Earth Sci Inform (2010) 3:19–28
DOI 10.1007/s12145-010-0048-1

SOFTWARE ARTICLE

VxOware: software for managing virtual observatory metadata

Robert S. Weigel · Mikhail Zhizhin · Dmitry Mishin · Dmitry Kokovin · Eric Kihn · Jeremy Faden

Received: 10 October 2009 / Accepted: 6 April 2010 / Published online: 28 April 2010

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. While it is intended for use by a VxO specifically, it can also be used by any entity that manages structured metadata. VxOware has many features of a content management system and extensively uses the W3C recommendations for XML (Extensible Markup Language), XQuery (XML Query), and XSLT (Extensible Style Sheet Language Transformations). 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 features 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.

Keywords Metadata · XML · XQuery · Virtual observatory

Communicated by Thomas Narock

R. S. Weigel (✉)
George Mason University,
4400 University Drive,
Fairfax, VA 22030, USA
e-mail: rweigel@gmu.edu

M. Zhizhin · D. Mishin · D. Kokovin
Geophysical Center Russian Academy of Sciences,
Molodezhnaya Str., 3,
Moscow 119296, Russia

E. Kihn
National Geophysical Data Center,
E/GC 325 Broadway,
Boulder, CO 80305-3328, USA

J. Faden
Cottage Systems,
1141 E. Court St.,
Iowa City, IA 52240, USA

Introduction

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 single integrated 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”.

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 metadata. Although many content managements systems exist, VxOware is the only one developed specifically for scientific use that is primarily XML-based, with search provided by XQuery, interfaces created using XSL transforms on scientific or general metadata, and the ability to simply connect various instances into a federation. This built-in connectability is one of the key reasons why VxOware is especially suited for a virtual observatory: an important function of the virtual observatory is to make connections with other observatories. Two VxO's running VxOware can connect with very little effort.

The observatory section

The section is the most basic grouping of metadata records in VxOware. A virtual observatory using 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 (Really Simple Syndication) document model. When a user uses a web browser to view 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 1a is a screenshot that shows a previously stored metadata record that has been rendered in

