VxOware: Software for Managing Virtual Observatory Metadata

R.S. Weigel, M. Zhizhin, D. Mishin, D. Kokovin, E. Kihn, and J. Faden

Abstract

The recent Heliophysics Virtual Observatory (VxO) effort involves the development of separate observatories with a low overlap in physical domain or area of scientific specialization and a high degree of overlap in metadata management needs. VxOware is a content and metadata management system and is intended for use by a VxO specifically, but can also be used by any entity that manages structured scientific metadata. VxOware has many features of a content management system and is fully based on the W3C recommendations for XML, XQuery, and XSLT. VxOware has features such as system and user administration, search, user-editable content, version tracking, and a wiki. Besides virtual observatories, the intended user-base of VxOware includes a group or an instrument team that has developed a directory structure of data files and would like to make this data, and its associated metadata, available in the virtual observatory network. One of the most powerful capabilities of VxOware is the ability to link any type of object in the observatory to other objects and the ability for every object to be tagged.

1 Introduction

1.1 Motivation and overview

There are many software components that must be pieced together to form the infrastructure of a virtual observatory. To date, there has not been an effort to synthesize the metadata-related tools and applications into a coherent application that is general enough for use by a group that seeks to develop a Virtual Observatory (VxO). The goal of a virtual observatory is to improve and simplify access to data used for scientific research. To do this, the VxO first needs to organize their metadata. In this paper, we will use "Virtual Observatory" to mean a specialized web portal that unites, simplifies, or improves access to the data ("observations") required for research in community "x".

The main components of VxOware are shown in Figure 1.

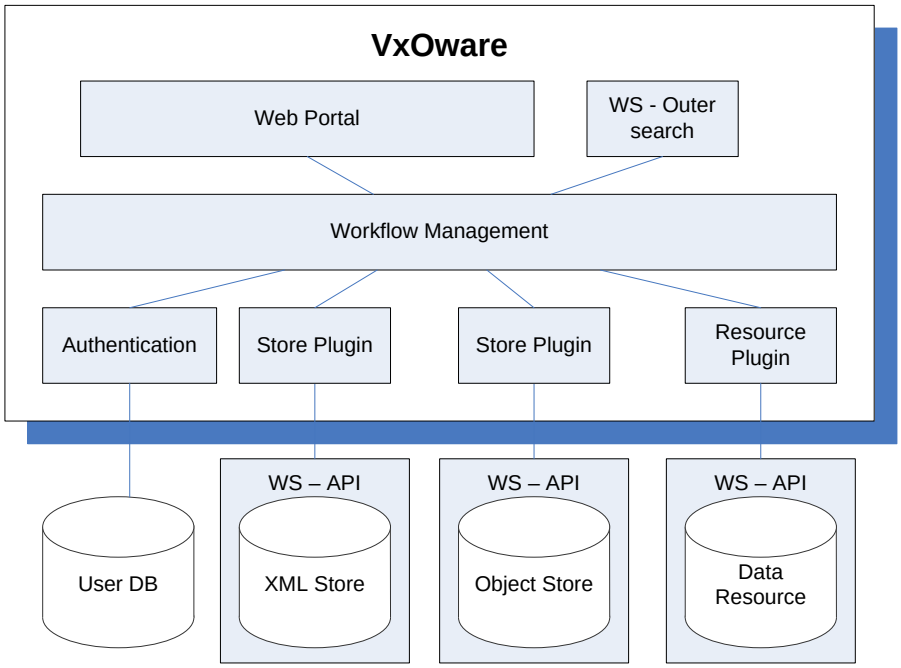

Figure 1. Overview of VxOware components. This release of VxOware is focused on the handling of metadata; although plug-ins exist for data resources, they are not described in this paper except in section 4 where example implementations are discussed. Section 2 covers the available user features of VxOware, which include the Web Portal and Workflow Management software. Section 3 covers (a) the store plug-in, which is a programming interface for unified access to a data storage unit, (b) the resource plug-in, which is an embeddable programming interface between a data source and the VxO, (c) the User DB, which is a database containing user account information, (d) XML storage, which contains information such as meta-records, top-level observatory configuration, and user profiles, (e) object storage, which is used for the storage of data objects obtained from external data sources and user-contributed objects (such as scientific papers or presentation documents), and (f) the data resource, which represents a remote data

source. Section 3 covers the Outer Search, which is a REST web service that allows other applications to access or query the VxO's holdings.

1.2 Sections

The section is the most basic grouping of metadata records in VxOware. An implementation of VxOware may contain any number of sections, each of them having different types of metadata records. Each section has an associated XML schema and may have a render XSLT file. For example, one section may contain metadata records that follow the RSS document model. When a user selects a metadata record from this section, an XSLT transform is applied to the raw XML to render it in HTML. Besides a render XSLT file, the section may have an edit XSLT file that creates an editable HTML form. Figure 2a shows a screenshot in which a previously stored metadata record has been rendered in HTML using a render XSLT file that is associated with that section. Figure 2b shows the rendering of the same metadata record using an edit XSLT file.

