VIRTUAL ASTRONOMICAL OBSERVATORY

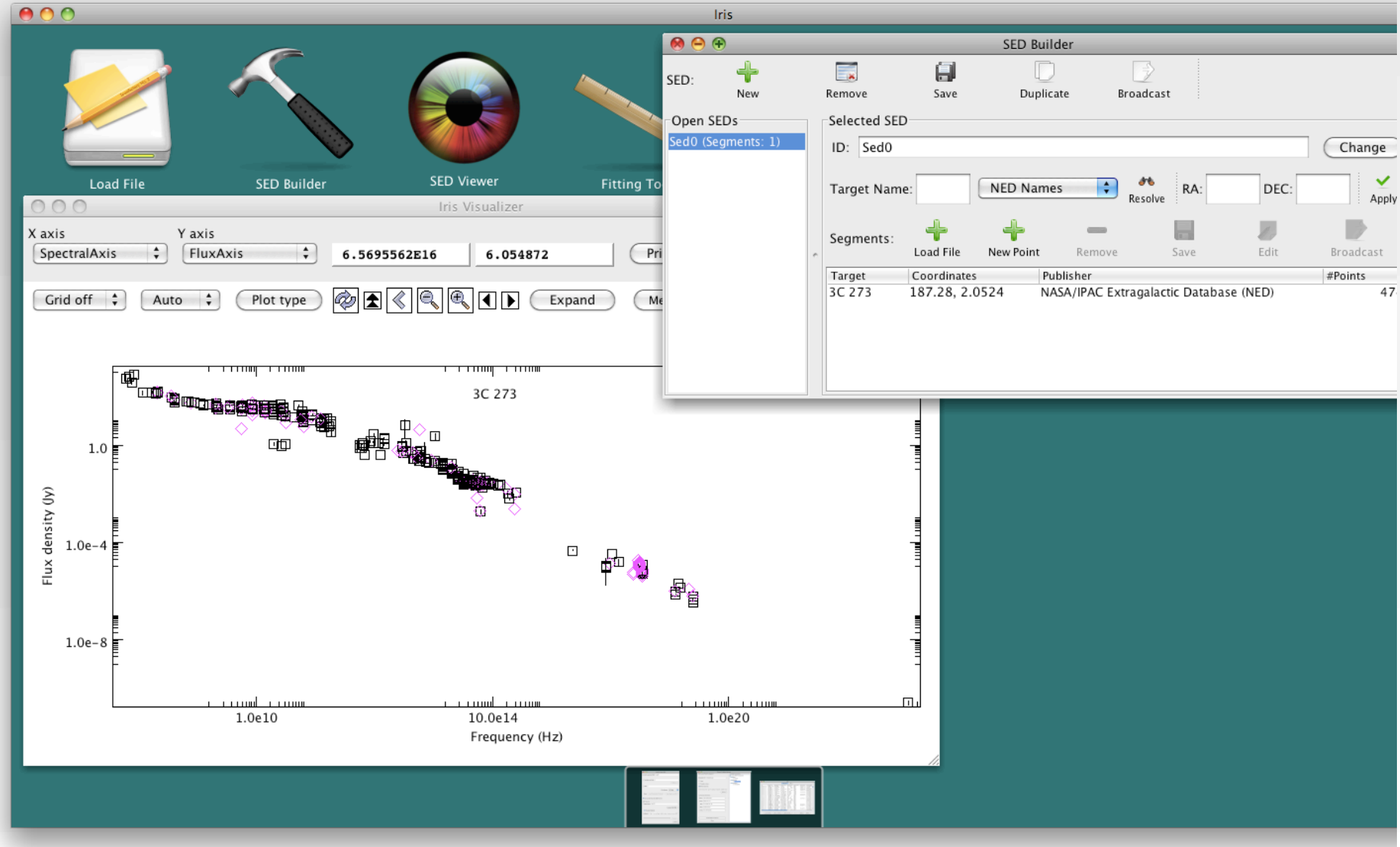
# Iris v1.1

Omar Laurino, Janet Evans, Mark Cresitello-Dittmar,
Raffaele D'Abrusco, Stephen Doe (SAO);
Ivo Busko (STScI);
Rick Ebert, Olga Pevunova (IPAC)

