledge representation and reasoning is used in the VO setting. Finally, we present our conclusions and future knowledge representation needs.

2. Semantic Web Methodology

A distributed multi-disciplinary internet-enabled virtual observatory requires a higher level of semantic interoperability than was previously required by many distributed data systems or discipline-specific virtual observatories. However, programs such as the NASA VxO program; VOHD 2006, has the goal of coupling discipline-specific VxOs into an interdisciplinary system. To enable explicit semantic interoperability for our projects, extensive engagement of the end user; the domain scientist, associate scientists, students and professional assistants was a key element in our methodology.

In order to provide a scientific infrastructure that is usable and extensible, VSTO required contributions concerning semantic integration, and knowledge representation while requiring depth in each of the science areas. To develop VSTO, we needed to systematize the methodology around the end-user in pursuit of the virtual observatory goals. These users had the constraints of a typical research project; modest team size and budget and several major objectives.

At the heart of the method is the use case (Cockburn, 2000), or user scenario. In this sense, we use the term to indicate a specific capability that drives both what knowledge is to be represented and used and also what software and interfaces are built for the user and to the underlying data. Thus, often a use case appears to be a brief statement but in practice is accompanied by detailed descriptions including functional and non-functional requirements, success and failure scenarios, etc.

In developing and analyzing the use cases, our methodology involves a small team made up of domain literate experts, data and instrument providers, knowledge representation and engineering experts, computer science/software engineers and a facilitator. The identification of domain experts was key: two to four carefully chosen experts were sufficient (and preferable) in the use case development and knowledge engineering. At later stages, for example in vetting the ontologies and evaluating extensibility or in augmenting the use cases, a larger group

can be effective, especially in getting community support and buy-in for the utilization of semantic technologies.

The application to virtual observatories requires a web implementation, i.e. some form of web-browser access via a portal, web services as well as native application programming interface (API) access. As a result of this requirement we work within a web-architecture using Java as much as possible but also utilize interface-level access to existing services that provide access to data, graphical representations, extant catalogs, etc.

Within the VSTO project, we elected to use free, or open-source software tools, packages and development environments that we will describe later in the paper with the intent of documenting an end-to-end methodology that would be reproducible and usable by others without the need for significant investments in time and resources.

Other applications of formal semantics in technical architectures, similar to those we have implemented for virtual observatories include work on workflow systems (Gil et al. 2006, Ludaescher et al. 2005), in grid computational settings (DeRoure et al. 2005) and in frameworks for Earth and space science data mining (Rushing et al. 2005). Basic knowledge representation and reasoning can support both computer-to-computer and computer-to-researcher interfaces that find, access and use data in a more effective, robust and reliable way.

3. Use Cases and Knowledge Capture

The use cases described below were developed by domain scientists with assistance from the project team to reflect the particular science areas of interest to the virtual observatory, define the actors and functional elements of the VO and also to scope and evolve the knowledge representation requirements for the ontologies. We started with the first two use cases for the initial ontology and framework development. We then added use case 3 and evolved both the ontology and the framework. Finally, use cases 6, 4 and 5 were added. While these use cases provide specific examples, we chose them to be specific examples that domain experts considered representative of the typical range of tasks domain scientists needed to perform.:

Use case 1: Plot the Neutral Temperature (Parameter) taken by the Millstone Hill Fabry-Perot interferometer (Instrument) looking in the vertical direction from January 2000 as a time series.

Use case 2: Find and retrieve quick look and science data for images of the solar corona during a recent observation period.

Use case 3: Find data, representing the state of the neutral atmosphere anywhere above 100 km and toward the Arctic circle (above 45N) at times of high geomagnetic activity.

Use case 4: Create a movie of the white light solar corona during the whole-Sun campaign month in 2005.

Use case 5: Find and plot/animate data that represents the terrestrial ionospheric effects of a geo-effective solar storm.

For each of these use cases, we generalize them using a template form. For example, the template for use case 1 would be:

Template 1: Plot the values of parameter X as taken by instrument Y subject to constraint Z during the period W in style S. All of the current templates for our use cases are included in McGuinness et al. (2007d).

The sixth use case contains technical and functional constraints.

Use case 6: Provide query services for the Virtual Ionosphere-Thermosphere-Mesosphere Observatory¹ that retrieve the availability of instruments, date-time ranges, and selectable parameters, searched for in any order and with constraints on other selections included in any combination and any order. In addition, provide services to return links to the underlying data once selections are made.

The general form of the use cases is “retrieve data (from appropriate collections) subject to (stated and implicit) constraints and create a representation of the data in a manner appropriate for the data and for the end-user.”

We examined the use case sentences to identify the initial concepts and relations between them. Use cases 1, 3 and 5 originate from the CEDAR program, which embodies a controlled vocabulary including terms related to observatories, instruments, operating modes, parameters, observations, etc. Another motivating scientific community responsible for use cases 2 and 4 – solar atmospheric physics observations from the Mauna Loa Solar Observatory – also embodies a controlled vocabulary with significant overlap.

A number of natural hierarchies were apparent (such as an instrument hierarchy), and important properties (such as instrument settings), as well as restrictions on the values for certain concepts within a given context. We also looked for and found useful simplifications in areas, such as temporal domain.

Our first and third use cases involve a heterogeneous collection of community data from a nationally funded global change research program - CEDAR. The data collection comprises over 310 different instruments, and the data holdings, which are often specific to each instrument, contains over 820 measured quantities (or parameters) including the representation of physical quantities, derived quantities, indices, and ancillary information. CEDAR is further complicated by the lack of specification of independent variables in datasets. Also, the original logical data record encoding for many instruments contains interleaved records representing data from the instrument operating in different modes. Thus odd and even records typically contain different parameters. Sometimes these records are returned without column headings so the user needs to be knowledgeable in the science domain and in the retrieval system just to make sense of the data.

In solar physics images, the original data presentation was that of complex data products, e.g. Mark IV White Light Polarization Brightness Vignetted Data (Rectangular Coordinates). This is a compound description containing Instrument name (Mark IV), parameter (Brightness), operating mode (White Light Polarization), and processing operations (Vignetted Data indicates it has not been corrected for that effect, and a coordinate transformation to rectangular coordinates). Further, the data content retrieved cannot be distinguished from another file unless the filename encoding is understood.

As we progressed through our analysis of the use cases, we followed the same methodology by building upon the initial hierarchies where appropriate and adding or modifying our concepts. The expanded use cases and their variants (e.g. slightly different parameter choice,

instrument choice, etc.) did not lead us to expand the science coverage much; they resulted in the need to integrate across domain areas. However, we did need to re-examine the simplifications we had initially put in place in the class and property structure of the ontology.

A key attribute of the use cases was the requirement of accessing and using the data within the user and application context. This meant that we needed concepts and relations (and properties) to describe the data collections, how data is requested, how constraints are specified and the various options for data products returned to a user or application. In translating the use cases, we also identified existing underlying services (e.g. plotting, and data access which we will discuss later) that we wanted to include in the data framework but mediated by the semantic representations we were developing for the science and instrument concepts. As a result of identifying the upper-level concepts and relations between them, the complex heterogeneity that previously was exposed to the user is now handled by the use of the ontology in the data framework, i.e. a user can deal with familiar terms like instruments and what they measure and not specific details peculiar to each specific instance of them.

Our use cases are documented in a standard² format along with the implemented solutions, process flows, technology choices, benefits of semantic representations, and, where appropriate, diagrams. We plan to continue to evolve the data framework and underlying ontologies using user (science and technical) provided use cases.

4. Ontology Classes, Relations and Properties

4.1 Ontologies

Our knowledge representation of the terms and their inter-relationships mentioned in use cases needed to be included in an operational data framework. Thus, we made the effort to create ontologies in OWL (McGuinness, D., and van Harmelen, 2004) defining concepts, relations, terms etc. noted previously so we could utilize their precise formal definitions for semantic search and interoperability. We limited ourselves to OWL-DL (as opposed to OWL-Full) so that we could leverage efficient reasoning tools for OWL-DL.