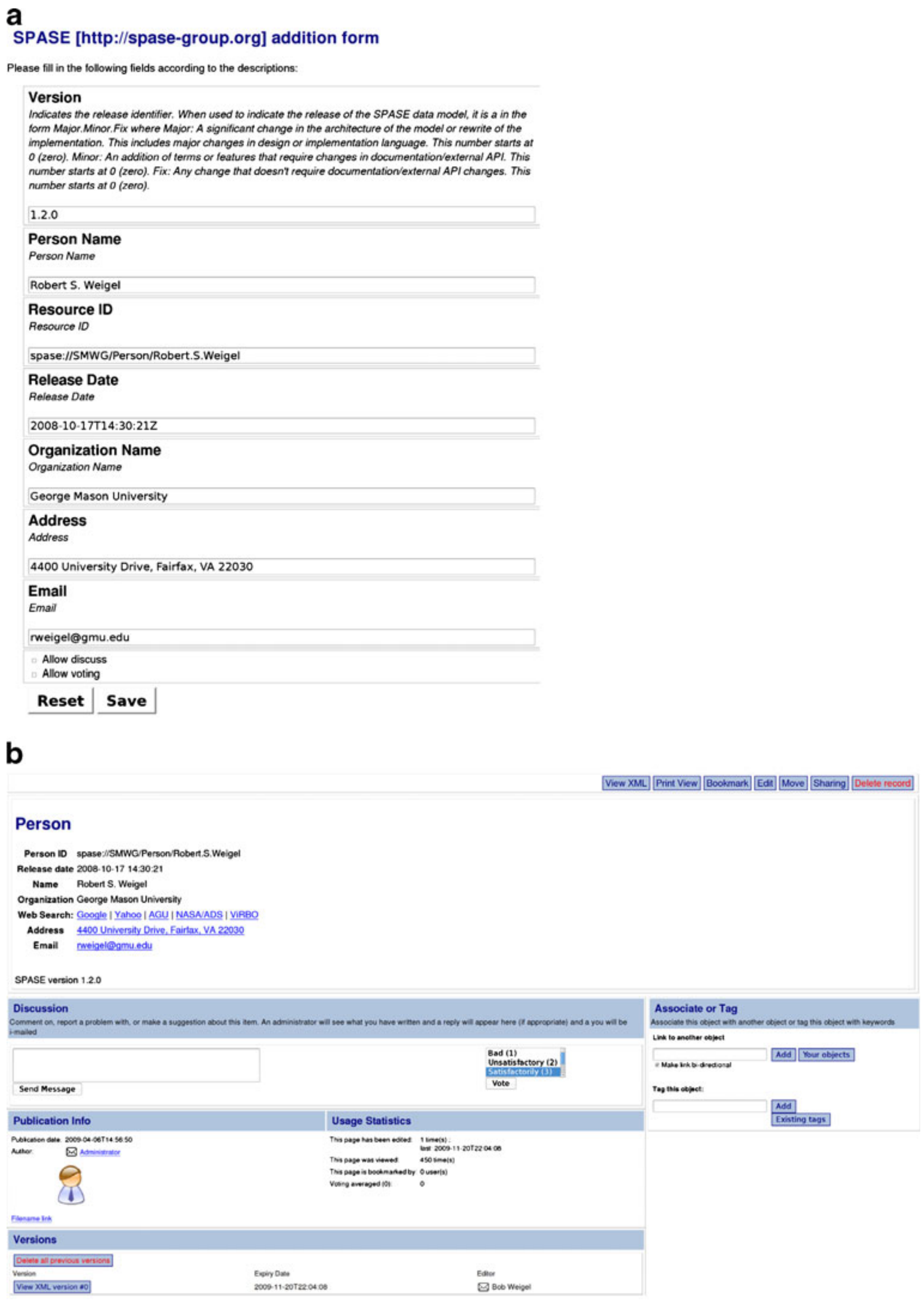


Fig. 1 **a** Screenshot of an editable form for a metadata record in a section named "Person". A raw XML record was transformed into this HTML form using the edit XSLT file that is associated with this section. **b** Screenshot showing the same metadata record rendered with the view XSLT file associated with this section

a

SPASE [http://spase-group.org] addition form

Please fill in the following fields according to the descriptions:

Version
Indicates the release identifier. When used to indicate the release of the SPASE data model, it is a in the form Major.Minor.Fix where Major: A significant change in the architecture of the model or rewrite of the implementation. This includes major changes in design or implementation language. This number starts at 0 (zero). Minor: An addition of terms or features that require changes in documentation/external API. This number starts at 0 (zero). Fix: Any change that doesn't require documentation/external API changes. This number starts at 0 (zero).

1.2.0

Person Name
Person Name

Robert S. Weigel

Resource ID
Resource ID

spase://SMWG/Person/Robert.S.Weigel

Release Date
Release Date

2008-10-17T14:30:21Z

Organization Name
Organization Name

George Mason University

Address
Address

4400 University Drive, Fairfax, VA 22030

Email
Email

rweigel@gmu.edu

☐ Allow discuss
☐ Allow voting

Reset Save

b

View XML | Print View | Bookmark | Edit | Move | Sharing | Delete record

Person

Person ID	spase://SMWG/Person/Robert.S.Weigel
Release date	2008-10-17 14:30:21
Name	Robert S. Weigel
Organization	George Mason University
Web Search:	Google | Yahoo | AGU | NASA/ADS | ViRBO
Address	4400 University Drive, Fairfax, VA 22030
Email	rweigel@gmu.edu

SPASE version 1.2.0

Discussion
Comment on, report a problem with, or make a suggestion about this item. An administrator will see what you have written and a reply will appear here (if appropriate) and a you will be i-mailed

Send Message

Bad (1)
Unsatisfactory (2)
[✓ Satisfactorily (3)]
Vote

Associate or Tag
Associate this object with another object or tag this object with keywords

Link to another object
Add | Your objects
Make link bi-directional

Tag this object:
Add | Existing tags

Publication Info

Publication date:	2009-04-06T14:56:50
Author:	Administrator

Filename link

Usage Statistics

This page has been edited:	1 time(s) ; last: 2009-11-20T22:04:08
This page was viewed:	450 time(s)
This page is bookmarked by	0 user(s)
Voting averaged (0):	0

Versions

Delete all previous versions

Version	Expiry Date	Editor
View XML version #0	2009-11-20T22:04:08	Bob Weigel

HTML using a render XSLT file that is associated with that section. Figure 1b shows the rendering of the same metadata record using an edit XSLT file. 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 does a search on the address in Google Maps.

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 that are included in the application installation package. These schemas include SPASE (Space Physics Archive Search and Extract) [http://spase-group.org/], FGDC (Federal Geographic Data Committee) [http://www.fgdc.gov/], WISER (Wireless Information System for Emergency Responders) [http://wiser.nlm.nih.gov], and OE (Ordering Extensions) [http://www.class.noaa.gov/].

Objects

VxOware can store 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 or created using a web interface 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 part 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; and (3) A binary object, which could be, for example, a PDF-formatted file or an image.

All 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 search and viewing,
- have an activity counter (number of reads, edits, and bookmarks), and
- have a change history.

Objects may be bookmarked, modified by a user with the appropriate rights, be opened for discussion and rating, and have "weak" or "strong" links to other objects. (Weak linking allows logical connections between different objects based on certain criteria, for example, all documentation can be tagged as "documentation". All objects in a section are "strong-linked".)

