Interoperable SED Access and Analysis

Project Development Plan

Project ID:	1.3
Revision:	1.0
Project Definition	http://dev.usvao.org/vao/wiki/Projects2010/SEDAccessAnalysisPDD
Project Lead:	Janet D Evans
QA&T Engineer	Randy Thompson
Document Status:	**Development-Ready**
Reviewed	April 2011

1 Summary of Products

1.1 New Product Releases Planned

Product Name	New/Exis.[1]	Type[2]	Summary of functionality
NED Service*	Existing	W	Access and deliver SEDs from NED
Iris*	New	D	SED access and analysis tool
Specview	Existing	D/W	SED visualization & editing
Sherpa	Exisiting	D	SED fitting
Sherpa-SAMP	New	L	Specview to Sherpa SAMP interface
SEDImporter*	New	D	Import data and transform to a compliant VOtable
SEDLib	New	L	Read, Write, Manipulate SED data file

[1]**New** or **Existing** indicates whether this is a new product or an existing one being extended.
[2]The type of product; one or more of: **L**=Library, **D**=Desktop tool, **W**=Programmatic Web Service, **P**=Browser-based web/portal application, **T**=service toolkit
*The project's integrating product. The product(s) that brings together all of the components that deliver the capabilities enumerated in the PDP.

1.2 Third Party Products

Third-party products/packages expected to be used in this project (this may change during development):

Product Name	Brief Description/Purpose	Origin (URL, if exists)
Specview	Base GUI that manages interactions; also provides data access & visualization	STScI
Sherpa	SED Modeling and Fitting	SAO
STIL library	Starlink Tables Infrastructure Library	http://www.star.bris.ac.uk/~mbt/stil/