Any XML metadata schema can be used to form a section. The predefined VxOware sections include News, which is used for notifications and automatically generated messages and is also the source of the VxOware RSS feed; Blogs, which contains records for user notes, Message Boards, which contains discussions between the users; and Documentation, which has an editor that accepts wiki-formatted text. Other schemas have been used in VxOware and have render and edit XSLT files available. These schemas include SPASE [http://spase-group.org/], FGDC [http://www.fgdc.gov/],

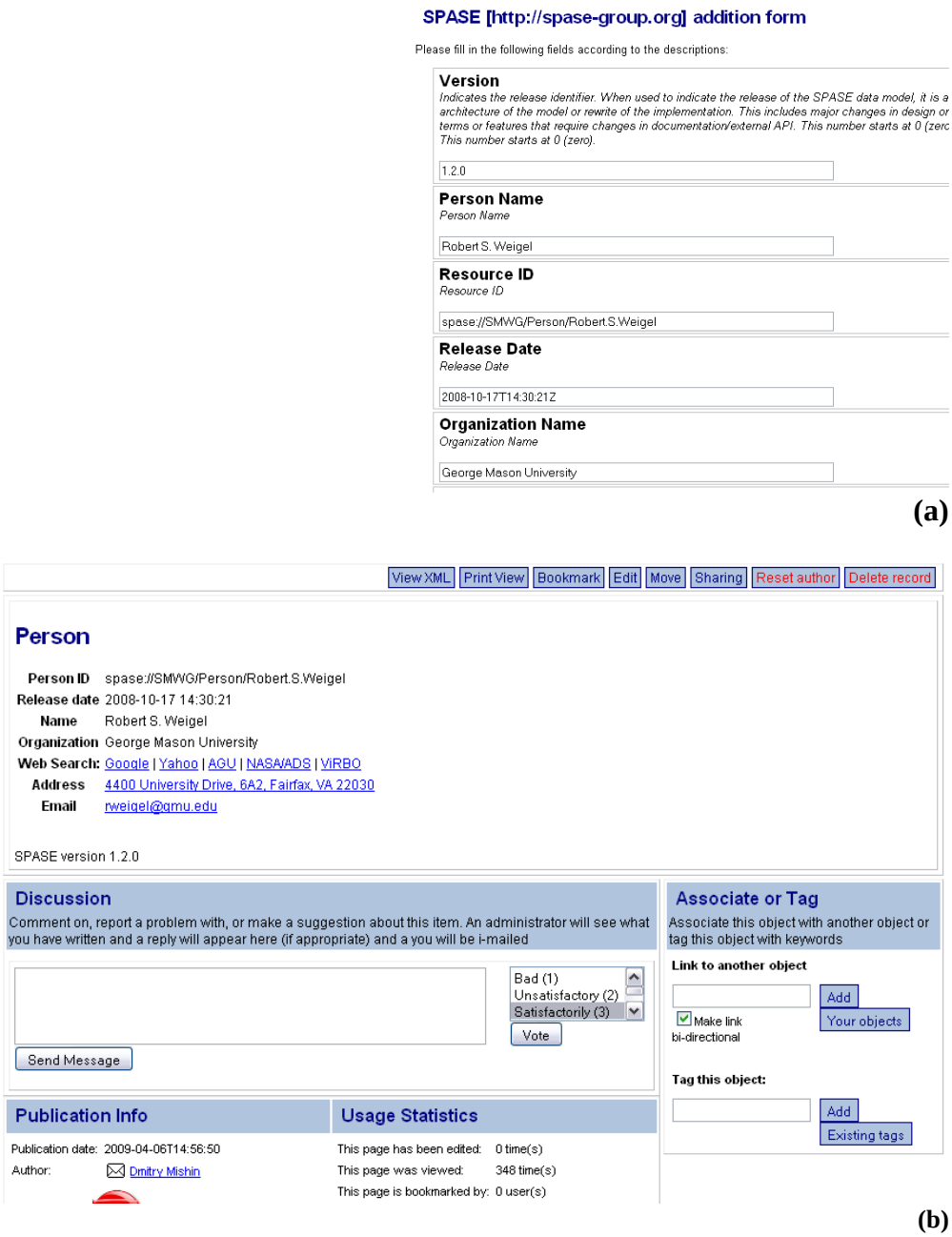

Figure 2: (a) Screenshot of an editable form for a metadata record in a section named "Person". A raw XML record was transformed into this HTML form using the edit XSLT file that is associated with this section. (b) Screenshot showing the same metadata record rendered with the view XSLT file associated with this section. The XSLT file uses the raw metadata to create links to other services. For example, the Google and Yahoo links do a search on the person's name, the NASA/ADS link does a search for journal and conference abstracts in NASA's Abstract Data Service, and the address link is connected to the Google Maps REST API.

1.3 Objects

VxOware stores three types of objects:

1. A metadata record, which is an XML file describing a scientific data product. Each metadata record has an associated forum record. The forum record is created at the time the metadata record is ingested and contains information such as when the metadata record was submitted, when it was modified, links to related metadata records, and user-contributed tags. Links and tags are described in section 2.5.
2. A VxO record, which is an XML file that contains auxiliary information about a metadata record. VxO records include forum records, log records, and user profile records.
3. A binary object, which could be, for example, a PDF-formatted file or an image.