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

Community

An important design goal for a VxO is that it should 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 using 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-mail forwarded to an external email address;
- Each user can have a personal page (blog) for recording notes on such topics as data usage or general items of interest throughout the VxO. A user also has the ability to make their blog posts 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 and allow comments to be added by other users, thus creating an additional communication channel;
- VxOware 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 feedback on contributed objects such as 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; and
- VxOware tracks, and reports on the same page of the object, the number of users that have accessed, viewed, or commented on the object. This serves to allow object contributors to easily determine usage frequency.

Overview of architecture and paper

The main components of VxOware are shown in Fig. 2.

This release of VxOware is focused on the handling of metadata; although plug-ins are available for data resources,

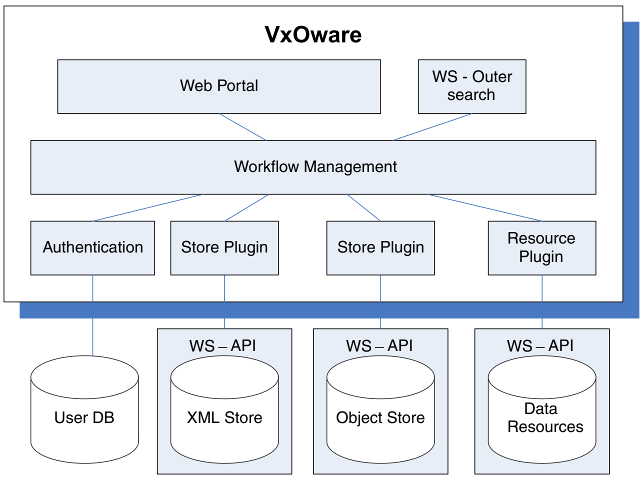


Fig. 2 Overview of VxOware components

they are not described in this paper except in part 4 where example implementations are discussed. Part 2 covers the available user features of VxOware, which include the Web Portal and Workflow Management software. Part 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. Part 3 covers the Outer Search, which is a RESTful web service that allows other applications to access or query a VxO's holdings.

User features

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 browsing through records in a section of interest and performing 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 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 accepting or declining the addition of objects that require moderation and viewing any comments left by users about problems with resources or questions about resources

Several basic components are required to support these workflows: a user authentication system, user profile records, an object storage system, and interaction services. These components are discussed below.

Authentication

VxOware uses a special database for user authentication, shown as "User DB" in Fig. 2. The User DB contains basic user profile information including username, password, user information, and user preferences. The User DB is isolated from the other databases.

User profile and user space

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 these objects, and VxO records 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, and 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; and
- 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.

Search

Search is an integral part of VxOware. VxOware allows for 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 in VxOware: free-text, context, and external.

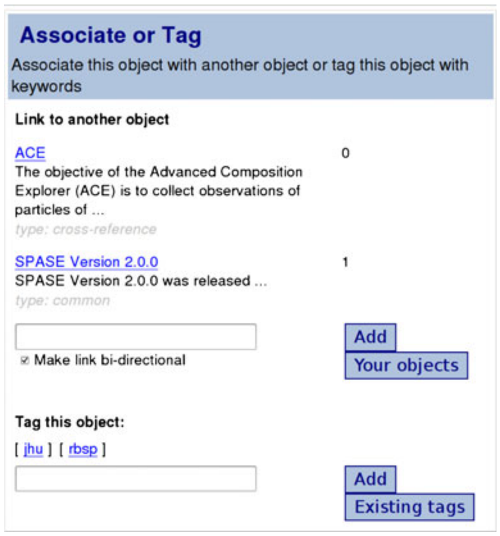
1. *A free-text search* is used for full-text search in all non-binary objects in the VxO.
2. *A context/attribute search* is done taking into consideration the properties of each type of record and is aimed at the contents of the record. The search space can be limited to one or more of the elements in the metadata schema.
3. *An external search* may be made on 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 for their users to the "News" section of an external VxO in their federation, the external VxO only needs to specify that their News section is 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 record elements that are available for search. One advantage of this approach is a VxO manager may select sections that can be searched by all users in the federation along with sections that cannot be searched unless the user has visited their specific observatory (to prevent outsiders from searching the contents of sections that are under development, for example).

Linking and tagging

Every object in the VxO can be tagged or linked to another object in the observatory. A screenshot of the tag and link interface is shown in Fig. 3. When the user enters the ID of another VxO object in the link field, the ID 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 forum file associated with the metadata record.

Update monitoring

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


Associate or Tag
Associate this object with another object or tag this object with keywords

Link to another object

ACE 0
The objective of the Advanced Composition Explorer (ACE) is to collect observations of particles of ...
type: cross-reference

SPASE Version 2.0.0 1
SPASE Version 2.0.0 was released ...
type: common

☑ Make link bi-directional

Tag this object:
[jhu] [rbsp]

Fig. 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)

There are three methods by which a registered user may track or monitor updates or changes to the VxO. First, they may define personal settings that filter all incoming metadata records by sections. Each time the user logs in to 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 Fig. 4 has a column that shows the number of items that have been added to each section since the user last logged in.