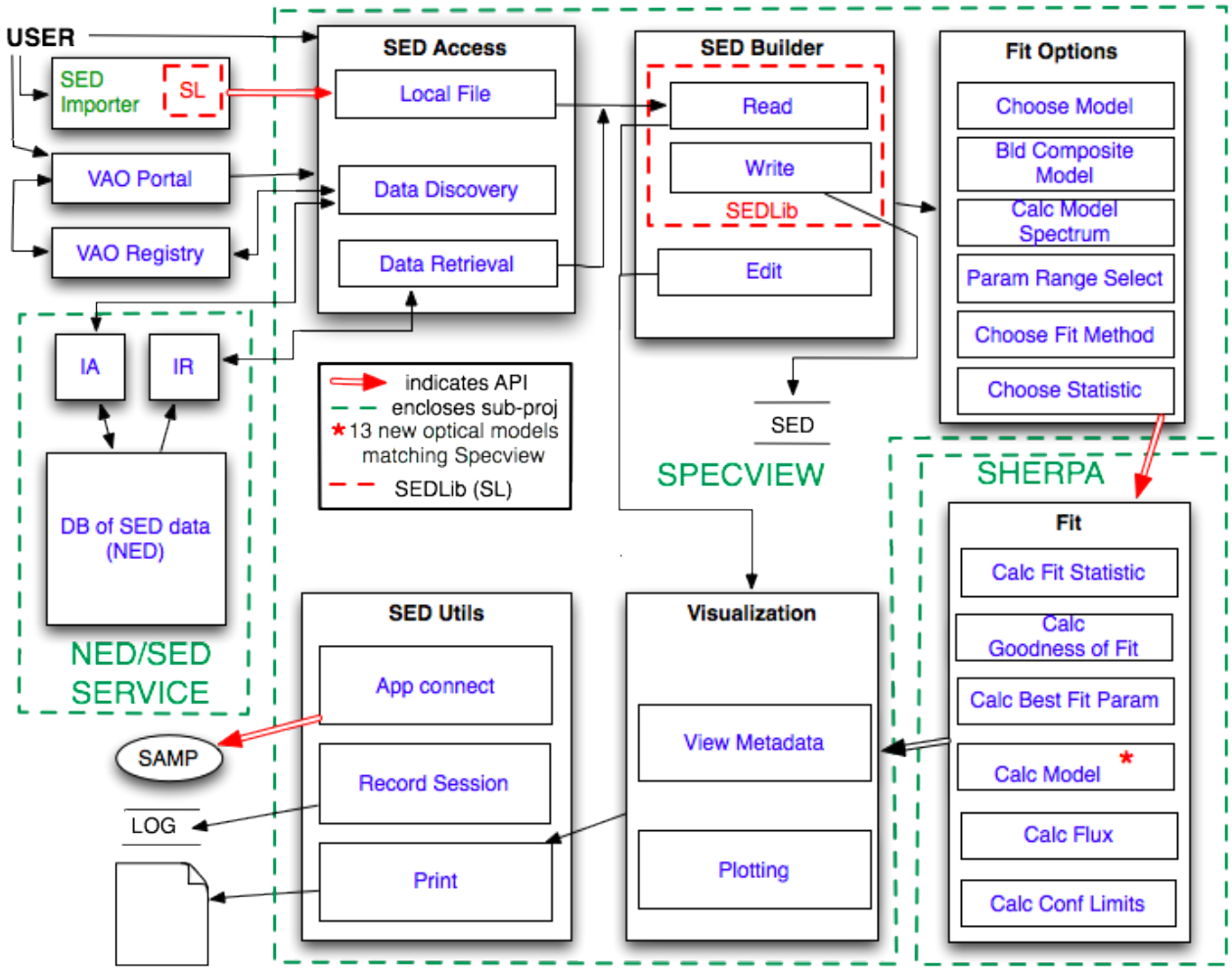
2 Summary of Overall Design

How do you expect users to encounter these products (particularly the integrating product) to begin using them?

- via the VAO registry (NED Service)
- via an index of available tools for VAO users (Iris)
- via an index of available tools for developers (SEDLib)

2.1 Overview

Spectral Energy Distribution (SED) access and analysis consists of 2 major components: the NED/SED Service and Iris (the name for the combined Specview & Sherpa package). The NED/SED Service is a standalone Web-based service that provides SED Information Discovery and Retrieval capabilities. Specview provides the base GUI for reading, writing, editing, and displaying SEDs, as well as defining a model and setting initial parameter values. Specview packages and sends data to Sherpa whenever a fit or confidence limits are requested. Sherpa provides a library of models, fit statistics, optimization methods, and a function for calculating confidence limits on best-fit parameter values.



Communication between Specview and Sherpa is managed by a SAMP hub. The goal is to seamlessly combine the power of Specview's GUI and data manipulation functions with Sherpa's robust modeling and fitting functions, and also provide easy access to NED's extensive database of extragalactic SEDs. The diagram below shows the major components of the NED/SED service and of Iris, and highlights the flow of information through the application.

Supporting components include: the SED library (SEDLib), written in Java, providing the ability to read, write, and manipulate information contained in a SED file; an interface written in Python, supporting SAMP communication between Specview and Sherpa; Specview visualization capabilities, to plot data and display meta-data associated with the points in the plot; a SEDImporter module, written in Java, that reads in non-compliant SEDs, and uses SEDLib to output compliant SEDs for input to Iris.

2.2 NED Service

The IPAC SED data interrogation and access service provides for three types of query. All queries are HTTP v1.1 GET requests and all responses are HTTP v1.1 RESPONSEs.

- NED SED Info Discovery – is based upon IVOA Simple Spectral Access Protocol Version 1.04. The request uses two parameters: POS in which is delivered a string containing RA[decimal degrees, ICRF/FK5J2000] and DEC[decimal degrees, ICRF/FK5J2000]; and a SIZE parameter which specifies the cone angle (degrees) to establish a search volume that is a Circular Spherical Cone; and returns an information object containing a table (list) of NED Object Identifiers (Names), a count of photometric SED measurements available for this Object, and the URI for the NED-SED Data Retrieval request (accessSED).
- NED SED Info Availability – based upon/extension of IVOA Simple Spectral Access Protocol Version 1.04. The request uses one parameter, ”object name” and returns an information object containing the relavant NED Object Identifier (Name), a flag indicating the availability (true/non-zero), or unavailability (false/zero) of photometric SED information from the archive, and a URI for the NED-SED-Data-Retrieval query for archive data for the specified object.
- NED SED Data Retrieval – IVOA Spectral Data Model Version 1.03. The request uses a NED qualified object Identifier, and returns a data object in XML format.