Before we began assembling our own ontology, we looked for open source ontologies that made sense to reuse. The primary concerns for reuse were subject area and community usage. We identified a number of controlled vocabularies that we needed to use in our data services. For example, the CEDAR project had developed an extensive set of instrument categories and parameters recorded in data files over 25 years – these were all in a flat listing and contained no semantic information to differentiate one from another. In the worst case there were 8 different representations of time, all with different parameter names and whose meaning could not even be inferred from the name alone but required the user to look up the definition in a detached table. In the case of the MLSO data archive, the controlled vocabulary consisted of compound terms most often referring to data products in which the instrument name, the parameter measured, its processing

¹ <http://vitmo.jhuapl.edu/>

² http://vsto.hao.ucar.edu/use_cases.php

² <http://sweet.jpl.nasa.gov/>

level and type and coordinate representation were all included in the product name. While this gave the user more detailed information on what could be selected, it did not provide the opportunity to select anything other than the specified products. We were able to use all of the basic controlled vocabulary components, i.e. the ‘atomic’ elements, from the CEDAR and MLSO repositories as a starting point.

We also identified the Semantic Web Earth and Environmental Terminology (SWEET¹) Ontology as a broad mid-level ontology covering content areas of interest. We also found that it was gaining acceptance in the science areas we covered. SWEET covered much more in breadth and noticeably less in depth in a few required areas. Instead of importing the entire ontology (thus importing a number of science terms not required), we selected the terms from portions of the ontology that we needed and reused the controlled vocabulary and the definitions of use. We reused the functional decomposition of SWEET as well, reusing for example, the notion of earth realm and sun realm. We leveraged the most from the data and sun realm modules. We then expanded the ontology significantly in the areas required for our efforts, particularly with respect to instruments (along with their operating modes and parameters), observatories, and data products.

Some initial design considerations included the ontology structure and granularity. We followed an iterative design methodology, using our lead domain scientist and lead knowledge representation expert to design and vet the design through use case analysis and other domain experts as well as our entire team. We began with a minimalist class, property, and value restriction structure, initially only adding terms needed to support the reasoning required in the generalized, templated form of the use cases.

This design style was chosen resulting from the following considerations:

(1) A relatively simple representation was more accessible to science domain experts and thus it was easier to get more scientists to review our ontology.

(2) More complex representations take longer to fully comprehend and more importantly take longer to generate community consensus around.

(3) Practical code generation considerations from our supporting environment in terms of java code generation for testing complicated structures took time, thus rapid prototyping and rapid changing of class structure was not convenient.

We used Protégé’s automatic generation capabilities for Java and factory classes (see Fig.1 and Fox et al. (2006a) for details). Our prototype implementation incorporated the Pellet reasoning engine to support the multiple workflow scenarios. The implementation included dependencies on the Java classes and their interconnected structure and if we were changing a large number of properties and their inter-relationships, our prototype implementation would need to be rewritten manually to update the dependencies. We wanted to maintain the ability to provide a prototype implementation for evaluation thus minimizing the complexity of interrelationships that generated dependencies was preferable.

Our current design preserves the simpler initial design and implementation, automatically generates the new classes, and adds incrementally to the existing code. A

rapid development paradigm is preserved and ontology updates can be done without changing the existing data framework.

We focused on six root classes: Instrument, Observatory, Operating Mode, Parameter, Coordinate (including Date/Time and Spatial Extent) and Data Archive. While this set of classes does not cover all observational data, it was interesting to note that as we added data sources to the VSTO use cases, we have found these classes to capture the key and defining characteristics of a significant number of observational data holdings in solar and solar-terrestrial physics. As a result, the knowledge represented in these classes is applicable across a range of disciplines. While we do not claim that we have designed a universal broad coverage representation for all observational data sources, we believe that this is a major step in that direction and has strong similarities to work in the geo-spatial application domain (Cox 2006, Wolff et al. 2006) as well as the recent efforts to develop a schematic follow-on to the Geography Markup Language (GML) known as GeoSciML (GeoSciML).

In Figures 1, 2 and 3 we highlight excerpts from portions of the ontology that was developed using use cases 1 and 2.

Several of the classes from the VSTO ontology are shown in schematic form in Figure 1. Classes are indicated by name in the solid rectangles. Asserted relations/properties between classes are indicated by name with a following ‘+’ sign by arrows with solid heads. Inferred relations are indicated with the same arrows and by name alone. Subclass relations are indicated by ‘is a’ relations by arrows with open heads. Finally, instances of a class are named in dashed line rectangles. In Fig. 1, the classes shown are:

- Instrument: An object that measures phenomenon or parameter.
- OpticalInstrument: An instrument that utilizes optical elements, i.e. passing photons (light) through the system elements, to measure phenomenon or parameter.
- Photometer: An optical instrument; a transducer capable of accepting an optical signal and producing an electrical signal containing the same information as in the optical signal.
- SingleChannelPhotometer: A Photometer which samples with one specified restricted wavelength/frequency range.
- Spectrometer: An optical instrument used to measure properties of light over a specific portion of the electromagnetic spectrum, used for producing spectral lines and measuring their wavelengths and intensities.
- Spectrophotometer: A subclass of both spectrometer and photometer since it can provide functions of both classes.
- Data Archive: A collection of information, a file, set of files, or database made available in machine readable form with associated metadata concerning the data’s origin, purpose and use.
- Data product: A formalized and reproducible representation of data elements for consumption by a user or machine process.
- Observatory: A physical location in which observations are made.

We see in Fig. 1 that an observatory operates each instrument. It is the observatory that has properties such as location (latitude, longitude, elevation), name, operating organization, etc. and the location of a particular instrument is deduced from its observatory. Each instrument has a property for the associated data archive

¹ <http://protege.stanford.edu/>

representing measured parameters, i.e. the representation of physical parameters of interest to the information and/or data collected by the instrument. Similarly, an instrument operating in a particular mode (see Fig. 2) has a measured parameter whose value is members of the class Parameter.

A complete discussion and presentation of the instrument ontology, including all classes, properties, relations and value restrictions is beyond the scope of this paper. A class excerpt from instrument ontology using some abbreviated names for clarity follows (indentation denotes levels of a sub-class hierarchy):

- Radar
 - Incoherent Scatter
 - Coherent Scatter
 - Ionospheric Doppler (sameas High Frequency)
 - MST (Mesosphere Stratosphere Troposphere)
 - Medium Frequency
 - Low Frequency
 - Meteor Wind
- Sounder
- Optical Instrument
 - Heliograph
 - Interferometer
 - Fabry-Perot
 - Michelson
 - InfraRed
 - Doppler
 - Imager
 - AirGlow
 - All-Sky Cameras
 - Lidar
 - Polarimeter
 - Photometer
 - Single-Channel
 - Multi-Channel
 - Spectrophotometer
 - Spectrometer
 - InfraRed
 - Mass Spectrometer
 - Spectrophotometer

Due to the structure in the class hierarchy we are able to take advantage of inheritance and other inferences in the semantic data framework. A similar list could be presented for the parameter ontology. We do note that to account for the differing types of parameters we include subclasses of the Parameter class such as TimeDependentParameter, SpatialDependentParameter, ErrorParameter, and three classes to denote groups of parameters related to the charge state of the terrestrial atmosphere; electron, ion and neutral. A complete list is available from the ontology file whose address was indicated earlier.

In developing the class properties and relations we elected to add minimal properties and assert only the most direct relations (e.g. hasOperatingMode in the case of Instrument) and then utilize reasoning engines to infer implicit information. We also wanted to be able to evolve the ontology, including the hierarchies if needed as we added use cases and terms and needed a lightweight way to achieve that. We also elected not to add large numbers of properties on classes even though many were suggested unless we were going to use them, again in satisfying use case requirements or in inferring information (both broadening and narrowing). An example of this is for the Observatory class where we added name and location

properties but not physical postal address, operating organization, opening hours, and so on.

Fig. 2 shows another portion of the VSTO ontology, and similar annotations are used as in Fig. 1. In the present figure, the dashed arrow indicates an inference that Neutral Temperature is a time dependent parameter and as such a time quantity needs to be retrieved from the Data Archive to accompany the quantity so it can be plotted.

One challenge faced when integrating scientific data taken from multiple instruments is in understanding the data collection conditions. It is important to collect not only the instrument (along with its geographic location) but also its operating modes and settings. Any user who needs to interpret data will need to know how an instrument is being used – i.e., using a spectrometer as a photometer. (The Davis Antarctica Spectrometer is a spectrophotometer and thus has the capability to observe data that other photometers may collect).