Administration

There is a special type of VxO user called an administrator. They are responsible for the maintenance of the observatory. Administrators may delete objects, change global and user-level settings, and edit all VxO objects without restriction.

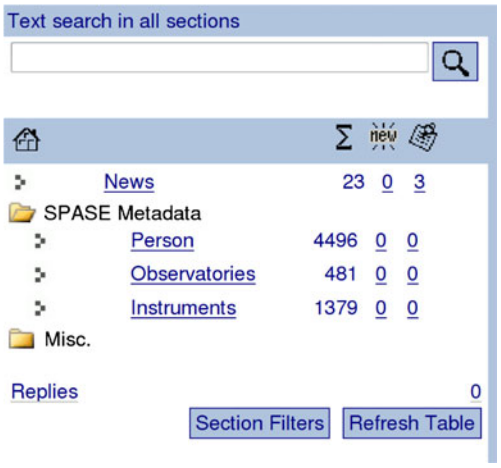
Editable content and wiki features

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 and XML to enable management of metadata for modern web applications and has features comparable to YouTube [http://youtube.com], Reddit [http://reddit.com], expert systems, and cloud computing services. A number of existing applications that use VxOware are described in part 4.

The wiki and discussion features are intended to be an improvement over the typical approach used by data server administrators for 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 to 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 the user's file reader. With the exchange between the data server administrator and the data user viewable from a convenient location, another 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 publicly available and accessible 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; and
- Based on other exchanges, a user will have a good idea of what to expect in terms of responsiveness from the data server administrator.

Another unique feature of 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 web page to see if the software has been updated. This feature is not intended to be a replacement for the approach in which a user receives updates by subscribing to a project's email notification list, but is rather intended to be a simplified alternative of this approach. 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.


Text search in all sections

	Σ	new	
News	23	0	3
SPASE Metadata			
Person	4496	0	0
Observatories	481	0	0
Instruments	1379	0	0
Misc.			
Replies			0

Fig. 4 Left-panel view for a 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

Storage, API, and services

Storage

The types of storage in VxOware are shown in Fig. 2 and details about them are given below.

- User DB: This database contains: usernames, emails, last-logged-in date, user type (admin or user).
- XML DB: This is an eXist database that contains a 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.
- Data Resource: This can be any type of database that contains scientific data. For each database, a plugin is required.

Upload services

Users may contribute objects to their VxO 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 many 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 observatory 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.

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. In response to a request for 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 that will result in a list of matching objects. An important feature of the external search web service is its dynamic result format. The user may select the meta-record fields that appear in the query result. These properties of the web service make it suitable for case-based reasoning applications.

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.

Virtual Radiation Belt Observatory (ViRBO)

The Virtual Radiation Belt Observatory [http://virbo.org] is an instance of one of the 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 metadata resources; its homepage is shown in Figure 5.

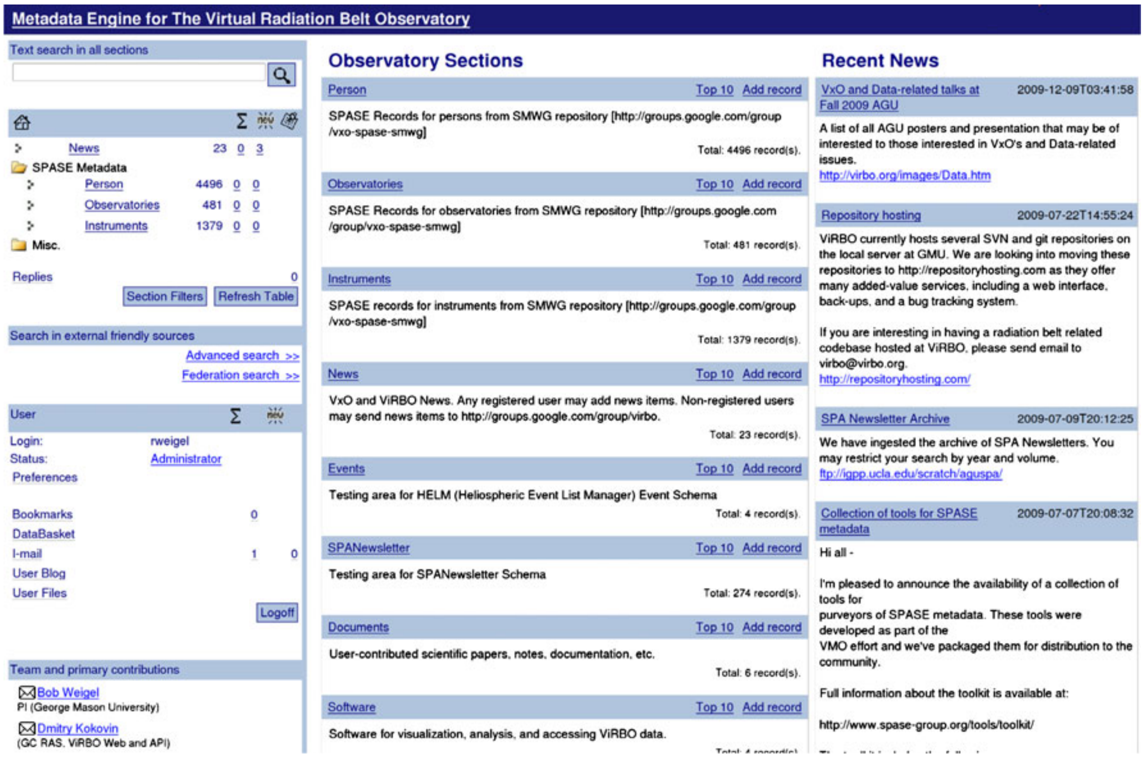
Comprehensive Large-Array Data Stewardship System (CLASS)