Objects have the following basic properties:

- a unique identifier,
- a name and a short description,
- an association with a certain section,
- are stored in a native XML database (eXist-db; [http://exist-db.org/]),
- are available for searching and viewing,
- have an activity counter (number of reads, edits, and bookmarks),
- may be bookmarked,
- may be modified by a user with the appropriate rights,
- have a change history,
- may have a “weak” link with other objects. (Weak linking allows to logical connections between different objects that relying on certain criteria, for example, all documentation can be tagged as “documentation”.),
- may have a “strong” link with other related records. (All objects in a top-level observatory section are strong-linked, for example.),
- may be opened for discussion and voting (rating among all records),

Object properties can also be used for grouping search results (for example, sorting by a certain property in ascending or descending order).

1.4 Community

An important part of the mission of a VxO is to provide an environment for efficient communication, to foster a sense of community, and to promote best practices related to metadata standards. VxOware has a number of features to support this:

- Users may exchange personal messages by an “internal e-mail” (i-mail) service. Each user has a personal mailbox and can send i-mail to any other user. The user also has the ability to have i-email forwarded to an external email address.
- Each user can have a personal page (blog) for recording notes on, for example, such topics as data usage or general items of interest throughout the VxO. A user also has the ability to make their blog visible for other users, which makes it possible to post reports on work progress or resource development. A blog may

also serve as a personal notebook for recording user-specific notes. A user can create a blog that allows comments to be added by other users, thus creating an additional communication channel.

- The observatory has a discussion system whereby each object in the observatory has a thread that is bound to a parent object. This allows the object contributor to get directed feedback on contributed metadata, software, or documents.
- When adding a new object, the author may receive feedback from other users about notifications of broken links or data quality reports. News items are also available via the same system. The object's author and all users who have added any object to their "basket" will receive i-mail notifications when new posts appear in the corresponding discussion of that object. This allows users to automatically track revisions of key metadata, for example.
- VxOware tracks, and reports on the same page of the object, the number of users that have accessed or viewed the object. This serves to allow object contributors to easily determine usage frequency.

2 User Features

2.1 Workflow

There are three groups of VxOware users which are defined by their privileges:

1. Anonymous users may search, browse, and download metadata;
2. Registered users have the ability to modify user-specific preferences and modify records in certain VxO sections; and
3. Administrators have full access to VxO data management and configuration.

A VxO workflow for an anonymous user could consist of the following steps:

- Log in using a conventional web-browser;
- Browse through records in a section of interest; and
- Perform a metadata search.

A VxO workflow for a registered user could consist of the following steps:

- Browse through the metadata records that have been added since their last login;
- Contribute to the discussions on existing or new records, or create a new discussion topic;
- Load their own information (e.g., links to information resources) in a public metadata section; and
- Modify or update metadata.

A VxO workflow for an administrator could consist of the following steps:

- Accept or decline the addition of items that require moderation and
- View any comments left by anonymous users about problems with resources or questions about resources

Several basic components are required to support these workflows: a user authentication system, user profiles records, an object storage system, and interaction services. These components are discussed in the following sections.

2.2 Authentication

VxOware uses a special database for user authentication, shown as "User DB" in Figure 1. The User DB contains basic user profile information including username, password, user information, and user preferences. The User DB is isolated from the other data bases.

2.3 User space and user profile

The user profile is stored as a structured metadata record and contains a description of the user's preferences and personal information. The user space builds on the user profile and contains a record of the user's VxO contributions such as contributed objects, comments in the discussion section of an object, and personal objects including i-mail, bookmarks, and blog entries.

When a new user account is created, a personal profile is created. The user profile includes:

- personal information (name, scientific interests, picture);
- settings and preferences including:
    - interface appearance (color scheme and font size);
    - a section-update filter that allows the user to choose sections of interest; each time these sections are updated or modified, the user will receive notification via i-mail; and
    - a discussion-update filter that allows the user to receive update notifications for bookmarked discussion topics or topics that the user created.

The record of the user's contributions and use of services after their account is created form the user space. The user services that are available to a registered user include:

- an internal messaging service (i-mail);
- an internet journal (blog);
- the ability to add objects to certain sections;
- the ability to add comments in the discussion section of an object;
- a notification (update monitoring) service;
- a list of VxO objects that have been contributed or modified by the user; and
- user-created metadata records. Each record is managed by the user who created it. The author has access to previous versions of the record and can allow others to edit his records by modifying the list of co-authors. The co-authors receive a notification when the discussion topic related to the record is updated;

- a bookmark list. Bookmarks allow for the creation of a permanent link to any observatory object. The user receives a notification when new messages appear in the discussion element of the bookmarked record;

2.4 Search

Search is an integral part of VxOware. VxOware allows for metadata searching not only for data and data services, but also for all information content in the VxO. News items, software packages, presentations, publications, reports, and wiki pages all have metadata stored in a native XML database (eXist; http://exist-db.org). This allows context-based searches for not only data sets but also for key elements which support the use of that data (i.e. documentation and software).