Fig. 3 shows another part of the VSTO ontology. In this figure a data request class requires inputs from underlying constraints on how to specify files of interest, sub-selection operations, sub-setting (within a selection), etc. and these are referred to with the appropriate constraint service for each data archive. The metadata service is utilized in conjunction with the data constraints to complete the data request. Ultimately access to the underlying datasets is achieved using the data response class via the data service, which again becomes specific to the archive of interest to access and deliver (or provide a URL for) the data to a user. For example, the URL that is built for one of the data archives which utilizes the OPeNDAP-style (Open source Project for a Network Data Access Protocol; <http://www.opendap.org>) access to the data consists of a server, data path, data file selections, individual parameter(s) that are appropriate to the file and a time range within that file, viz. Millstone Hill Fabry-Perot data files from January 2000, selecting the neutral temperature parameter ‘tn’ (code 810) for 10 days out of the record. Any data archive specific access method can thus be utilized in this framework.

As a final note in regard to our initial ontology development, we elected to limit the specification and use of integrative ontologies to those related to datasets, and data products. This meant that we captured concepts such as instrument, parameter, date-time as they related to the underlying data and not higher level science concepts – such as those that start to arise out of use cases 3, 4 and 5. We imported time, space (coordinate) and realm ontologies from SWEET but did not import any integrative ontologies from any other source. The higher-level concepts are: a phenomenon, an event, a feature, etc. which necessarily connect to the underlying concepts of measured quantities. For example, an aurora (phenomenon and feature of the Earth’s ionosphere usually occurring in high latitudes) is described by certain physical parameters (auroral brightness, density, etc.) that are measured by certain instruments (operating at observatories in parts of the world that can observe the aurora) at certain times. Thus, we have added these integrative classes (and their properties) as dictated by the use cases. These classes connect to the underlying data via the more basic instrument, parameter, date-time (and space) representation we captured initially.

4.2 Tools

We chose to represent our ontology in OWL, rather than other languages such as RDF, because we needed the expressive power of OWL to capture restrictions and inter-relationships that we used to support reasoning. The reasoning was used to minimize the burden on end users who were attempting to form consistent, complete, and semantically meaningful queries that would obtain the data they were interested in.

We initially represented the taxonomy structure just in indented text form for broad circulation and agreement on terminology and structure. We then augmented the text file with property structure and value restrictions. When the team had a strong level of agreement and convergence we encoded this information in OWL-DL. We rely on a combination of editors (Protégé¹ and Swoop²). We use Protégé for its plug in support for java code generation. Earlier iterations had some glitches with interoperation in a distributed fashion that supported incremental updates but we overcame these issues and the team now uses a distributed, multi-component platform.

The definitions in the ontologies are used (via the Jena³ and Eclipse⁴ Protégé plug-ins) to generate java classes in a java object model. We built java services that use this java code to access the catalog data services. We use the PELLET⁵ reasoning engine to compute information that is implied and also to identify contradictions. The user interface uses the Spring⁶ framework for supporting workflow and navigation.

VSTO depends on background ontologies, reasoners, and from a maintenance perspective, the supporting semantic technology tools including ontology editors, validators, and plug-ins for code development. We designed the ontology to use only the expressive power of OWL-DL rather than moving to OWL-Full so that we could leverage the reasoners available for OWL-DL. Within OWL-DL, we basically had the expressive power we needed with the following two exceptions. We could use support for numerics (representation and comparison) and defaults. The current application does not use an encoding for default values. Our current application handles numerical analysis with special purpose query and comparison code. While it would have been nice to have more support within the semantic web technology toolkit, this issue is some what less of an issue for our application since the sheer quantity of numerical data meant that we needed special purpose handling anyway. The quantity of date data in the distributed repositories is overwhelming, so we have support functions for accessing it directly from those repositories instead of actually retrieving it into some cached or local store. Our solution uses semantically-enhanced web services to retrieve the date data directly.

We used only open source free software for our project. From an ontology editing and reasoning perspective, this mostly met our needs. On rare occasions, it would have been convenient to have the 24x7 support typically available from commercially supported tools. The one thing that we would make the most use of if it existed would be a commercial strength collaborative

ontology evolution and source control system. Our initial rounds of development on the ontology were distributed in design but centralized in input because our initial environment was a bit fragile in terms of building the ontology and then generating robust functional java code. We resolved the initial development environment issues and we are now doing distributed ontology development and maintenance using modularization and social conventions.

4.3 Reasoning

Our goal was to create a system usable by a broad range of people, some of whom will not be trained in all areas of science covered in the collection. The previous online data access and analysis systems required a significant amount of domain knowledge to formulate meaningful and correct queries. Previous interfaces required multiple decisions (8 for CEDAR and 5 for MLSO) to be made by the query generator and those decisions were difficult to make without depth in the subject matter. We used the background ontologies together with the reasoning system to do more work for users and to help them form queries that are both syntactically correct and semantically meaningful. For example, in one work flow pattern, users are prompted for an instrument and they may choose to filter the instruments by class. If, they ask for photometers, they will be given options shown in Figures 1 and 6, at least some of which would not be obvious by name that they can act as a photometer. An unexpected outcome of the additional knowledge representation and reasoning was that the same data query workflow is used across the two disciplines. We expect it to generalize to a variety of other datasets as well and we have seen evidence supporting this expectation in our work on other semantically-enabled data integration efforts in domains including volcanology, plate tectonics, and climate change (Fox et al. 2006b, McGuinness et al., 2007b).

The reasoner is also used to deduce the potential plot type and return products as well as the independent variable for plotting on the axes. Previously, users needed to specify all of these items without assistance. One useful reasoning calculation is the determination of parameters that make sense to plot along with the parameter specified. The background ontology is leveraged to determine for example, that if one is retrieving data concerning neutral temperature (subject to certain conditions) that a time series plot is the appropriate plotting method and neutral winds (the velocity field components) should be shown.

4.4 Unifying Concepts

During the development of our ontologies, we fully expected each of the two different initial discipline areas to require a modest core set of concepts and relations. We expected the terms to need reasonably specific representations in order to support accurate retrieval. In practice this meant we expected to build two quite different web portals, requiring differing selection workflows for the two distinct user communities. As noted earlier, the CEDAR user was used to flat listings of instruments and parameters and having to select specific operating modes (or kinds of data products) in addition to a date-time range before getting near the data. The MLSO user dealt primarily with compound data products. As a result of our use case analysis and class development, separating data products/ operating modes into their underlying

¹ <http://www.mindswap.org/2004/SWOOP/>

² <http://jena.sourceforge.net/>

³ <http://www.eclipse.org/>

⁴ <http://www.mindswap.org/2003/pellet/>

⁵ <http://www.springframework.org/>

⁶ www.planetont.org

components, we found that both CEDAR and MLSO had the basic triad of instrument, parameter and date-time at their core. We found that the related concepts such as operating mode, etc. could either be inferred or constructed using the service classes we had developed.

The result of this conclusion was that the unifying core concepts allowed us to build a single method of selection for both disciplines in a single interface (web portal and services) and differences that occurred as the user got closer to the specific selection were simplified or filled in for them by the underlying semantic web framework.

This was a significant and unexpected outcome of the ontology development and reduced development effort to one portal and set of web services to provide access to data holdings ranging from solar physics images to incoherent scatter radar data as a function of time and altitude. As we have started to add other datasets from other disciplines which feature observational data, we have found that the basic unifying concept remains. The only additional concept which we currently include but expect to explicitly expose in the near future is for spatial selection – which we have also found is common to many (but not all) observational datasets.

4.5 Maintenance and Evolution Design

Evolution and maintenance issues for ontology-enhanced applications are an active area of work in both academic and industrial settings. A nice list of requirements for “industrial strength ontology management” is available in (Das *et al.*, 2001). Ultimately, we need to address the entire list. Currently we are focusing on a smaller list of issues, some of which we report on here.

Extensible and Reusable Knowledge Representation: We designed a relatively simple set of root classes using terms of emerging best-in-class taxonomies and ontologies in the domains of interest. We made efforts to vet our design internally using use cases and externally among a broad range of science domain experts. We are finding that the ontology structure is meeting with community acceptance and is also proving to be reusable and extensible. For example, we have investigated reuse of our root classes and the related term definitions in other science areas including those required for a NASA-funded effort aimed at semantically-enabling scientific data integration in the areas of volcanoes, plate tectonics, and atmosphere. After multiple knowledge acquisition meetings with leading science experts in diverse domains required for this project, we are finding that the basic infrastructure is relatively rich in structure (i.e., the properties on instruments, observatories, data products, etc. are reusable and do not need much extension) and where extension is required (e.g., new instruments specific for new domains), it is relatively straight forward for subject matter experts to do so. On our internal project, the entire team can now make updates to the ontology and in fact, the lead scientist and lead KR expert are only consulted when significant updates are contemplated; routine maintenance is done by other team members.