The CLASS application provides an interface to NOAA's large satellite data archive. 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. VxOware is used in CLASS to provide metadata storage and search and to create order forms based on OE records. The interaction of CLASS with other VxOs 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 direct access to the meta-record storage database is available. The CLASS homepage is shown in Fig 6.

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



Metadata Engine for The Virtual Radiation Belt Observatory

Text search in all sections

News	23	0	3
SPASE Metadata			
Person	4496	0	0
Observatories	481	0	0
Instruments	1379	0	0
Misc.			
Replies	0		

Section Filters | Refresh Table

Search in external friendly sources

Advanced search >>

Federation search >>

User

Login: rweigel

Status: Administrator

Preferences

Bookmarks 0

DataBasket

I-mail 1 0

User Blog

User Files

Logoff

Team and primary contributions

Bob Weigel
PI (George Mason University)

Dmitry Kokovin
(GC RAS, ViRBO Web and API)

Observatory Sections

Person — Top 10 Add record

SPASE Records for persons from SMWG repository [http://groups.google.com/group/vxo-spase-smwg]

Total: 4496 record(s).

Observatories — Top 10 Add record

SPASE Records for observatories from SMWG repository [http://groups.google.com/group/vxo-spase-smwg]

Total: 481 record(s).

Instruments — Top 10 Add record

SPASE records for instruments from SMWG repository [http://groups.google.com/group/vxo-spase-smwg]

Total: 1379 record(s).

News — Top 10 Add record

VxO and ViRBO News. Any registered user may add news items. Non-registered users may send news items to http://groups.google.com/group/virbo.

Total: 23 record(s).

Events — Top 10 Add record

Testing area for HELM (Heliospheric Event List Manager) Event Schema

Total: 4 record(s).

SPANewsletter — Top 10 Add record

Testing area for SPANewsletter Schema

Total: 274 record(s).

Documents — Top 10 Add record

User-contributed scientific papers, notes, documentation, etc.

Total: 6 record(s).

Software — Top 10 Add record

Software for visualization, analysis, and accessing ViRBO data.

Recent News

VxO and Data-related talks at Fall 2009 AGU — 2009-12-09T03:41:58

A list of all AGU posters and presentation that may be of interest to those interested in VxO's and Data-related issues.
http://virbo.org/images/Data.htm

Repository hosting — 2009-07-22T14:55:24

ViRBO currently hosts several SVN and git repositories on the local server at GMU. We are looking into moving these repositories to http://repositoryhosting.com as they offer many added-value services, including a web interface, back-ups, and a bug tracking system.

If you are interesting in having a radiation belt related codebase hosted at ViRBO, please send email to virbo@virbo.org.
http://repositoryhosting.com/

SPA Newsletter Archive — 2009-07-09T20:12:25

We have ingested the archive of SPA Newsletters. You may restrict your search by year and volume.
ftp://igpp.ucla.edu/scratch/aguspa/