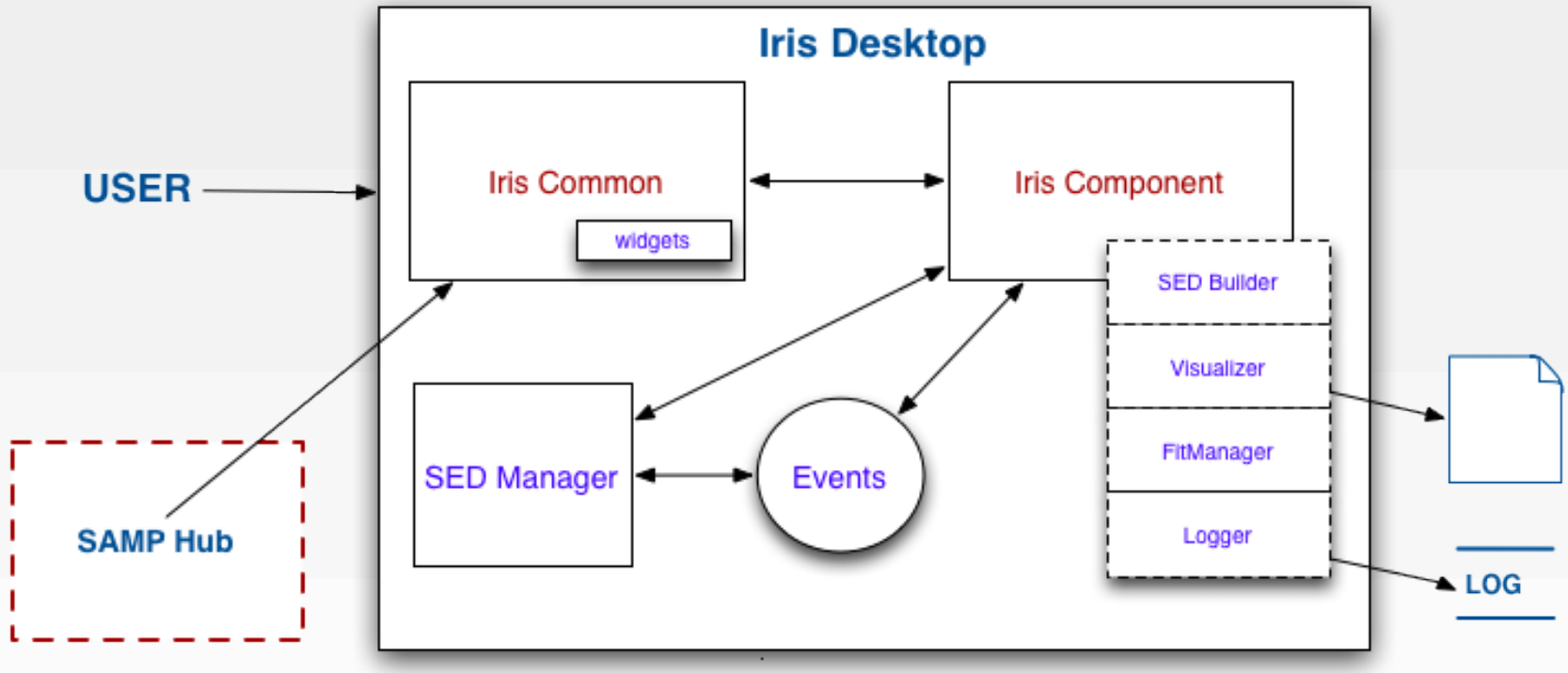
# Iris Desktop

- New Integrated Application Framework:
  - Consolidation of GUI and CLI

# Iris Desktop



# Architecture



# Iris Common Framework (1/2)

- IrisComponent Java interface:
  - Binding of components is deferred to runtime.
  - New components leveraging the Iris Common Framework can be developed by third parties to extend functionality.
- IrisComponent implementations are loaded and bound at runtime. They can contribute:
  - Menu items (in the Tools or File Menu)
  - Desktop buttons
  - SAMP Handlers.
  - Basic Command Line Interface processors (e.g. builder, smoketest) -> integrated basic CLI.
- Hooks to the Application context are provided during IrisComponent initialization (dependency injection).