There are three kinds of searches for data stored in a VxO built on VxOware: free-text, context, and external.

1. *A free-text search* is used for full-text search in all non-binary objects in the VxO.
2. *A context/attribute search* is done taking into consideration the properties of each type of record and is aimed at the contents of the record. The search space can be limited with one or more metadata record elements.
3. *An external search* may be made on VxO sections that are available using a framework of web-services. A key feature of VxOware is that a federation of instances can be created and connected. For example if another VxO wanted to provide search capability to the "News" section of another VxO in their federation, they would specify that their News section was available for an external search. To make a section available for external search, an administrator adds it to a list of sections that are web-service accessible along with a list of metadata fields that are available for search. One advantage of this approach is that a VxO can select sections that can be searched by all users in the federation along with sections which cannot be searched unless the user has visited their specific observatory (for example to prevent outsiders from searching the contents of sections that are under development).

2.5 Linking and Tagging

Every object in the VxO can be linked to another object in the observatory or tagged. A screenshot of the link and tag interface is shown in Figure 3. When the user enters the URL to another VxO object in the link field, the link is stored in the metadata record's forum file. When another user visits this metadata record, they will see that a different user has decided that another VxO object is connected in some way to that metadata record. A metadata record may also be tagged. In this case a user may specify a set of keywords that describe the record. The tags are also stored in the metadata records forum file.

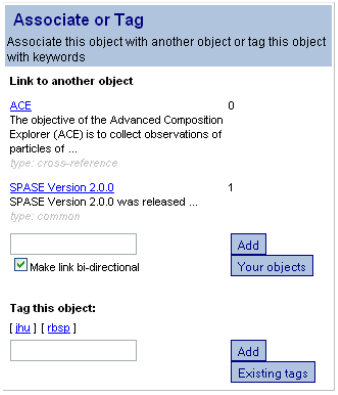

Figure 3: This metadata record has several links and tags associated with it. The record has information about an instrument and another user has added a link to a metadata record that has information about the satellite that the instrument is on and a link to a place where data from this instrument are available. (Ideally, the metadata schema would formally support such links, but this approach can be used as a replacement.)

2.6 Update monitoring

All objects in the observatory have the option of being tracked via an RSS feed by any type of user.

There are three methods by which a registered user may track or monitor updates or changes to the VO. First, they may define personal settings that filter all incoming metadata records by sections. Each time the user logs into the system, they may browse a list of all current metadata updates in the sections specified in their personal settings. Second, a user may track individual items by bookmarking them. When a bookmarked item has been modified or deleted, the user will receive an i-mail. Finally, the main navigation bar shown in the upper left panel of Figure 4 has a column that shows the number of items that have been added to each section since the users last logged in.

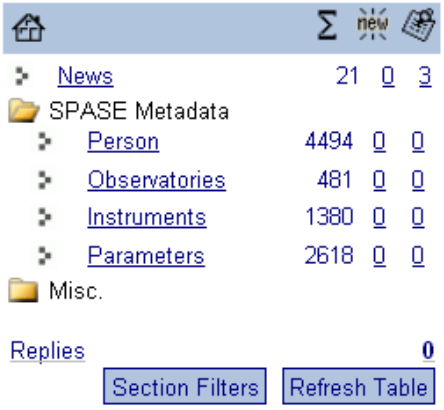

Figure 4: Left-panel view for logged-in user. The three columns with numbers show the number of items in the section, the number of new items in the section since the user last logged in, and the number of items in the section that match the user's update filter.

2.7 Administration

There is a special type of VxO user called an administrator. They are responsible for the maintenance of the observatory. Administrators may

1. delete objects,
2. change global and user-level settings, and
3. edit all VxO objects without restriction.

2.8 Editable content and wiki features

One of the most unique features of VxOware is the version control of user-contributed objects and the ability for users to generate version-controlled documentation and notes using a wiki in a manner similar to many standard content management systems. However, in contrast to the majority of existing content management systems, VxOware uses XSLT, XML, to enable management of metadata for modern web applications comparable to YouTube, expert systems, and cloud computing services. A number of existing applications that use VxOware are described in section 4.

The wiki features are intended to be an improvement over the current state of data servers' communication with users. As an example, consider the situation in which a user discovers a problem with a data file. The current approach is for the user to email a contact person who is then expected to make a fix at some point. With VxOware, a user will be able comment on a data file, and this comment will be viewable by other users. With this system, users who experience the same problem before the problem is fixed will not be forced to write the same email. In addition, suppose the original user's problem was not due to a faulty data file, but rather a bug in their file reader. With an exchange between the data provider contact and the data user that is viewable from a convenient location any other user that experiences the same problem will not need to search the FAQ and/or email the contact person with the same question. The improvement that this system provides is

- The expert knowledge exchanged between the user and manager is on public record, and not inaccessible to others because it is in a location where a user will most likely find it, as opposed to being the personal email archives of a few people.
- Data providers are more likely to fix a problem that is publicly documented.
- Based on other exchanges, a user will have a good idea of what to expect in terms of responsiveness from the data center.