We promote use case-based design and extensions. When we plan for extensions, we begin with use cases to identify additional vocabulary and inferences that need to be supported. We have also used standard naming conventions and have maintained as much compatibility as possible with terms in existing controlled vocabularies.

* Performance in large data settings: Our new system needed to be at least as robust and useful as the previously

available community system. It was imperative that our application had at least adequate performance in the face of large and growing data volumes. We designed for performance in terms of raw quantity of data. We do not import all of the information into a local knowledge base when we know that volumes of data are large; instead we use database calls to existing data services. In the present application, the data repositories have very large time records (over 65 million in one case). When we need to query over time, we convert the OWL representation to a SQL statement and execute this on the repository’s existing metadata catalog (this is represented in Fig. 3 - in the `DataRequest` class which has input from the `DataConstraint` and then the appropriate metadata service. Also in Fig. 4 in the lower left corner shows the way the external services add to the query selection workflow) We have found this method extensible to new external catalogs where that is required. Upon return of the SQL response, a class re-encodes the result into OWL and thus we are able to use reasoning, etc. as if the information was always available. Thus, we do not achieve decreased performance or functionality. We address reasoning performance by limiting our representation to OWL-DL.

* Multi-user community settings: Our approach to distributed multi-user collaboration is a combination of social and technical conventions. This is largely due to the state of the art, where there is no single best multi-user ontology evolution environment. We have one person in charge of all VSTO releases and this person maintains a versioned, stable version at all times. We also maintain an evolving, working version. The ontology is modular so that different team members can work on different pieces of the ontology in parallel. We also have our ontologies publicly available both on our service web sites and also on a jointly created community web site¹ aimed at supporting community ontology sharing.

* Provenance: We are just beginning our work on transparency and provenance. Our design leverages the Proof Markup Language (Pinheiro da Silva, et al. 2006, McGuinness et al., 2007) – an Interlingua for representing provenance, justification, and trust information. Our initial provenance plans include capturing content such as where the data came from. Once captured in PML, the Inference Web toolkit (McGuinness et al. 2004) may be used to display information about why an answer was generated, where it came from, and how much the information might be believed and why. The latter is particularly important in our science application areas when an end-user searches for and finds data that are new to them, and potentially from instruments and methods for which they are unfamiliar. At present, the need to know a lot about the data before using it is one burden that the present application of semantic technologies is intended to ease. In a new complementary NSF-funded effort, we will build out this approach to support provenance on the input stream to our virtual observatory settings.

5 The VSTO Data Framework

Figure 4 graphically displays the high-level interaction view of how selections and services are combined in the

¹ <http://jena.sourceforge.net>

VSTO data framework. Based on the background VSTO ontology (upper left) and semantic filters (which allow selection by discipline area or class/sub-class hierarchy selection, see Fig. 5 for an example) together with reasoning, the central selection procedure has been integrated across a variety of previous data workflows down to the basic combination of instrument, date/time and parameter as noted in section 4.4.

The VSTO ontology also captures concepts of services to retrieve metadata from external sources, i.e. both classes and instances not encoded in the ontology as well as data requests, data responses, etc. These are an abstraction of metadata and data services that allow a user to obtain the data that is essential for carrying our scientific investigations. Underlying these services are specific capabilities for the CEDAR and MLSO metadata and data services. These include data access with OPeNDAP, file download via HTTP, and visualization services presently using ION (IDL on the Net – software from ITT Visual Systems; <http://www.ittvis.com>). The metadata services utilize MySQL relational databases (<http://www.mysql.org>). One important aspect of the VSTO data framework is that semantic service classes were built to abstract out the capabilities of the underlying services while still allowing the re-use of the existing services with little or no modifications.

The overall software architecture is indicated in Fig. 5. Starting at the top of the figure where the initial ontology files comprising core concepts (`vsto_core.owl`) including service classes and then project domain specific concepts (`mlso.owl` and `cedar.owl`) and instances (`mlso_instances.owl` and `cedar_instances.owl`). Using the Protégé Version 3.2 software tool, we generate the Java™ class interfaces, which define the native application-programming interface (API) for the ontology and thus the data framework. Having defined the Java Object Model, the classes are instantiated with a Java runtime environment such as the Tomcat Version 5.5.2 servlet engine running within an Apache HTTP server using the Protégé Java API, the Jena triple store in memory.

The VSTO services utilize the CEDAR and MLSO services along with the Pellet reasoning engine which runs on the same HTTP server to respond to queries and reasoning operations.

The first implementation for user access was a web portal featuring three guided query/selection workflows; Instrument, Date-Time, Parameter; or Date-Time, Instrument, Parameter; or Parameter, Date-Time, Instrument. An example of the portal interface is shown in Fig. 6, which we discuss, in the next paragraph. The user interface components were developed using the Spring framework and HTML pages are served using Java Server Pages.

A later implementation added end-points for Web-Service interfaces for both the three-query service starting points as well as a data retrieval web service. These services were developed using use case 6 which expressed some technical requirements for how a services client would access the services, in what order, etc. These services are discussed in more detail in Fox et al. (2007). Fig. 7 shows one of the end-points for the QueryByInstrument web service. This provides a user who wishes to consume the service with a client, the required and optional inputs, constraints, and outputs as well as the Web Service Description Language (WSDL; Christensen, et. al., 2001) document required to invoke the service. A

user can enter some example values/ constraints and submit the request. Upon completion, a SOAP document is returned containing the response encoded in OWL and not simply XML. An example of this output, using an Instrument selection that measures neutral temperature is displayed in Fig. 8. In this figure, concepts such as instrument subclass and properties such as name, description and identifier can be seen in the upper portion of the document. This document can be used syntactically (as current non-semantic web services are now) whereby a person reads the output and uses a regular XML parser to look for hard-coded keywords. Or, a client can use the services semantically with access to the VSTO ontology and running in a similar environment to the VSTO data framework. That client then has the opportunity to do further reasoning, run queries (e.g. with SPARQL; (Prud'hommeaux and Seaborne 2007) etc. See Fox et al. (2007) for more details. The use case that motivated this web service required a SOAP-based implementation but a REST (Representational State Transfer) implementation is equally viable.

6 Conclusions and Future Needs

We have presented our knowledge representation and reasoning needs for our interdisciplinary virtual observatory project – VSTO. We used semantic technologies to quickly design, develop and deploy an integrated, virtual repository of scientific data in the fields of solar and solar-terrestrial physics. Our new VO can be used in ways the previous system was not conveniently able to be used to address emerging science area topics such as the correctness of temperature measurements from Fabry-Perot Interferometers.

A few highlights of the knowledge representation that may be of interest follow.

VSTO is proving to be an extensible, reusable ontology for solar-terrestrial physics. It is compatible with controlled vocabularies in use in the most widely used relevant data collections. Further, and potentially much more leverageable, is that the structure of the ontology is withstanding reuse in multiple virtual observatory projects. We have reviewed the ontology with respect to needs for the NSF-funded GEON project, the NASA-funded SEDI project, and the NASA-funded SKIF project. Our ontologies are open source and have been delivered to the SWEET community for integration. A web site is available for obtaining status information on this effort: www.planetont.org.

We also were able to unify the data selection workflow between the two initial distinct disciplines of solar physics and terrestrial middle-upper atmospheric physics using the core concepts of instrument-parameter-date/time, using our inferencing capabilities to fill in related but required information. Further we were able to leverage our existing set of data and plotting services, and metadata services within the new semantic data framework.

Our approach to the formal representation of the knowledge in ontologies followed a particular methodology which we believe and are finding is robust and repeatable. Key to this methodology is the combination of use cases, small expert teams, use of tools, rapid prototyping and iterative vetting of ontologies, redesign, redeploy, etc.

As our use case sophistication has grown, we have been able to build upon the core concepts in the ontology and start to add higher-level science concepts such as features, events, and phenomena, which have led to the need for more integrative reasoning and knowledge representation (additional properties, relations, range restrictions, etc.). These concepts will be added using the same use case methodology, knowledge extraction and representation we have successfully used to date. Importantly for these concepts, is that multiple interpretations are allowed as long as the formal properties, inheritance, etc. for ontologies are respected.

Yet more remains to be done to continue to advance the capabilities of virtual observatories. A more sophisticated notion is capturing the assumptions embedded in the experiment in which the data was collected and potentially the goal of the experiment. The next phase of our work will address these issues.