# Iris Common Framework (2/2)

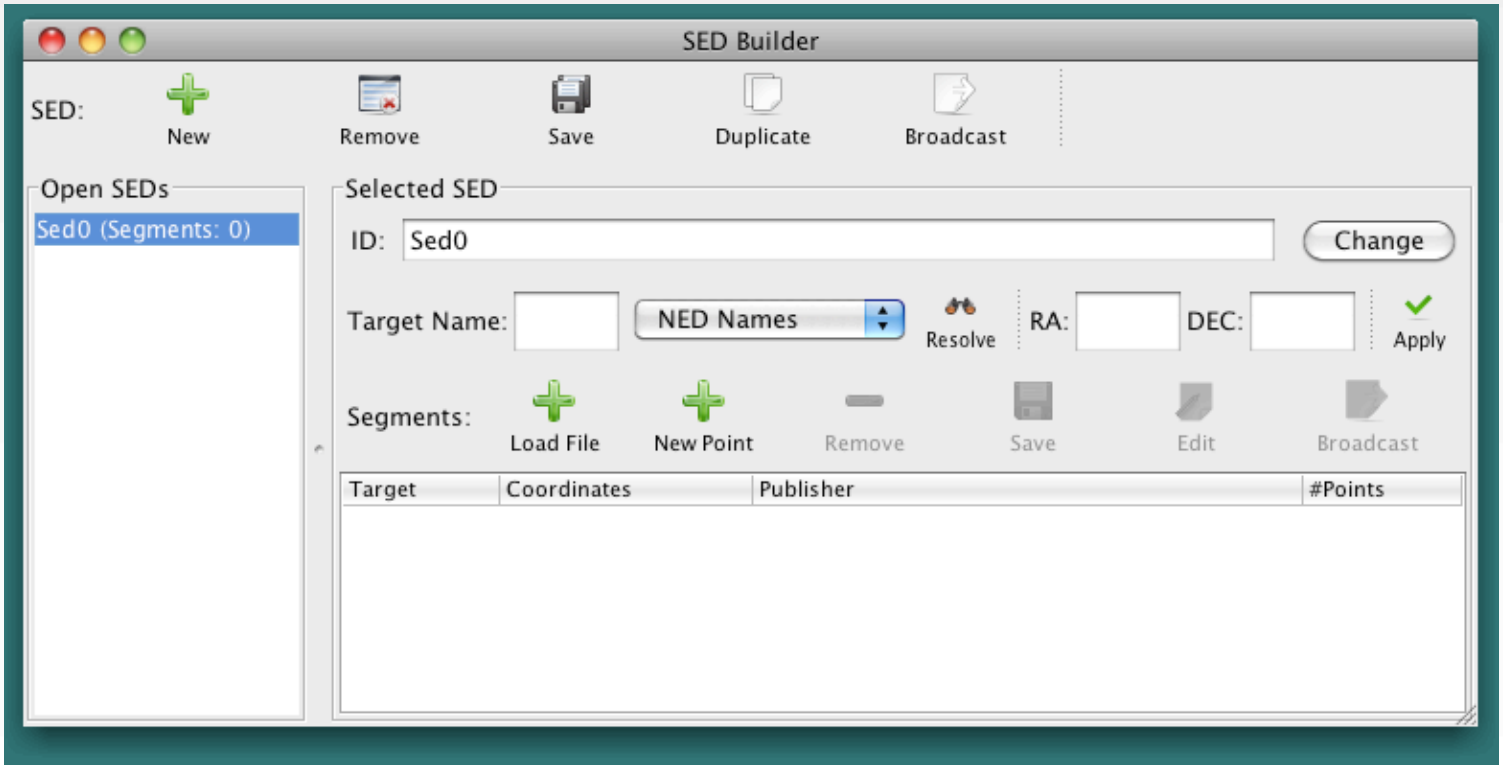
- SED Manager
  - Abstracts common SED services (CRUD operations + SED attachments).
- Extensible Events Framework
  - Enables loose coupling, extensibility. Improves user experience.
  - Baseline for registering user's session as a runnable script.
- Interoperability
  - SAMP Controller is part of the Common Framework.
  - Components are abstracted from SAMP connection details.
  - A hook to the SAMP Controller is provided, if ever needed.