.

Another unique feature of the VxOware is the Software section, which allows users to upload packages or source code for analysis programs. Users of a software package have the option of being notified when a new version of the software is available. This feature is intended to be an alternative to the typical approach used by researchers in which the user must re-visit a page to see if the software has been updated. This feature is not intended to be a replacement for the model where a user subscribes to a project's email notification list, but is rather intended to be a simplified model of this option. It is our experience that many scientists are not interested in managing their software packages with a full-featured source code management system, but rather only need a simple process for communicating changes to their software to interested parties.

3 Storage, API, and Services

3.1 Storage

The types of storage are shown in Figure 1 and their basic layout is described below.

- User DB: This database contains: usernames, emails, last-logged-in date, credential (e.g., admin, user).
- XML DB: This is an eXist data base that contains directory called “profiles”, one directory for each section, and a directory named “logs”.
- Object DB: This database contains binary objects and the data are stored on disk using the operating system’s filesystem.
- Data Resource: This can be any time of data base that contains scientific data. For each data base, a plugin is required.

3.2 Upload Services

Users may contribute objects to the VO in three ways:

1. Direct upload of a single record or object. When a user is logged in, they will see an “upload record” button when they visit an observatory section.
2. Direct upload of a compilation of records using the “upload record” button. The user can create a zip file containing any number of records for a given section.
3. Submission using a command line interface. The user can use a script that takes an input of the local directory path to a record (or a directory containing records to be uploaded), the section that it should be submitted to, and the user’s login credentials.

An administrator may also upload objects using the eXist-db interface. The Exist codebase allows for a number of ways to put files in the database. This is not a recommended procedure because when a metadata record is directly inserted into the database, it will not have an associated forum file.

3.3 Search Services

The external search web-service is available via the REST protocol. The web-service returns a configuration-based self-description of the VxO (information about the available capabilities) with a list of sections open for search. On requesting the information for a certain section, the web service returns a list of meta-record elements (and possibly a set of valid values), with values that may be used to find objects in this section. With this self-description, a context search query can be created which results in a list of objects. An important feature of the external search web-service is its dynamic result format. You may choose which of the available meta-record fields you need to see in the query result. These properties of the web-service make it suitable for case-based reasoning applications.

4 Applications

There are a number of applications that use VxOware for a wide variety of virtual observatory activities. Some of these implementations use features that were not described in this paper including data visualization and data retrieval.

4.1 Virtual Radiation Belt Observatory (ViRBO)

The Virtual Radiation Belt Observatory [http://virbo.org] is an instance of one of the 5 VxOs supported by a recent NASA program [http://hpde.gsfc.nasa.gov/]. The general goal of the ViRBO project is to integrate information about radiation belt research and data resources.

Because of the size and nature of the radiation belt community, ViRBO uses a number of the community development aspects of VxOware. The primary sections used by ViRBO are Data, Metadata, Documents, Software, News, and Wiki. The default user configuration allows any registered user to upload documents (such as presentations, papers, and movies).

ViRBO uses the SPASE metadata schema to catalogue data resources.