Additionally, we plan to augment the ontology to capture more detail for example in value restrictions and thus be able to support more sophisticated reasoning. Additionally, the current implementation has limited support for encoding provenance of data. Thus we will use the provenance Interlingua PML-P to capture knowledge provenance so that end users may ask about data lineage and be given explanations both via the web portal and via the web services.

Acknowledgements

The VSTO project is funded by the National Science Foundation, Office of Cyber Infrastructure under the SEI+II program, grant number 0431153. The authors wish to acknowledge contributions and discussions from Tony Darnell, Rob Raskin, Ken Murata and David Fulker. The National Center for Atmospheric Research is operated by the University Corporation for Atmospheric Research with substantial sponsorship from the National Science Foundation.

References

- Berners-Lee, T, Hall, W., Hendler, J, Shadbolt, N, and Weitzner, J. 2006, Enhanced: Creating a Science of the Web, Science, 313 #5788, pp. 769-771, DOI: 10.1126/science.1126902
- Bernholdt, D.; Bharathi, S.; Brown, D.; Chanchio, K.; Chen, M.; Chervenak, A.; Cinquini, L.; Drach, B.; Foster, I.; Fox, P.; Garcia, J.; Kesselman, C.; Markel, R.; Middleton, D.; Nefedova, V.; Pouchard, L.; Shoshani, A.; Sim, A.; Strand, G.; Williams, D. 2005, The Earth System Grid: Supporting the Next Generation of Climate Modeling Research Proceedings of the IEEE, Vol: 93, Issue: 3, pp: 485- 495.
- Christensen, E., Curbera, F., Meredith, G., and Weerawarana, S. 2001, Web Services Description Language (WSDL) 1.1 W3C Note 15 March.
- Cockburn, A., 2000, Writing Effective Use Cases, Addison-Wesley, Boston, MA.
- Cox, S. 2006, Exchanging observations and measurements data: applications of a generic model and encoding, Eos Trans. AGU Fall Meet., Suppl., 87(52) IN53C-01.
- De Roure, D. Jennings, N.R. Shadbolt, N.R. 2005, The semantic grid: past, present, and future, Proceedings of

- the IEEE, 93, Issue: 3, pp. 669-681, DOI: 10.1109/JPROC.2004.842781.
- Fox, P., McGuinness, D.L., Middleton, D., Cinquini, L., Darnell, J.A., Garcia, J., West, P., Benedict, J., Solomon, S. 2006a, Semantically-Enabled Large-Scale Science Data Repositories. the 5th International Semantic Web Conference (ISWC06), LNCS, ed. Cruz et al., vol. 4273, pp. 792-805, Springer-Verlag, Berlin.
- Fox, P., McGuinness, D.L., Raskin, R. Sinha, A.K. 2006b, Semantically-Enabled Scientific Data Integration. Proceedings of the Geoinformatics Conference, Reston, Virginia, May 10-12, 2006.
- Fox, P., Cinquini, L., McGuinness, D.L., West, P., Garcia, J., Benedict, J.L. and Zednik, S. 2007, Semantic web services for interdisciplinary scientific data query and retrieval, Proc. AAAI Semantic e-Science Workshop, in press.
- GeoSciML - <http://www.opengis.net/GeoSciML/>
- GEOSS – the 10 Year Implementation Plan. The Group on Earth Observations, <http://www.earthobservations.org/docs/10-Year%20Implementation%20Plan.pdf>
- Gil, Y., Ratnakar, V. and Deelman, E. 2006, Metadata Catalogs with Semantic Representations, International Provenance and Annotation Workshop 2006 (IPAW2006), Chicago, IL, Eds. L. Moreau and I. Foster, LNCS 4145, pp90-100, Springer-Verlag, Berlin.
- GML – The Geography Markup Language (ISO 19136); <http://www.opengis.net/gml/>
- Ludaescher, B., Altintas, I., Berkeley, C. et al. 2005, Scientific Workflow Management and the Kepler System. Concurrency and Computation: Practice & Experience, pp. 36
- Martin, D., Burstein, M., McDermott, D., McGuinness, D., McIlraith, S., Paolucci, M., Sirin, E., Srinivasan, N, and Sycara, K. 2006, Bringing Semantics to Web Services with OWL-S. World Wide Web Journal, to appear. Also, Stanford KSL Tech Report KSL-06-21.
- McGuinness, D.L.; Ding, L.; Pinheiro da Silva, P.; Chang, C. 2007, PML 2: A Modular Explanation Interlingua. Proceedings of the 2007 Workshop on Explanation-aware Computing (ExaCt-2007), Vancouver, Canada, July 22-23, 2007.
- McGuinness, D. L., Fox, P., Cinquini, L., Darnell, J. A., West, P., Benedict, J. L., Garcia, J., and Middleton, D. 2006, Ontology-Enabled Virtual Observatories: Semantic Integration in Practice. Proc. of OWL Experiences and Directions 2006 (OWLED2006), CEUR Workshop Proceedings, vol. 216, online at http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-216/submission_14.pdf
- McGuinness, D. L., Fox, P., Cinquini, L., West, P., Garcia, J., Benedict, J. L., and Middleton, D.. 2007a, The Virtual Solar-Terrestrial Observatory: A Deployed Semantic Web Application Case Study for Scientific Research. In the proceedings of the Nineteenth Conference on Innovative Applications of Artificial Intelligence (IAAI-07). Vancouver, British Columbia, Canada, July 22-26, 2007.
- McGuinness, D. L., Fox, P., Sinha, A. K., and Raskin, R. 2007b, Semantic Integration of Heterogeneous Volcanic and Atmospheric Data.: Proceedings of the Geoinformatics Conference, San Diego, CA., May 17-18, 2007.

McGuinness, D., L., Fox, P., Cinquini, L., West, P., Benedict, J., and Garcia, J. 2007c, Current and future uses of OWL for Earth and Space Science Data Frameworks: successes and limitations, Proc. of OWL Experiences and Directions, June 6-7, 2007 (OWLED2007), CEUR Workshop Proceedings.

McGuinness, D. L., Fox, P., Cinquini, L., West, P., Benedict, J. L., and Garcia, J. 2007d, Current and future uses of OWL for Earth and Space science data frameworks: successes and limitations. Proc. of OWL Experiences and Directions 2007 (OWLED2007), CEUR Workshop Proceedings, in press.

McGuinness, D. and Pinheiro da Silva, P. Explaining Answers from the Semantic Web: The Inference Web Approach. Web Semantics: Science, Services and Agents on the World Wide Web Special issue: International Semantic Web Conference 2003 - Edited by K.Sycara and J. Mylopoulos. 1(4). Fall, 2004.

McGuinness, D., and van Harmelen, F. 2004, OWL Web Ontology Language Overview. World Wide Web Consortium (W3C) Recommendation. February 10, 2004. www.w3.org/TR/owl-features/

Pinheiro da Silva, P., McGuinness, D., and Fikes, R. A 2006, Proof Markup Language for Semantic Web Services. Information Systems, 31(4-5), June-July, pp 381-395. Prev. version, KSL Tech Report KSL-04-01.

Prud'hommeaux, E., and Seaborne, A. 2007, editors. SPARQL Query Language for RDF, W3C Candidate Recommendation 14 June 2007.

Rushing, J., R. Ramachandran, U. Nair, S. Graves, R. Welch, and A. Lin, 2005, ADaM: A Data Mining Toolkit for Scientists and Engineers, Computers & Geosciences, vol. 31, pp. 607-618.

VOHD 2006 Virtual Observatories for Heliophysics Data NASA program; <http://nspires.nasaprs.com/external/viewrepositorydocument/77454/B.09%20Virtual%20Observatories.pdf>

Wolff, A., Lawrence, B. N., Tandy, J., Millard, K. and Lowe, D. 2006, Feature Types' as an Integration Bridge in the Climate Sciences, Eos Trans. AGU Fall Meet., Suppl., 87(52) Abstract IN53C-02.

Figure 1. Portion of VSTO ontology 1.0 indicating that with certain properties a Spectrophotometer can act as a photometer and that filtering instrument selection will include the spectrophotometer (when applicable) and that instrument choices will be available that previously were not.

Figure 2. Portion of VSTO ontology 1.0 indicating the associations between instrument operating modes, parameters, and coordinates such as time.

Figure 3. Portion of VSTO ontology 1.0 indicating the data and service classes represented and their associations. The legend in the upper right corner indicates the data and service classes, inferred inheritance, etc.