Collection of tools for SPASE metadata — 2009-07-07T20:08:32

Hi all -

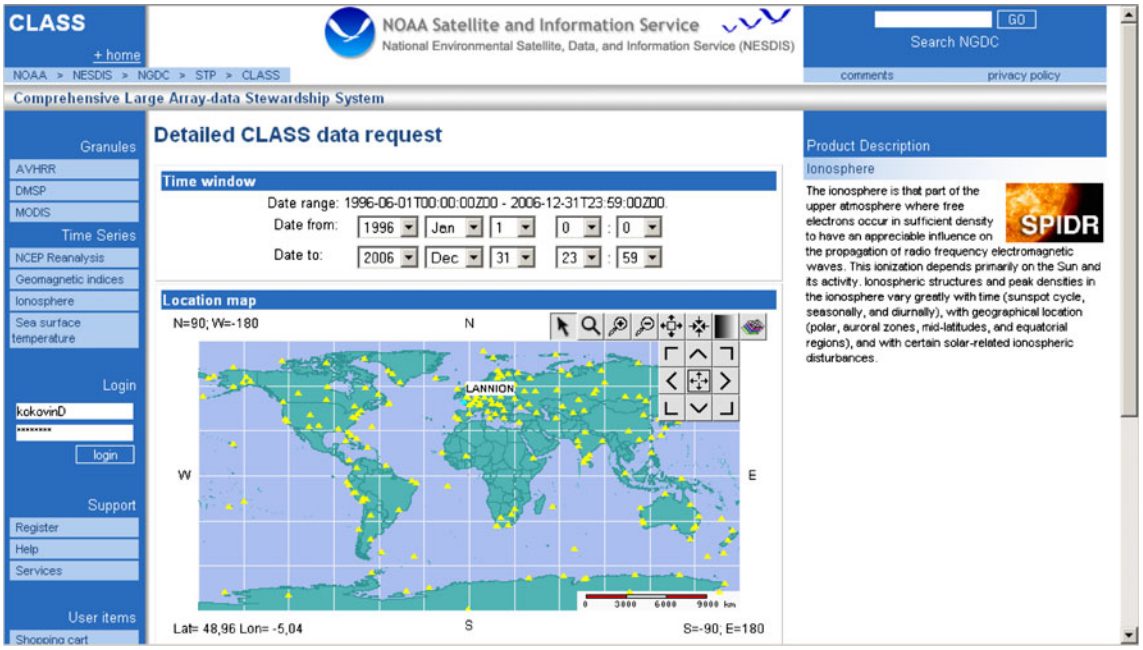
I'm pleased to announce the availability of a collection of tools for
purveyors of SPASE metadata. These tools were developed as part of the
VMO effort and we've packaged them for distribution to the community.

Full information about the toolkit is available at:

http://www.spase-group.org/tools/toolkit/

Fig. 5 Homepage of the ViRBO metadata engine

historical space physics (heliophysics) 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 (Global Observation Information Network). It has since evolved into a joint development effort with the Russian Academy of Sciences and is now more than 10 years old with more than 10,000 users. VxOware is used to manage its large metadata holdings.



Detailed CLASS data request

Time window

Date range: 1996-06-01T00:00:00Z00 - 2006-12-31T23:59:00Z00.