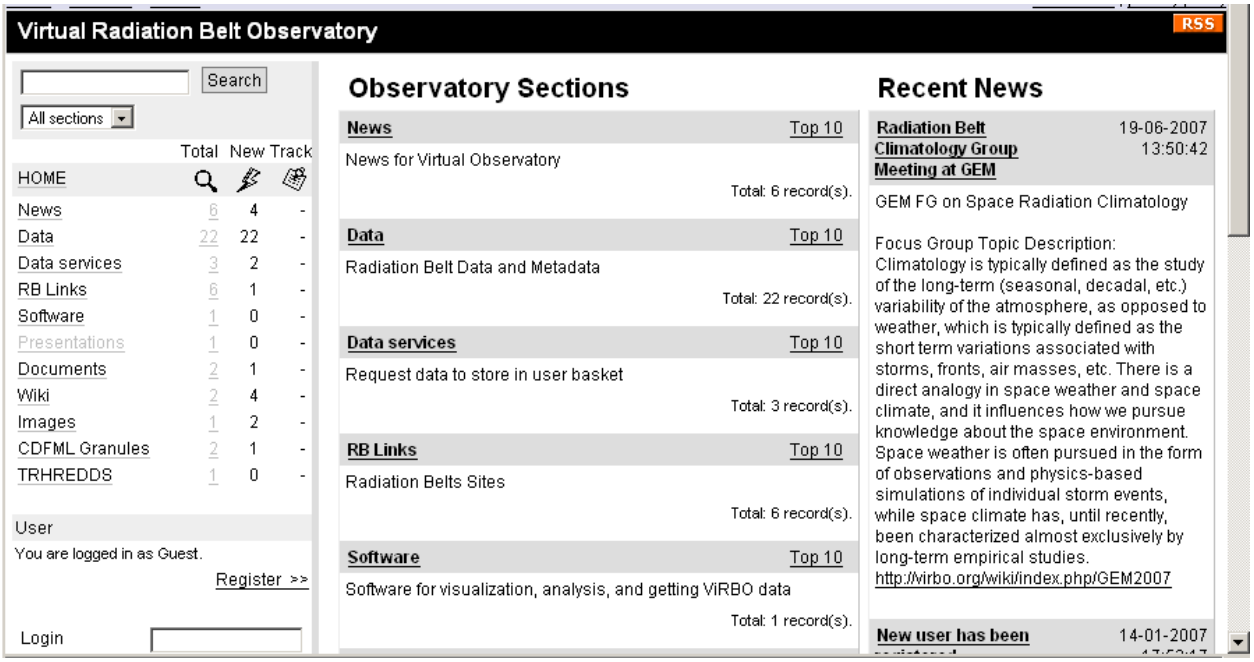

Figure 5: Homepage of the ViRBO metadata engine.

4.2 Comprehensive Large-Array Data Stewardship System (CLASS)

The CLASS application provides an interface to the large satellite data archive located in NOAA. Users may access different data resources from an extensive satellite product list.

The sections created for this application contain metadata in the FGDC format and data service descriptions in the Ordering Extensions (OE) format. The VO for CLASS provides metadata storage, search and order form creation based on OE records. The interaction of CLASS with other VOs is carried out using the VxOware software, which is integrated with the general CLASS structure. For request processing an external search service is used, and for metadata records there is direct access to meta-record storage.

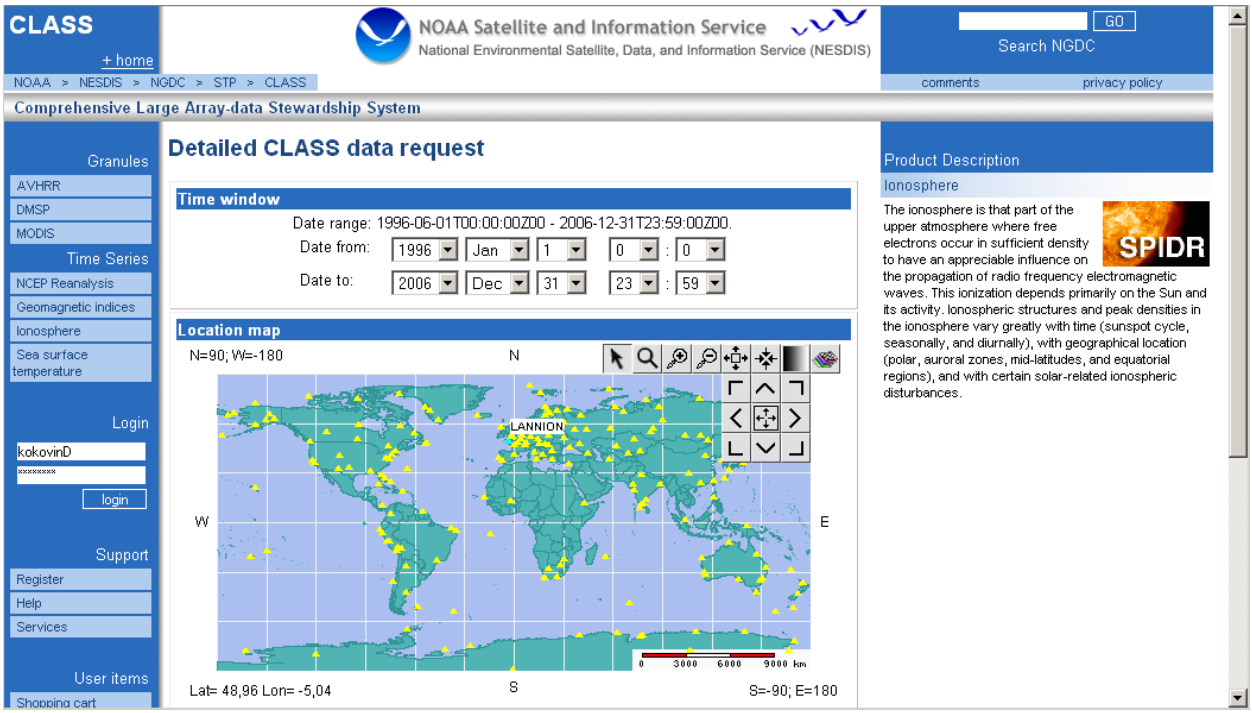

Figure 6: Homepage of the CLASS system

4.3 Space Physics Interactive Data Resource (SPIDR)

SPIDR is a distributed network of synchronous databases and application servers designed to allow a modeling and prediction customer to intelligently access and manage historical space physics data for integration with virtual environment models and real-time space weather forecasts (http://spidr.ngdc.noaa.gov). SPIDR was originally developed by the National Geophysical Data Center (NGDC) in support of the international GOIN project. It has since evolved into a joint development effort with the Russian Academy of Science and is now more than 10 years old with more than 10,000 users. VxOware is used to manage its large metadata holdings.

