A Web Services Interface to Advanced Composition Explorer (ACE) Data

Andrew Davis, Glenn Hamell

Spring AGU, May 2005
ACE Level 2 (Verified) Data

| Interplanetary Magnetic Field Parameters: MAG | MAG Data | Documentation |
| Solar Wind Parameters: SWEPAM | SWEPAM Data | Documentation |
| Solar Wind Temperatures, Speeds, and Species Ratios: SWICS | SWICS Data | Documentation |
| Solar Suprathermal and Energetic Particle Intensities: ULEIS | ULEIS Data | Documentation |
| Solar Energetic Particle Intensities: EPAM | EPAM Data | Documentation |
| Solar Energetic Particle Intensities: SEPICA | SEPICA Data | Documentation |
| SEP, GCR, and ACR Intensities: SIS | SIS Data | Documentation |
| Galactic Cosmic Ray Intensities: CRIS | CRIS Data | Documentation |
| **NEW** Merged IMF and Solar Wind 64-second Averages | MAG/SWEPAM Data | Documentation |
| **NEW** Merged IMF, Solar Wind, and Energetic Particle Hourly Averages | Multi-instrument Data | Documentation |
Define the Problem…

We want to open up ACE Science Center services and data to remote computers and services. However…

• Access to ACE data is currently via a web-based forms interface, designed for human interaction.

• The interface is non-standard, undocumented.

• We do not have resources to build a new interface from scratch – we want to integrate our existing (legacy) interface.

• Current descriptions of ACE data products are also non-standard.
Proposed Solution…

• Describe ACE data products in XML, using the Space Physics Archive Search and Extract (SPASE) Project Data Model.

• Use Simple Object Access Protocol (SOAP) as the communications protocol to handle Discovery queries and Data queries.
  ➢ Build a SOAP server to handle queries

• Build a middleware that mediates between our legacy interface and the new service.

• Use HTTP protocol for final transfer of data.
Service Architecture

ACE Science Center

XML Product Descriptions
XML Parser
Discovery Query Handler
Data Query Handler
SOAP Server
Data URI
Query
Reply
Data
Data URI

ACE Data (HDF Files)
Legacy API
Middleware
Web Server

New
Existing
SOAP Server

• Implemented in Perl, using SOAP::Lite and XML::Simple modules
• Currently supports the following queries:
  ➢ **Discovery** – Server replies with summary information on each data product available from the ASC
  ➢ **Get_Details(Product_ID)** – Server replies with detailed info on each data item for a given data product
  ➢ **Get_Data(Params)** – Client requests a subset of data; Server replies with a HTTP GET URI to the data subset.

• Easily adaptable to emerging standards… 😊
• Example client application available on request (written in Perl)
Middleware

- Mediates between legacy API and new service.
- Accepts simple HTTP GET requests for data subsets.
- Translates requests into the more obtuse HTTP POST requests required by the legacy API.
- Streams ASCII data from the legacy API to the client as HTTP file download (bent pipe).
- Implemented as a Perl CGI script.
• Could simplify by having middleware interface directly to Legacy API.
<Product><Header>
<Product_ID>ASC_ACE_ACE_MAG_ASCII_3600s_v1</Product_ID>
<Product_Name>ACE MAG Hourly Level 2 Data</Product_Name>
<Description>Hourly Averaged Interplanetary Magnetic Field Data from ACE/MAG</Description>
<Repository_Name>ACE Science Center (ASC)</Repository_Name>
</Header>
<Project>
  <Observatory>ACE</Observatory>
  <Instrument_Name>MAG</Instrument_Name>
  <Instrument_Type>Magnetometer</Instrument_Type>
</Project>
<Contact>
  <Name><First>Chuck</First><Middle>W</Middle><Last>Smith</Last></Name>
  <Institution>University of New Hampshire</Institution>
  <Role>Instrument Scientist</Role>
  <E-mail>Charles.Smith@unh.edu</E-mail>
</Contact>
<Product_Type>Numerical_Data</Product_Type>
  <Format>ASCII</Format>
  <File_Encoding>UNIX (ASCII)</File_Encoding>
</Numerical_Data>
<Availability>Online</Availability>
<Access_Rights>Open</Access_Rights>
<Measurement_Type>Magnetic Field</Measurement_Type>
<Processing_Level>Calibrated</Processing_Level>
<Instrument_Region>L1</Instrument_Region>
<Observed_Region>L1</Observed_Region>
<Resolution>1hr</Resolution>
… and so on and so on..
Detailed Time-Series Descriptions…

<Time_Series>
  <Time_Series_Description>
  GSM X component of hour-averaged magnetic field
  </Time_Series_Description>
  <Physical_Quantity>MAG_FIELD</Physical_Quantity>
  <Entity>FIELD</Entity>
  <Qualifier>COMPONENT</Qualifier>
  <Qualifier>X</Qualifier>
  <Unit>NT</Unit>
  <Coordinate_System>GSM</Coordinate_System>
  <Query_Code>Bgsm_x</Query_Code>
</Time_Series>

One of these for each data item in a product…
Current Status

• The service is up and running - beginning to connect to VHO, VSPO, etc.

• Example client application is available on request (Perl script) – beta-testers needed!

• Interested in learning about standards for formulating space-physics discovery and data queries using the SOAP protocol.
Lessons Learned?

• Current toolkits (e.g. Perl’s SOAP::Lite) make implementation of a SOAP service relatively painless – not too much prior expertise needed.

• Most important step – getting product metadata into a structured format – not necessarily XML!

• Not a task one could tackle on a single slow afternoon…
#!/usr/local/bin/perl -w

use SOAP::Transport::HTTP;

use Discover;
use GetDetails;
use GetData;

SOAP::Transport::HTTP::CGI
    ->dispatch_to('Discover','GetDetails','GetData')
    ->handle;
Example Data Query

```php
$som = $soap->getdata(
    SOAP::Data->name('DataQuery' => SOAP::Data->value(
        SOAP::Data->name("instrument_name" => "MAG"),
        SOAP::Data->name("resolution" => "1hr"),
        SOAP::Data->name("data1" => "Brtn_r"),
        SOAP::Data->name("data2" => "Brtn_t"),
        SOAP::Data->name("data3" => "Brtn_n"),
        SOAP::Data->name("start" => "2002 1"),
        SOAP::Data->name("end" => "2002 2")
    )))
```