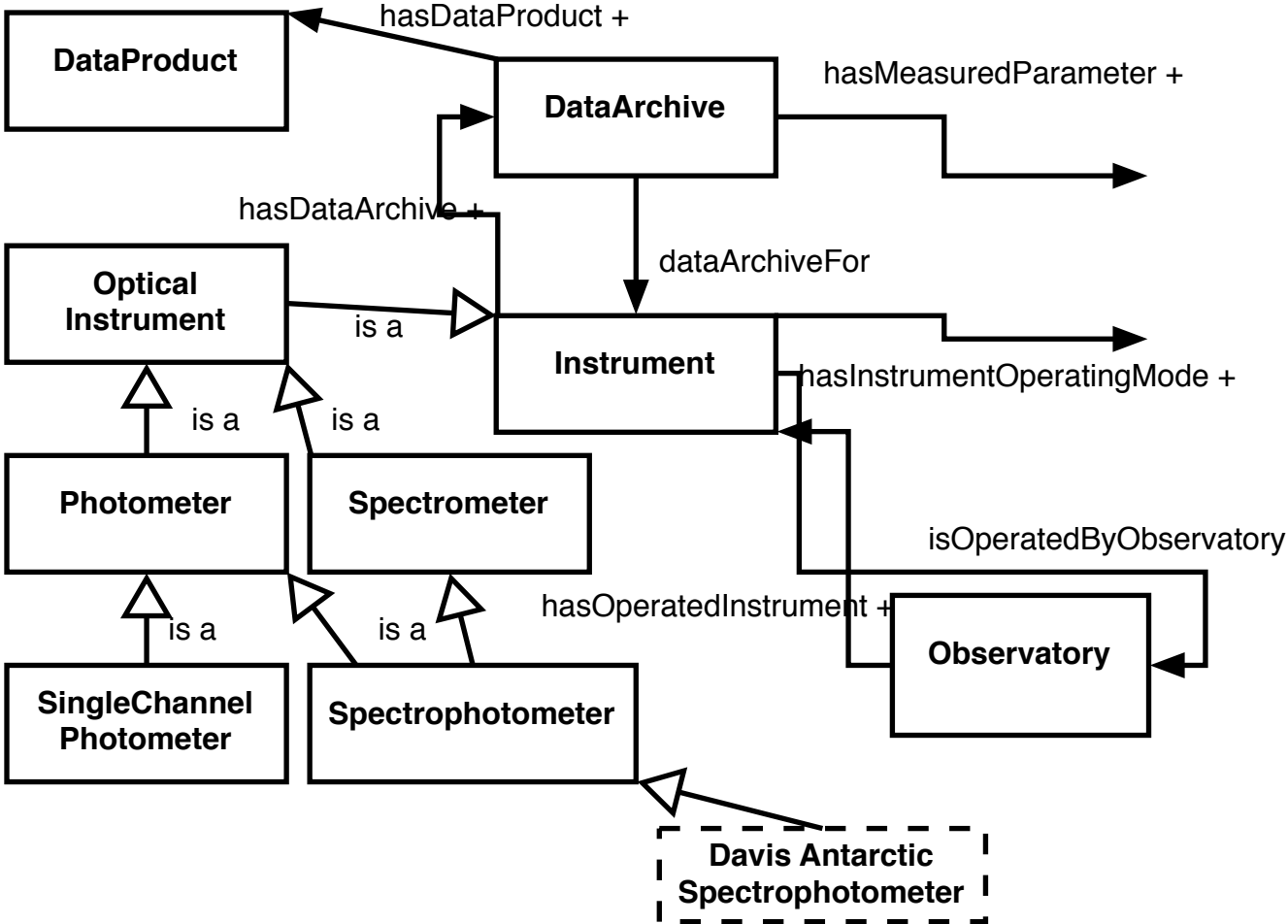
Figure 4. Relation of semantics, data selection workflow and external services for the VSTO production portal based on first two use cases.

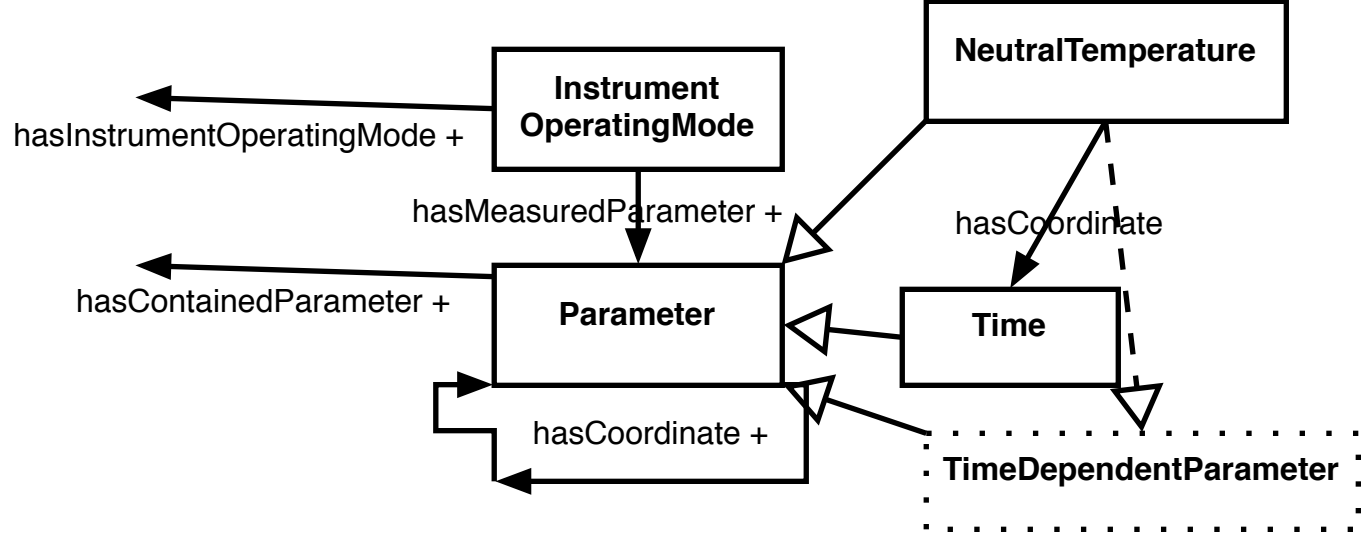
Figure 5: The VSTO software architecture layers and generation procedures.

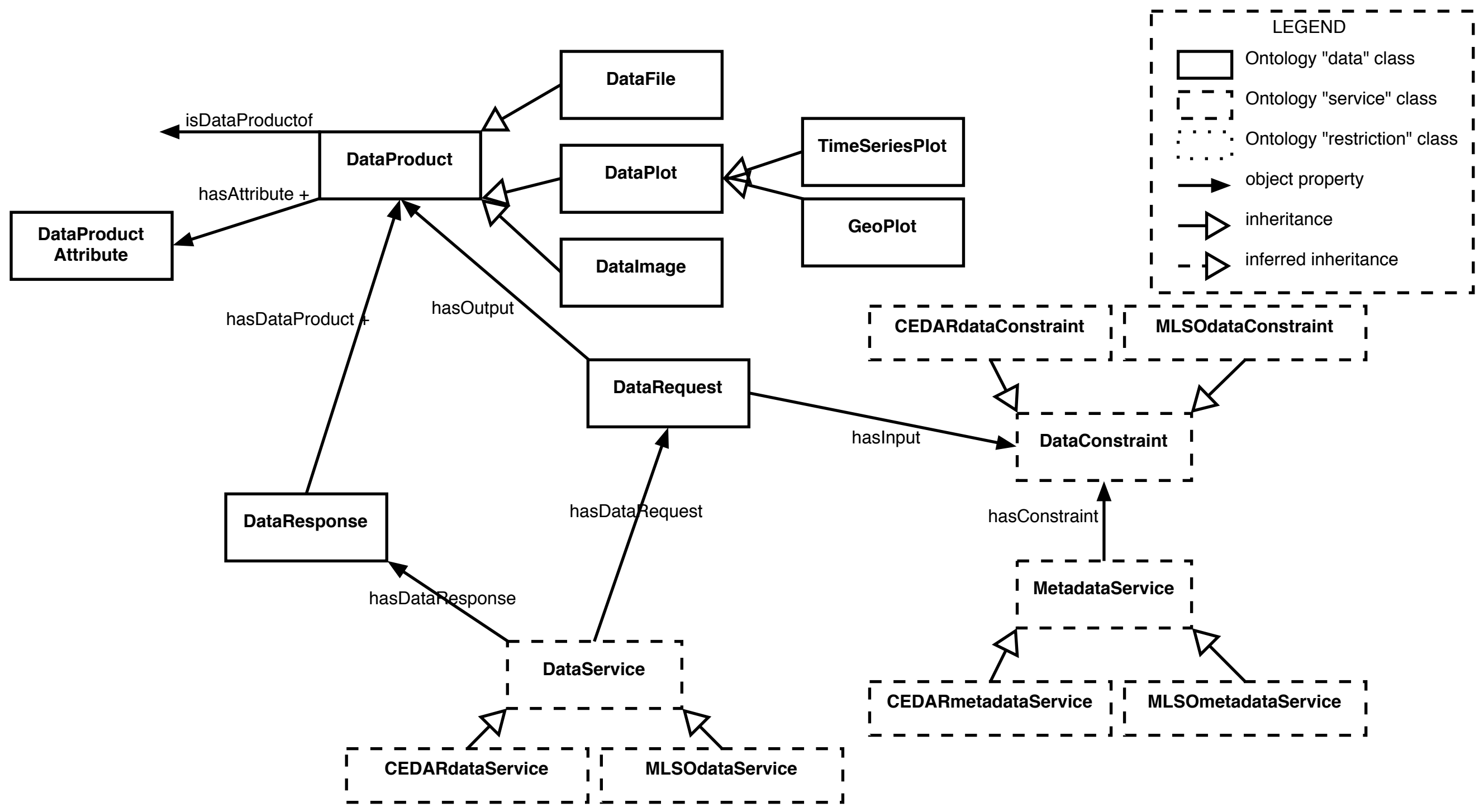
Figure 6: VSTO data search and query interface, exposing taxonomy-based instrument selection.