Date from: 1996 Jan 1 0 : 0

Date to: 2006 Dec 31 23 : 59

Location map

N=90; W=-180

Lat= 48,96 Lon= -5,04

S=-90; E=180

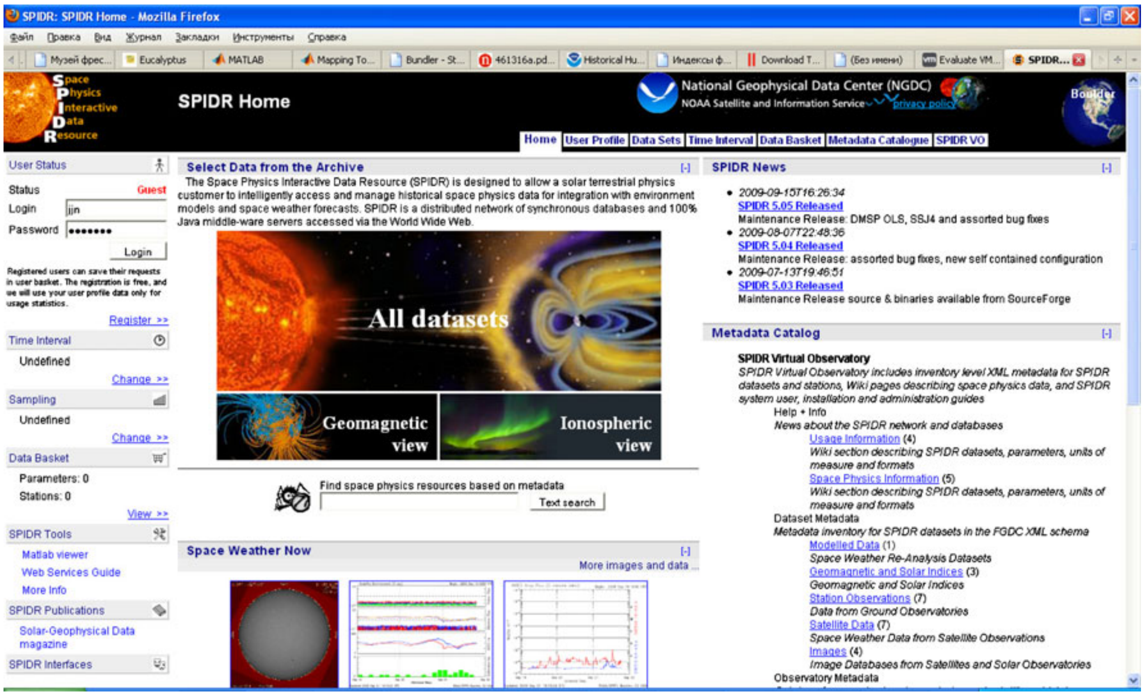
kokovinD

Product Description

Ionosphere

The ionosphere is that part of the upper atmosphere where free electrons occur in sufficient density to have an appreciable influence on the propagation of radio frequency electromagnetic waves. This ionization depends primarily on the Sun and its activity. Ionospheric structures and peak densities in the ionosphere vary greatly with time (sunspot cycle, seasonally, and diurnally), with geographical location (polar, auroral zones, mid-latitudes, and equatorial regions), and with certain solar-related ionospheric disturbances.

Fig. 6 Homepage of the CLASS system [http://class.ngdc.noaa.gov/]



Registered users can save their requests in user basket. The registration is free, and we will use your user profile data only for usage statistics.

The Space Physics Interactive Data Resource (SPIDR) is designed to allow a solar terrestrial physics customer to intelligently access and manage historical space physics data for integration with environment models and space weather forecasts. SPIDR is a distributed network of synchronous databases and 100% Java middle-ware servers accessed via the World Wide Web.

- *2009-09-15T16:26:34*
  SPIDR 5.05 Released
  Maintenance Release: DMSP OLS, SSJ4 and assorted bug fixes
- *2009-08-07T22:48:36*
  SPIDR 5.04 Released
  Maintenance Release: assorted bug fixes, new self contained configuration
- *2009-07-13T19:46:51*
  SPIDR 5.03 Released
  Maintenance Release source & binaries available from SourceForge

SPIDR Virtual Observatory
SPIDR Virtual Observatory includes inventory level XML metadata for SPIDR datasets and stations, Wiki pages describing space physics data, and SPIDR system user, installation and administration guides
Help + Info
News about the SPIDR network and databases
Usage Information (4)
Wiki section describing SPIDR datasets, parameters, units of measure and formats
Space Physics Information (5)
Wiki section describing SPIDR datasets, parameters, units of measure and formats
Dataset Metadata
Metadata inventory for SPIDR datasets in the FGDC XML schema
Modelled Data (1)
Space Weather Re-Analysis Datasets
Geomagnetic and Solar Indices (3)
Geomagnetic and Solar Indices
Station Observations (7)
Data from Ground Observatories
Satellite Data (7)
Space Weather Data from Satellite Observations
Images (4)
Image Databases from Satellites and Solar Observatories
Observatory Metadata

Fig. 7 SPIDR homepage

SPIDR also uses VxOware for its metadata search web service that provides access to the XML-formatted records describing the data sets and observatories and includes links to the metadata from SPIDR web-portal maps and data export pages. Data managers can use the VxOware administrator interface to edit metadata records related to their data product and services. SPIDR's homepage is shown in Fig 7.

Expert system for Recognition of Toxic Events (RTE)

The expert system for Recognition of Toxic Events (RTE) was developed as part of the European Union R&D project named 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 involving explosive, flammable, and toxic substances and methods of overcoming the consequences of accidents with them. Its XML schema, shown in Fig. 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.

The RTE expert system allows users to respond to an emergency involving a toxic substance using a search that can be constrained on physical properties of the substance and/or the medical symptoms associated with interaction with the substance. VxOware middleware is used by RTE as a web application layer above the XML knowledge base. It allows for interoperability with other applications by providing self-describing metadata (that indicate capabilities) and a 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 a synonyms dictionary, for example) and returns a list of candidate substances, which may only approximately satisfy the