Most of the space weather time series stored in SPIDR databases and provided to users by its web-services and web-portal are observed at ground stations or satellites. VxOware part of SPIDR provides metadata search web service to the XML-formatted records describing the data sets and observatories, including links to the metadata from SPIDR web-portal maps and data export pages. Data managers from the observatories and satellite operators can use the SPIDR VxOware administrator interface to edit metadata records related to their data product and services.

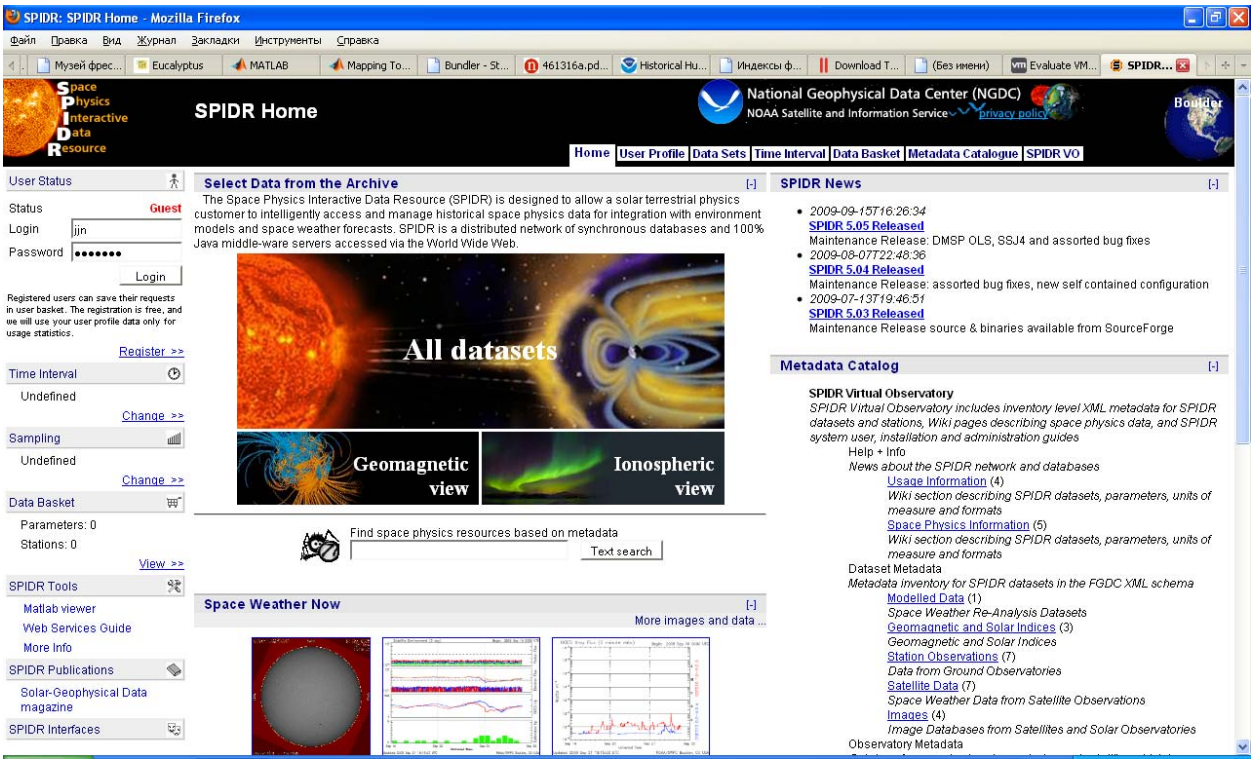

Figure 7: SPIDR home page; VxOware is used for metadata search and the news and subject tree sections.

4.4 Expert system for Recognition of Toxic Events (RTE)

The expert system for Recognition of Toxic Events (RTE) was developed as a part of the European Union R&D project Open Advanced System for dISaster and emergency management (OASIS, http://www.oasis-fp6.org/). The expert system knowledge base for the RTE application contains information about toxic substances and methods of overcoming the consequences of accidents involving dangerously explosive, flammable and toxic substances. Its XML schema shown in Figure 8 is derived from the Wireless Information System for Emergency Responders (WISER, http://wiser.nlm.nih.gov/) developed by the National Library of Medicine of the National Institutes of Health. Depending on the problem domain, the knowledge base can be extended with additional fields and elements, for example, with separate document collections linked by a common substance ID, which allows for different representations (views) of the substance data.

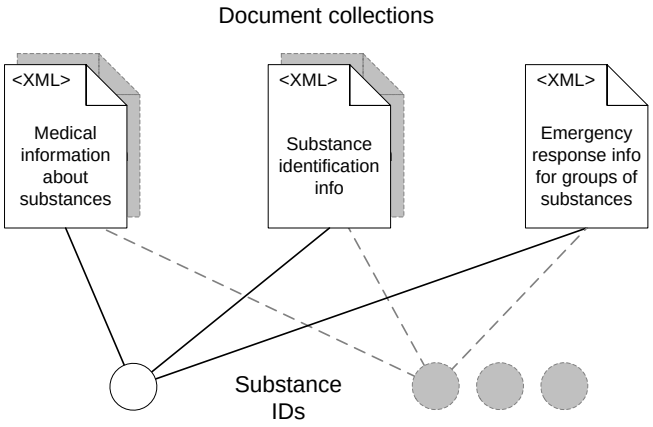

Figure 8: RTE knowledge base structure.