2.3 Specview and Sherpa interaction

Communication between Specview and Sherpa is based on the SAMP VO communication protocol. The interface to Sherpa replaces the existing fitting capabilities in Specview.

Specview is the Iris front-end. Specview supports all user interactions with its GUI and Sherpa supports fitting tasks as a background process. Specview launches a SAMP hub (if one is not already running) and Sherpa as background processes at start-up. Similarly, upon exiting, Specview shuts down the Sherpa process and the SAMP hub (if necessary). When the user is ready to fit an SED, Specview contacts Sherpa with a SAMP call and waits for a response. Sherpa, running as a background process, accepts the Specview call, completes the requested operation, and responds with the calculated result or an error. Then, Specview receives the SAMP reply from Sherpa and displays the fit result or handles an error accordingly.

Original Specview fitting and data i/o capabilities will be maintained and a mode switch will bring Specview up in VAO mode for Iris.

2.4 SEDImporter

The SEDImporter reads tabular data in TSV, CSV, FITS, or VOTable format and transforms the data into a VO compliant SED. Output data is written to disk or made available via a SAMP connection. The STILs library is utilized to read input data and SEDLib is used to write the resulting SED. The user interacts with the GUI to provide metadata information. Alternatively a template file manages metadata input when in batch mode. The tool can input 1 or more files. In the later case, it provides the ability to build a SED containing multiple segments.

2.5 SEDLib

The SED Library is a Java software package and provides the ability to read, write, and manipulate information contained in a SED file. Several standard formats are supported including VOTable, FITS, and XML. The library classes correspond to the objects described in the "IVOA Spectral Data Model - Version 1.03. The relationship between the objects as well as their contents closely match the schema presented in that document. All of the core classes that are defined include a common set of methods. These methods allow a user to traverse the SED model and provides basic access to all data stored within the class hierarchy. They include:

- set<attribute>() - basic 'set' method for specified attribute.
- get<attribute>() - basic 'get' method for specified attribute.
- boolean isSet<attribute>() – check if an attribute or subobject exists (is not null).
- create<attribute>() - get or instantiate the specified attribute.

SEDLib is used in Specview and SEDImporter for SED data reading and writing. The plan is also to make the library publicly available to developers.

3 Product Details

3.1 NED Service*

Type	**web service**
Languages	C
Derived Requirements	http://dev.usvao.org/vao/attachment/wiki/Projects2010/SEDAccessAnalysis/SED_Svc_Rqmts.pdf
Detailed Design	http://dev.usvao.org/vao/attachment/wiki/Projects2010/SEDAccessAnalysis/SED_Svc_Design.pdf
SVN Directory Name	EXTERNAL: IPAC
Test Plan	http://dev.usvao.org/vao/wiki/Projects2010/SEDNEDServiceTestPlan
Target Platforms	Linux, OSX

3.2 Iris*

Type	**desktop tool**
Languages	Java, Python, C++
SVN Directory Name	EXTERNAL: http://cxc.cfa.harvard.edu/contrib/sed/ (Beta1 location)

Test Plan	http://dev.usvao.org/vao/wiki/Projects2010/SEDUAIntegrationTestPlan
Target Platforms	Linux, OSX

3.3 Specview

Type	**toolkit -- 3rd party**
Languages	Java
SVN Directory Name	EXTERNAL: http://specview.stsci.edu/download/ (STScI location)
Test Plan	http://dev.usvao.org/vao/wiki/Projects2010/SEDSPECViewTestPlan
Target Platforms	Linux, OSX

3.4 Sherpa

Type	**toolkit -- 3rd party**
Languages	Python, C++, Fortran
SVN Directory Name	EXTERNAL: http://cxc.harvard.edu/contrib/sherpa/ (SAO location)
Test Plan	http://dev.usvao.org/vao/wiki/SEDSherpaTestPlan
Target Platforms	Linux, OSX

3.5 Specview-Sherpa Access

Type	**toolkit**
Languages	Java
Derived Requirements	http://dev.usvao.org/vao/attachment/wiki/Projects2010/SEDAccessAnalysis/Specview_req.rtf
Detailed Design	http://dev.usvao.org/vao/attachment/wiki/Projects2010/SpecviewSED/SpecviewSherpaAccess.pdf
SVN Directory Name	EXTERNAL: http://specview.stsci.edu/download/ (STScI location)
Test Plan	http://dev.usvao.org/vao/wiki/Projects2010/SEDSPECViewTestPlan
Target Platforms	Linux, OSX

