



Virtual Solar Observatory

Virtual Solar Observatory Applications

F. I. Suárez-Solá¹, R.S. Bogart², A.R. Davey³, G. Dimitoglou⁴, J.B. Gurman⁴, F. Hill¹, J. Hourclé⁴, P.C. Martens⁵, K.Q. Tian², S. Wampler¹, K. Yoshimura⁵

¹National Solar Observatory, Tucson AZ – ²Stanford University, Stanford CA – ³Southwest Research Institute, Boulder CO – ⁴NASA/GSFC, Greenbelt MD – ⁵Montana State University, Bozeman MT

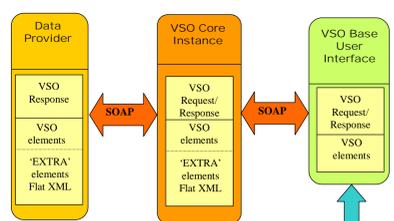
ABSTRACT:

This poster shows two different uses of the Virtual Solar Observatory framework (currently in development), which are both possible thanks to the virtual, lightweight and distributed design. The first application takes advantage of the VSO user interface to generate another one similar in look and feel to the VSO UI HTML but with extended functionality. This would be ideal for a low-budget front-end implementation of an in-house querying tool. The second one is a dedicated Virtual Solar Observatory user interface (VSO “shopping” Cart) to enable solar scientists to track and log their VSO queries and results sets. These applications are in the early stages of development. We expect to have part of the functionality in place in December 2004.

Customizing the Virtual Solar Observatory

Using the VSO as a cost effective way to power a specific content WEB site

- Light weight, distributed back end
- Minimum coding required
- Quick implementation
- Automatically includes improvements as VSO development continues

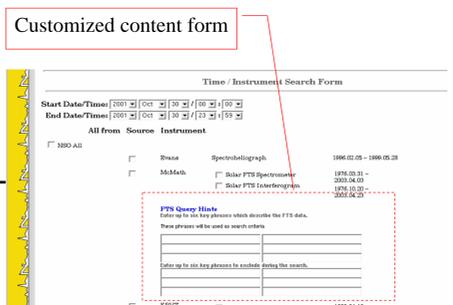
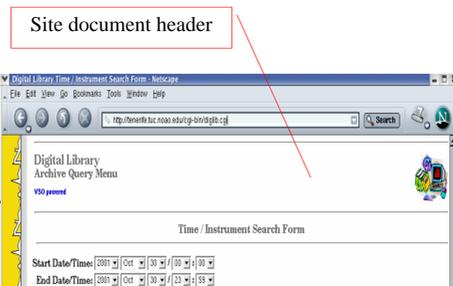


The VSO architecture allows non-VSO information to flow to and from the provider. The core acts as a relay for the extra information.

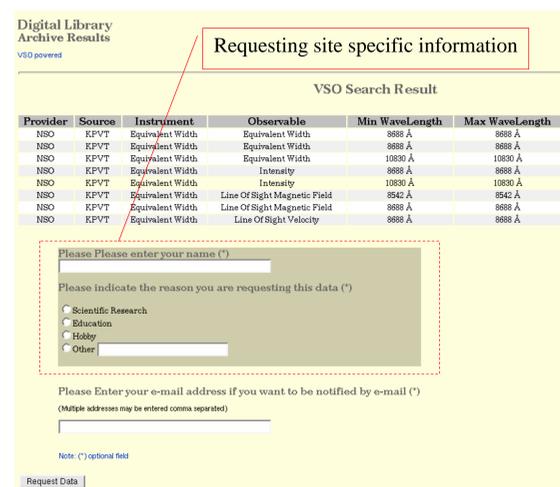
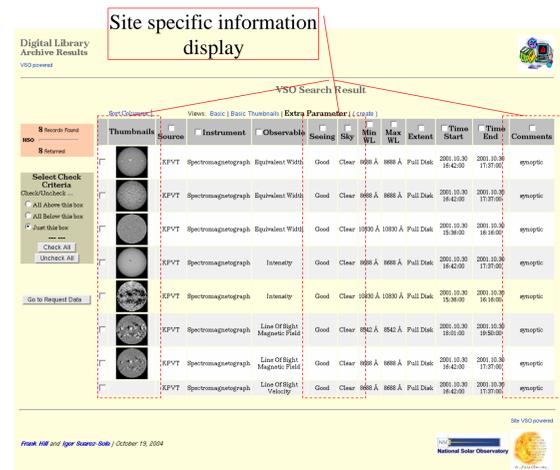
This permeability takes the constraint out of the user interface, allowing site specific customization



Query document for the Digital Library powered by a VSO engine



- A code generator builds the WEB site perl CGIs:
- XML schema document
 - XML document holding the configuration parameter
 - Perl modules or separate code insets
 - VSO user interface base classes and templates

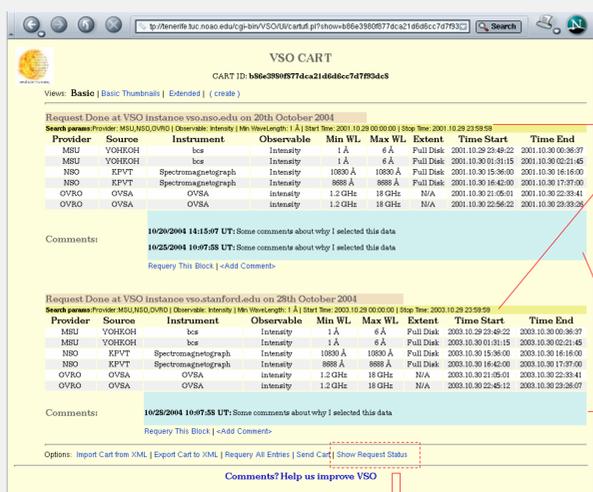


Viewing the VSO Cart

The VSO cart is designed to provide the solar scientist with functionality to track, log and facilitate the handling of their VSO queries and results sets.

The cart information can be exported to the user's desktop or pass around for others to use in any VSO instance running on a local machine or a centralized server.

Since the cart id is the only identifier the contents remain private and yet accessible to the user. (given of course that the user hasn't forgotten the cart id in first place)



Two requests done at different times and with different VSO instances.

In this case the user decided to keep the second request under the same cart id as it was related to the first request

General comments



Breakdown of request per provider

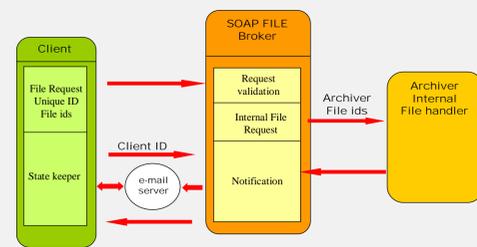
Hyperlinks to previous requests under the same cart id

Overcoming hurdles:

One of the biggest hurdles the cart faces in the use of automatic file request programming, is the lack of standardized interfaces for asynchronous file requests. I.e. The provider will handle the request and notify the user once it is ready or alternatively the client will check in a timely manner for its request status.

Currently for non URL-FILE type request, (e.g. STAGING -- asynchronous requests -- see poster "Data Transfer Negotiation Within the Virtual Solar Observatory" J. Hourclé et al), the cart is forced to use an e-mail system as intermediary to centralized its file request responses. This is because at present the only way providers can notify the client is via e-mail and there is no functionality in place for the clients to tag (e.g. with an user id) the requests.

At NSO and MSU we are developing a file SOAP request server that allows client status querying and/or client notification via SOAP, mail etc.



The file ids are unique identifiers known to the archiver.