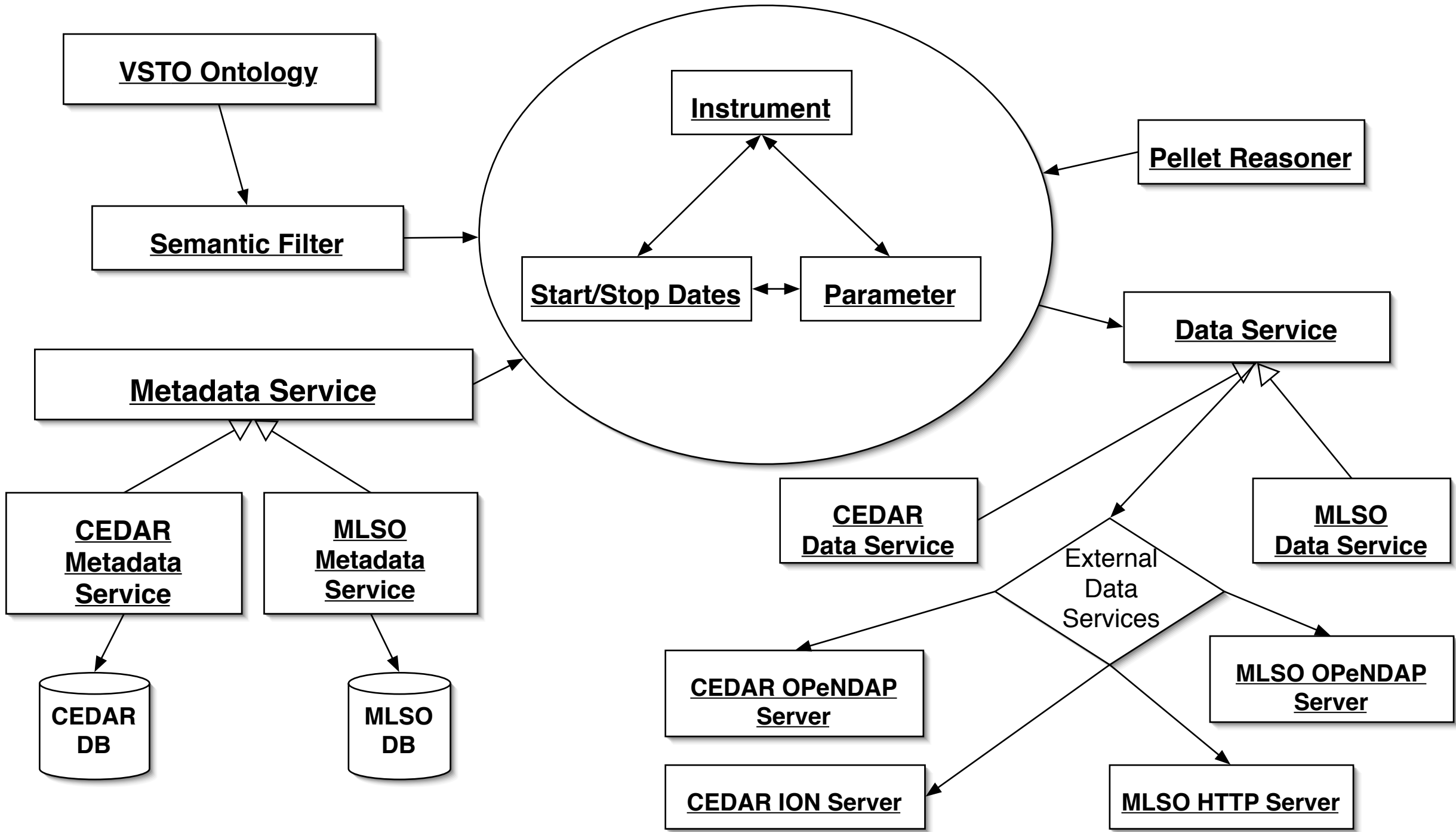
Figure 7: VSTO web services. query by instrument, end-point interface.

Figure 8: VSTO web services, query by instrument return document, in OWL.