3.6 Sherpa-SAMP module

Type	**library**
Languages	Python
Derived Requirements	http://dev.usvao.org/vao/attachment/wiki/Projects2010/SherpaFittingPDD/SED_fitting_requirements.pdf
Detailed Design	http://dev.usvao.org/vao/attachment/wiki/Projects2010/SEDAccessAnalysis/SED_fitting_design.pdf
SVN Directory Name	VAO: sherpa-samp
Test Plan	http://dev.usvao.org/vao/wiki/Projects2010/SEDSAMPTestPlan
Target Platforms	Linux, OSX

3.7 SEDImporter*

Type	**toolkit**
Languages	Java
Derived Requirements	http://dev.usvao.org/vao/attachment/wiki/Projects2010/SEDAccessAnalysis/SedImporterRequirements.pdf
Detailed Design	http://dev.usvao.org/vao/attachment/wiki/Projects2010/SEDAccessAnalysis/SedImporterDesign.pdf
SVN Directory Name	VAO: sedimporter
Test Plan	TBD
Target Platforms	Linux, OSX

3.8 SEDLib

Type	**library**
Languages	Java
Derived Requirements	http://dev.usvao.org/vao/attachment/wiki/Projects2010/SEDAccessAnalysis/SedLib_Requirements_20110120.pdf
Detailed Design	http://dev.usvao.org/vao/attachment/wiki/Projects2010/SEDAccessAnalysis/SedLib_Design_20110120.pdf
SVN Directory Name	VAO: sedlib
Test Plan	http://dev.usvao.org/vao/wiki/SEDLibraryTestPlan
Target Platforms	Linux, OSX

4 Exceptions to Development and Testing Guidelines

This section enumerates and justifies any expected deviations from the baseline guidelines for development and testing.

- **NED Service code will be archived at IPAC:** Because of the tight integration of the service with NED, it is appropriate to consider this derived from NED and subject to IP restrictions.
- **Specview code will be archived in an STScI repository:** Specview is a pre-existing product originally developed outside the VAO with purposes and an intended user base that is beyond the scope of the VAO; this code will continue to be maintained by its original author (Busko).
- **Sherpa code will be archived in an SAO repository:** Sherpa is a pre-existing product originally developed outside the VAO with purposes and an intended user base that is beyond the scope of the VAO; this code will continue to be maintained by the Sherpa development team at SAO.

5 VAO Project Inputs needed to support release

ID	Requirement statement	Notes
1	VAO Software License	Need license/policy; will use institution licenses otherwise
2	VAO Registry for NED Service	Getting NED Service registered
3	VAO Host for tool/doc	Host Iris from a VAO Ops site -or- SAO? NED Service from IPAC.
4	VAO User help/Helpdesk plans	Need help pointer in docs; web pointer, Jira, ???

6 Project Schedule

6.1 Milestones

Date	**Milestone description**	**Requirements enabled (if appl.)**
Apr 1	Beta1 Code Frz	PEP thread: NED->Iris
Apr 15	Beta1 Release	Complete integration, testing, packaging, documentation
May 27	Beta2 Code Frz	PEP thread details: editing, metadata display, conf limits, merge data; handle TSV, CVS formats
Jun 03	Beta2 Release	Complete integration, testing, packaging, download testing, documentation
Jul 15*	Final Code freeze	Science data & High priority requirement completion
Jul 29	Release 1 (Beta)	Final Integration, testing, packaging, download testing, documentation

*Date marked with * is approximate.*

6.2 Resources

Developer Name	**Contributions**
Rick Ebert	NED Service lead, NED Service coordination
Olga Pervuna	NED Service developer
Ivo Busko	Specview lead/developer
Stephen Doe	Sherpa Lead / Optical Model Module deveoper
Brian Refsdal	Sherpa-SAMP module developer, Specview/Sherpa? Integration, Iris Packaging
Mark Cresitello Dittmar	SEDLib lead
Joe Miller	SEDLib developer
Omar Laurino	SEDImporter developer, SEDLib support, Test Integration