# Application Development Kit

- Create a new basic SAMP-enabled application in minutes:
  - AbstractIrisApplication provides boilerplate code, management of registered components, SAMP infrastructure and command line parsing/dispatching to components.
- Useful extensions:
  - ASDC custom component
  - Data Mining add-on
  - Aladin photometry extraction from images ?
  - Integration of Topcat, Aladin, Application Launcher?

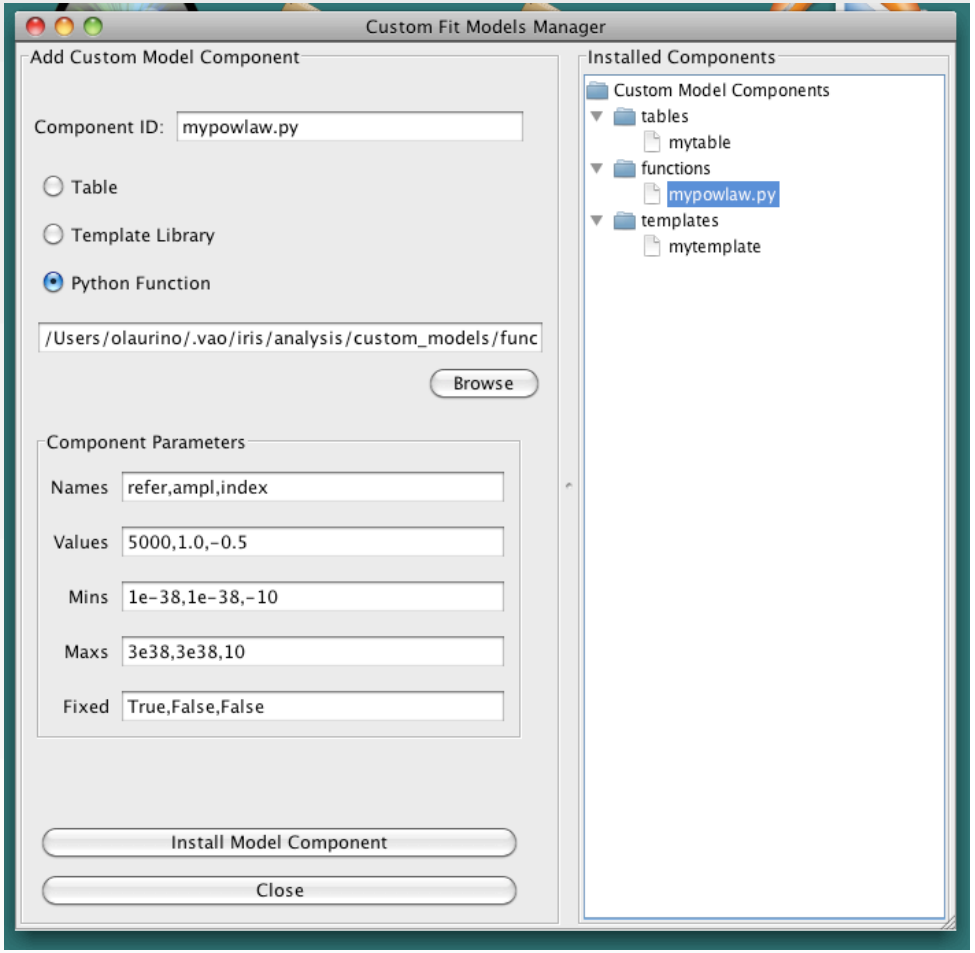
# From SED Importer to SED Builder

- Support for Photometry Catalogs
- Support for Photometry Points
- Several enhancements to User Interface
- Online Validation Framework