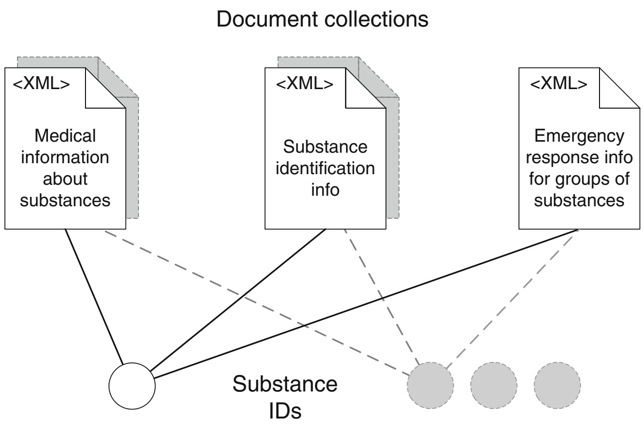
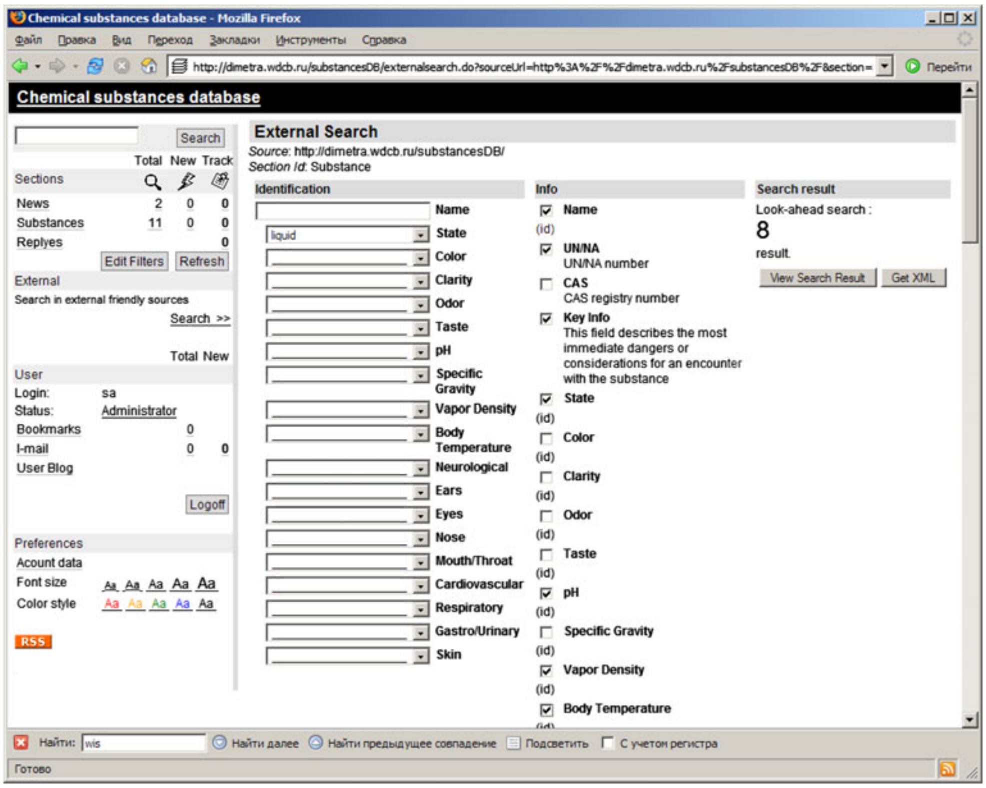

Document collections
<XML>
Medical information about substances
<XML>
Substance identification info
<XML>
Emergency response info for groups of substances
Substance IDs

Fig. 8 RTE knowledge base structure



Chemical substances database - Mozilla Firefox

http://dimetra.wdcb.ru/substancesDB/externalsearch.do?sourceUrl=http%3A%2F%2Fdimetra.wdcb.ru%2FsubstancesDB%2F&section=

Chemical substances database

	Total	New	Track
Sections			
News	2	0	0
Substances	11	0	0
Replyes			0

External
Search in external friendly sources

User
Login: sa
Status: Administrator
Bookmarks 0
I-mail 0 0

External Search
Source: http://dimetra.wdcb.ru/substancesDB/
Section Id: Substance

Identification
State: liquid

Info
[✓ Name] (id)
[✓ UN/NA] UN/NA number
CAS — CAS registry number
[✓ Key Info] This field describes the most immediate dangers or considerations for an encounter with the substance
[✓ State] (id)
Color (id)
Clarity (id)
Odor (id)
Taste (id)
[✓ pH] (id)
Specific Gravity (id)
[✓ Vapor Density] (id)
[✓ Body Temperature]

Search result
Look-ahead search :
8
result.

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

search criteria based on a partial description that includes such properties as color, physical state, and medical symptom. The search web service allows the user to view an AJAX-updated list of candidate toxic substances as shown in Fig. 9.

Summary

VxOware is a software package that provides a highly integrated set of metadata management and communication functions that are needed by a virtual observatory. VxOware can also be seen as a content managements system for XML documents that extensively uses the W3C recommendations for XML (Extensible Markup Language), XQuery (XML Query), and XSLT (Extensible Style Sheet Language Transformations). Instances of VxOware may be installed on various servers and connected to form a federation of VxOs that communicate via web services.

Software files, availability, and requirements

The second public release of VxOware, which includes all of the features described in this paper, will at the time of publication of this paper. 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-db (metadata database), Struts (database interface), Lucene (for search), and BBCode (for wiki syntax parsing). VxOware's external dependencies are Tomcat and MySQL (for password and account management).

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).