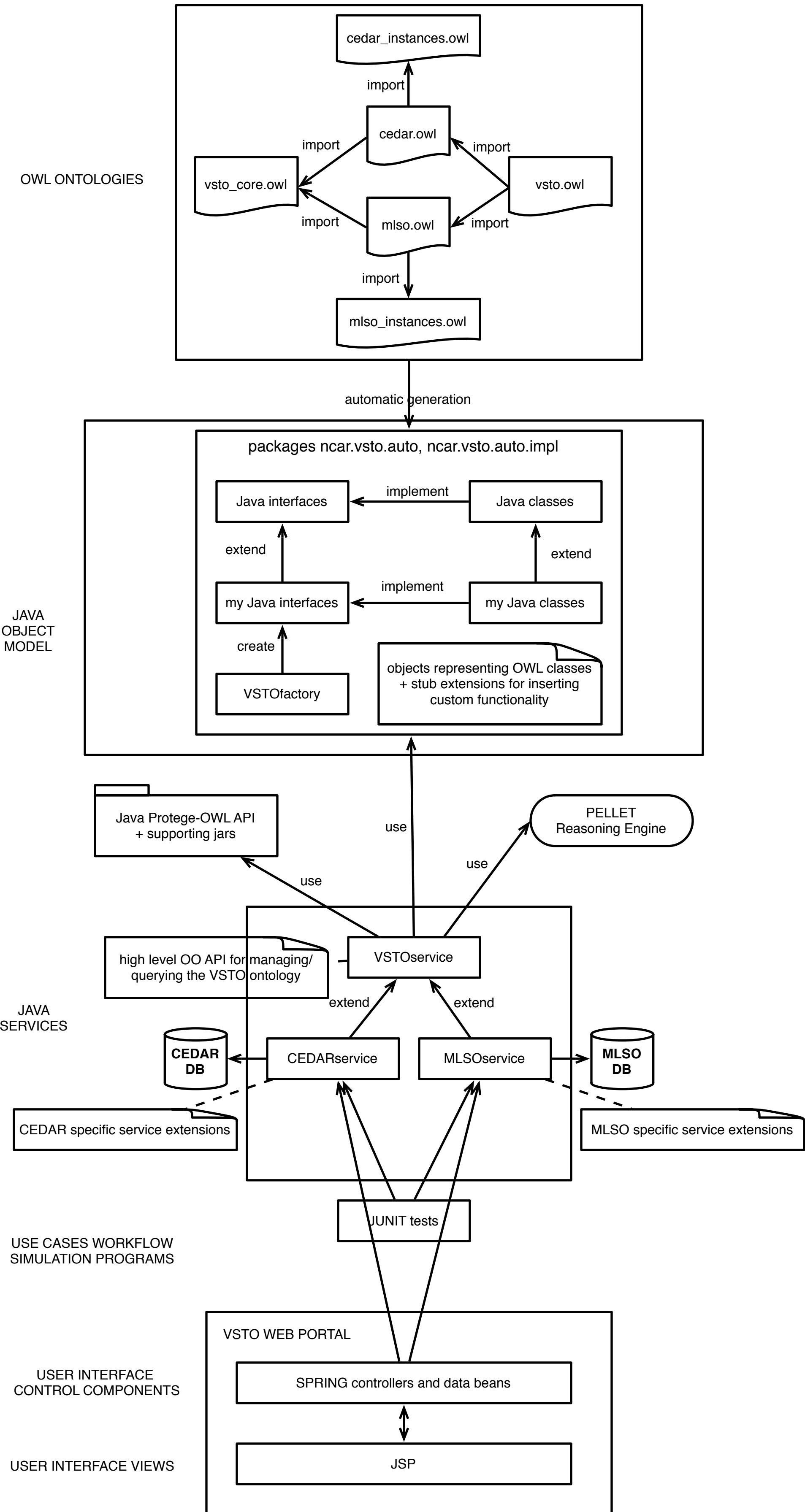








VSTO SOFTWARE DESIGN





NCAR

Virtual Solar Terrestrial Observatory

[Home](#)[Data](#)[Communities](#)[About Us](#)[Login](#)**Guided Workflows:** [Start by Instrument](#) | [Start by Dates](#) | [Start by Parameter](#) **Web Services:** [Query Instrument](#) | [Query Parameter](#) | [Query Dates](#) | [Query Data](#)**VSTO Guided Workflow: Start by Instrument****Data Request Summary****1. Instrument:****2. Start Date:**
Stop Date:**3. Parameters:****Input Step 1 of 3: Choose Instrument**

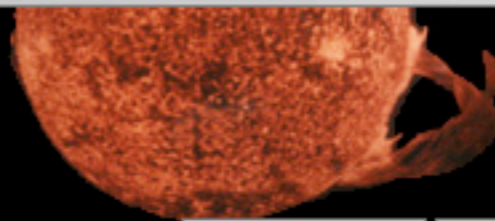
Please select an instrument

You may filter the instruments selection by *one* of the following criteria:Filter by Physical Domain: -OR- filter by Instrument Type: Show Instrument Code

- [?] Instrument:**
- OpticalInstrument > Photometer > Chromospheric Helium Imaging Photometer [?]
 - OpticalInstrument > Photometer > MK3-K Coronameter [?]
 - OpticalInstrument > Photometer > MK4-K Coronameter [?]
 - OpticalInstrument > Photometer > H-alpha prominence and solar disk monitor [?]
 - OpticalInstrument > Photometer > MultiChannelPhotometer > Poker Flat 4 Channel Photometer [?]
 - OpticalInstrument > Photometer > MultiChannelPhotometer > Fort Yukon Alaska 4 Channel Photometer [?]
 - OpticalInstrument > Spectrometer > SpectroPhotometer > Davis Antarctica Spectrometer [?]



NCAR



Virtual Solar Terrestrial Observatory

[Home](#)[Data](#)[Communities](#)[About Us](#)[Login](#)**Guided Workflows:** [Start by Instrument](#) | [Start by Dates](#) | [Start by Parameter](#) **Web Services:** [Query Instrument](#) | [Query Parameter](#) | [Query Data](#)

VSTO Web Services

Query Instrument Web Service

Description: Web Service used to query the VSTO ontology to retrieve all the Instrument instances matching one or more optional constraints.

Input: String parameterClass (optional, must be valid Parameter class name from VSTO ontology)
String startDate (optional, formatted as yyyy-mm-dd)
int nDays (required if startDate is used, must be $1 < nDays < 31$)
String domain (optional, must be 'CEDAR' or 'MLSO')
String instrumentClass (optional, must be valid instrument class name from VSTO ontology)

Output: XML/OWL document containing the Instrument instances matching the query. The XML is serialized as a String.

Exception: Thrown if invalid input is used in the query

Endpoint: <http://www.vsto.org:8080/services/VSTOqueryService>

WSDL: <http://www.vsto.org:8080/services/VSTOqueryService?wsdl>

Example: Find all Instruments that measure Neutral Temperature

Input: parameterClass='NeutralTemperature', startDate=null, ndays=0, domain=null, instrumentClass=null

Example: Find all Instruments of type Interferometer that measured data in August 1999

Input: parameterClass=null, startDate='1999-08-01', ndays=31, domain=null, instrumentClass='Interferometer'

Query Input

Use the following interface to perform a live test of the VSTO Query Instrument Web Service:

Parameter Type:

Optional: return only instruments that measured this type of parameter

Select from list:

Start Date:

(yyyy-mm-dd)

Number of Days:

Optional: return only instruments that measured data within this time interval

Domain:

Optional: return only instruments in this domain

Instrument Type:

Optional: return only instruments of this kind

Select from list:

Query Output

Number of results returned: 13

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns="http://dataportal.ucar.edu/schemas/vsto_all.owl#"
  xmlns:vsto="http://dataportal.ucar.edu/schemas/vsto.owl"
  xmlns:cedar="http://dataportal.ucar.edu/schemas/cedar.owl#"
  xmlns:mlso="http://dataportal.ucar.edu/schemas/mlso.owl#"
  xmlns:owl="http://www.w3.org/2002/07/owl#" xml:base="http://dataportal.ucar.edu/schemas/vsto_all.owl">
  <vsto:FabryPerot rdf:ID="cedar_instrument_5000">
    <vsto:hasDescription>South Pole Fabry-Perot Interfer Spectr</vsto:hasDescription>
    <vsto:hasName>SPF</vsto:hasName>
    <vsto:hasIdentifier>5000</vsto:hasIdentifier>
  </vsto:FabryPerot>
  <vsto:FabryPerot rdf:ID="cedar_instrument_5015">
    <vsto:hasDescription>Arrival Heights Fabry-Perot Interf Sp</vsto:hasDescription>
    <vsto:hasName>AHF</vsto:hasName>
    <vsto:hasIdentifier>5015</vsto:hasIdentifier>
  </vsto:FabryPerot>
  <vsto:FabryPerot rdf:ID="cedar_instrument_5060">
    <vsto:hasDescription>Mount John New Zealand Fabry-Perot</vsto:hasDescription>
    <vsto:hasName>MJF</vsto:hasName>
    <vsto:hasIdentifier>5060</vsto:hasIdentifier>
  </vsto:FabryPerot>
  <vsto:FabryPerot rdf:ID="cedar_instrument_5140">
    <vsto:hasDescription>Arequipa, Peru Fabry-Perot</vsto:hasDescription>
    <vsto:hasName>AQF</vsto:hasName>
    <vsto:hasIdentifier>5140</vsto:hasIdentifier>
  </vsto:FabryPerot>
  <vsto:FabryPerot rdf:ID="cedar_instrument_5300">
    <vsto:hasDescription>Peach Mountain Fabry-Perot</vsto:hasDescription>
    <vsto:hasName>PPF</vsto:hasName>
    <vsto:hasIdentifier>5300</vsto:hasIdentifier>
  </vsto:FabryPerot>
  <vsto:FabryPerot rdf:ID="cedar_instrument_5340">
    <vsto:hasDescription>Millstone Hill Fabry-Perot</vsto:hasDescription>
    <vsto:hasName>MFP</vsto:hasName>
    <vsto:hasIdentifier>5340</vsto:hasIdentifier>
  </vsto:FabryPerot>
  <vsto:FabryPerot rdf:ID="cedar_instrument_5430">
    <vsto:hasDescription>Watson Lake, Canada Fabry-Perot</vsto:hasDescription>
    <vsto:hasName>WFP</vsto:hasName>
    <vsto:hasIdentifier>5430</vsto:hasIdentifier>
  </vsto:FabryPerot>
  <vsto:FabryPerot rdf:ID="cedar_instrument_5460">
    <vsto:hasDescription>College Fabry-Perot</vsto:hasDescription>
    <vsto:hasName>CFP</vsto:hasName>
    <vsto:hasIdentifier>5460</vsto:hasIdentifier>
  </vsto:FabryPerot>
  <vsto:FabryPerot rdf:ID="cedar_instrument_5465">
    <vsto:hasDescription>Poker Flat AK Scanning Imaging F-P Spec</vsto:hasDescription>
    <vsto:hasName>PKF</vsto:hasName>
```