# SED Analysis

- Support for custom models:
  - Templates
  - User defined tabular models
  - User defined python functions



# Other features

- SEDLib: achieved interoperability with TOPCAT by removing dependency on VOTable GROUPs
- Visualization: new metadata browser with hierarchical filtering
- Logger: first step towards logging of user actions into runnable scripts
- Flexible SAMP I/O (and experimental support for SSA)
- Photometry Filters Browser (uses a static filters database from SVO. Dynamic support to be added)

# Extension points

- File Filters:
  - Read files in unsupported formats (HDF5, ASCII flavors, etc..)
- Photometry Catalog import plug-ins:
  - Import data for specific instruments/collaborations
- Custom Components:
  - Add menus, desktop buttons, SAMP handlers to add functionality.
- Custom Fit models (template fitting, python functions, simple tables)

# Technical Remarks

- Issues with current Spectral DM draft:
  - Had to workaround the lack of support for photometry catalogs (more than one flux per row, possibly one row per source), unless we give up FITS support and interoperability with TOPCAT

- Iris Common Framework:
  - A customizable, reusable, extensible, SAMP-enabled application development framework.

- SAMP Factory:
  - A meta-programmed, SAMP-backed implementation for Java Interfaces (baseline for SAMP-API and Iris Python CLI).

# SAMP Factory (1/2)

- SAMP-backed implementation for Java interfaces.
- Currently supported data structures:
  - Java primitives (+ boxing classes and String).
  - Lists of primitives and complex objects.
  - Arrays of Doubles (serialized as base64 strings, directly deserializable by Python).
  - Complex objects are recursively rendered in a hierarchical structure until primitives are found and (de)serialized.

# SAMP Factory (2/2)

- Features:
  - Instantiate a SAMP-backed java interface obeying to a generic convention.
  - Create a SAMP message using an object instance, according to a certain interface, with a given mtype.
  - Get an object instance backed by a SAMP message (or a slightly simpler Map), deserialized according to a certain interface.
- Baseline for a generic SAMP-API dispatcher that would dynamically render interfaces available to any SAMP client (e.g. Iris Python API).

# Interoperability and Infrastructure

- SAMP I/O:
  - Generic Input: votable and fits mtypes -> First try to import the file using SEDLib; if the file is not compliant, launch a SAMP Chooser (is it a Spectrum/SED or a Photometry Catalog?).
  - SSA Input.
  - Generic Output: SAMP specs forced only one segment per SAMP message. VAO custom semantics allow reconstruction of the SED.
- We are still relying on a direct NED client for SED data discovery (direct clients to an ASDC service and to the SVO photometry service might follow).

# Priorities for version 2.0

- Batch Mode and Command Line Interface
- SED Display Enhancements

- Upgrade to SpectralDM 2.0 (SEDLib and NED)
- SED DAL Service?

- Stellar models
- Synthetic Photometry

- Study science use cases coming from feedback (e.g. CANDELS) and evaluate what 'glue' is missing

# Iris 1.1-beta6

- Available for testing!
  - http://cxc.cfa.harvard.edu/contrib/sed/
  - Documentation is being updated
  - Distribution comes with a README-BETA file with instructions
- Please contact me if you need more information or assistance

- Documentation for v1.0 (w/ video tutorial):
  - http://www.usvao.org/tools/index.html
- Send feedback and/or bug reports to the VAO Help Desk:
  - http://www.usvao.org/contact.php