The RTE expert system allows the users to recognize a toxic event based on the emergency response data such as physical properties of the substance and medical symptoms registered in the majority of the population. VxOware middleware is used by RTE to build a web-application layer above the XML knowledge base. It allows for interoperability with other applications providing self-describing metadata (capabilities) and data search web-service.

To identify substances by their properties and symptoms RTE uses a special substance identifier module. This module queries the knowledge base (it may be supplied with additional logic facilities, such as synonyms dictionary, fuzzy logic, etc.) and returns a list of candidate substances, which may only approximately satisfy the search criteria based on partial description such as color/physical state/medical symptom. The search web-service returns the list of candidate toxic substances to the OASIS portal in a background using AJAX protocol.

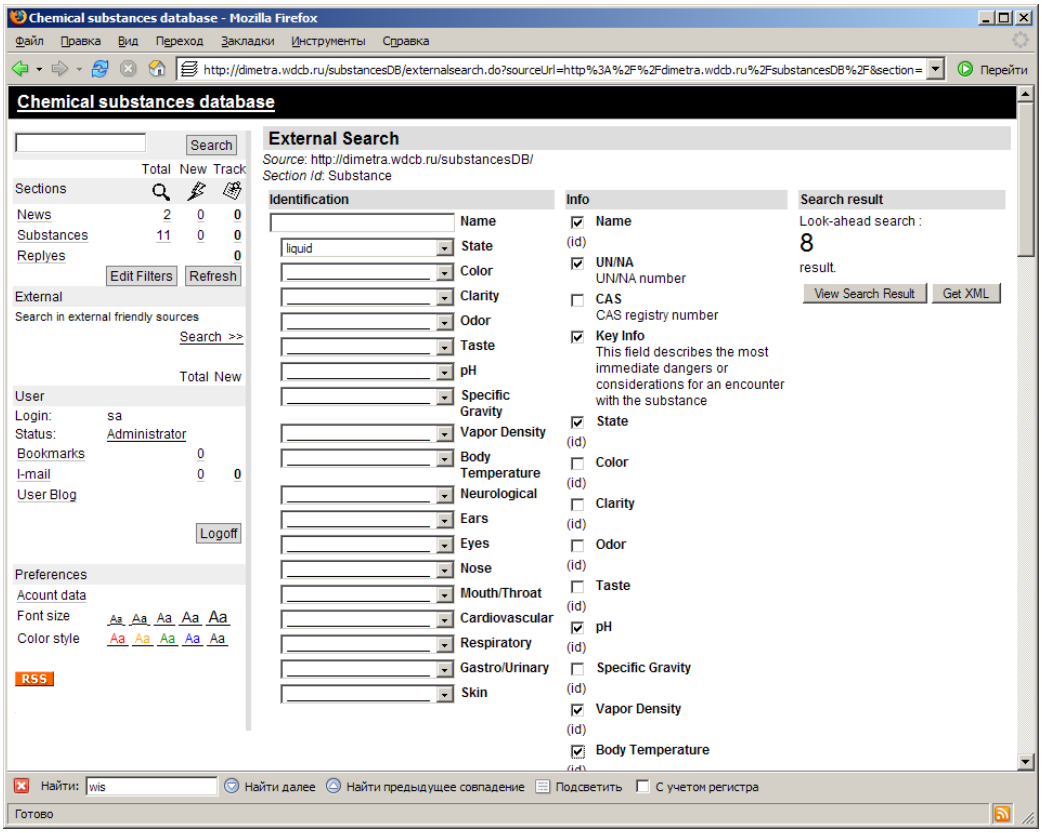

Figure 9: Toxic substance identification form. The External Search form is used to query the data base. In this example, the user has identified that the substance is in a liquid state and has requested that substances that have this property are returned along with details such as UN/NA number, Key Info, pH, and Vapor Density.

5 Software Files, Availability, and Requirements

The second public release of VxOware, which includes all of the features described in this paper, will be in January, 2010. VxOware is currently available in source code form via http://vxoware.org/. VxOware has been successfully installed on Windows XP and Linux operating systems and is only restricted to systems that support its external dependencies.

The code base of VxOware is primarily Java, and the following dependencies are included as a part of the VxOware distribution:

- eXist (metadata database)
- Struts (database interface)
- Lucene (search)
- BBCode (wiki syntax parsing)

VxOware's external dependencies are

- Tomcat (server)
- MySQL (password and account management)

6 Acknowledgments

Development was supported in part by NASA grant NNX07AB70G (VxO for S3C Data: The Virtual Radiation Belt Observatory) and NSF grant 0457577 (FDSS: Faculty Development in Space Weather Research: A Systems Perspective).