Science Data Reduction and Analysis Planning

Perry Greenfield & Harry Ferguson

Space Telescope Users Committee

14 November 2008

Focus

- Cross-mission planning
- Cooperation/Collaboration with other observatories
- Infrastructure
  - E.g. IRAF/STSDAS/pyraf
  - To a lesser extent, pipeline control system
- Mission-specific calibrations
- More generic applications and tool sets

Background: Future of IRAF and STSDAS

- Coherent long-term planning is needed
  - Affects large segments of the community
  - Requires coordination between centers
- Widespread recognition that SPP and CL should be phased out.
- STScI has already moved away from a strong dependence on IRAF infrastructure (pyraf, python)
- Common interests for STScI, NOAO, Gemini, LSST, Subaru
  - Meeting in June 2008 generated some momentum
  - Common NOAO/STScI release targeted for 2009
  - Agreed to work together on a whitepaper and subsequent architecture document.
  - Funding realities appear to be subverting the goal of short term results.

Why IRAF needs to be replaced

- Although users may find it still it quite usable, the architecture is increasingly difficult to maintain.
- Languages (SPP, CL) are nonstandard
  - Lack modern features and tools (e.g. debuggers)
    - Low productivity programming environment
  - Difficulty hiring developers; SPP experience is evaporating
- Difficult to integrate outside software libraries
  - Unable to use much useful software developed outside IRAF
- NOAO support funding very limited for IRAF
- Few big projects using it for new software

Current STScI software approach

- All new software written in Python/C
  - As much as possible in Python
  - As little as necessary in C
  - Expose functionality at all levels for use by astronomers
    - Python can be used much like IDL for data analysis
    - IRAF provides only large, black-box task units of functionality
- PyRAF provides bridge to legacy IRAF/STSDAS/TABLES tasks and functionality
  - PyRAF provides a common user interface to IRAF tasks and new Python/C tasks.
  - Allows us to add non-IRAF tasks to be used alongside existing IRAF/STSDAS/TABLES tasks.
- Leverage as much as possible off of open source efforts
  - As well as contribute to them (e.g., numpy, scipy, matplotlib)
- So, why didn't we…

Why not IDL?

- At the time, not possible to do what PyRAF does for IRAF
  - Perhaps possible now, but with much greater difficulty
- Poorer general purpose language than Python
  - Namespace issues hinder large projects
  - But Python has nearly all the functionality that IDL has
- Lacks consistent set of applications, user interfaces.
- Not open source
  - Fate hinges on fortunes of one company
- Licensing cost issue for smaller departments.
- Poorly organized open source community
  - ASTROLIB big success, not much else exists though.
  - Other code passed around informally
- Many major projects adopting Python as their scripting language (STScI/Gemini, LSST, ALMA/CASA, PyMIDAS, AIPS). Opticon makes it the primary choice for scripting.
- STScI already invested much in Python approach now.

AURA center and affiliates perspectives

- NOAO: Still IRAF-centered, but realizes that IRAF needs to be replaced.
  - Currently has severe lack of resources
- Gemini: Adopted PyRAF, and general STScI approach to writing new tasks in Python and C
  - Large base of CL code
- LSST: Developing new C++ libraries for processing (with python wrappers)
  - Would like to share code (and effort) with other centers
- VAO: Some overlap in needs with data analysis
  - Short term funding gap
  - Seeking inclusion of VO-based data access methods into analysis software.
- Suburu: Largely IRAF-based.
  - Trying to make IRAF 64-bit compatible

Internal working group: work in progress

- Formulating use cases
- Propagating toward goals/attributes of a new software framework
  - Standards, protocols
  - Middleware
  - Programming languages
  - Applications, tools, libraries
- Transition planning
  - Evolution rather than revolution
- Use of external libraries
  - Licensing issues
- Recommendations

Issues being discussed

- New Capabilities
    - Optimal co-addition & spectral extraction from data cubes (JWST in particular)
    - Optimal photometry on mixed-resolution, multiwavelength datasets (e.g. HST & Spitzer, JWST & ALMA)
    - Improved data mining - better connection between catalogs and source data.
- Infrastructure
    - Large datasets, 64-bit computing
    - Multi-core computing
    - Long-term support for STSDAS tables (.tab format)?
    - Long-term support for SPP and CL
- Connection to VO and other efforts
    - Relevance of Opticon requirements
- Accessibility to astronomers
  - Making easy for astronomers to:
    - Modify existing programs
    - Write their own
    - Access functionality at all levels

High Level Goals

1. Ensure that scientific results can be derived reliably and efficiently from our missions.
2. Facilitate reliable and efficient joint analysis of data from our missions and other major observing facilities.
3. Ensure that the overall community-wide effort to develop tools for data reduction and analysis is cost effective.

Derived Attributes

1. Open source.
2. Support most computing platforms in use by astronomers.
3. Evolve in response to changes in the computing industry.
4. Convenient for individual astronomers to install and use.
5. Allow users to re-run the calibration routines used in our pipelines.
6. Allow users to work on data interactively and in batch mode.
7. Read and write standard astronomical data formats, implement standard astronomical protocols, and interact with standard services such VAO.
8. Provide modular tools for building customized data reduction and analysis applications.
9. Provide high-level applications for reducing and analyzing data from our missions.
10. Version control and numbered releases with documentation and a schedule that allows astronomers to build around the system.
11. Standard languages and interprocess communication protocols.
12. A graphical user interface (GUI) layered on scriptable components

High-level services

- Image display and manipulation
- Two-dimensional and three-dimensional graphics.
- Image arithmetic, with the ability to deal with and propagate header information, uncertainties, and flags.
- Image registration and rectification
- PSF measurement, generation, and convolution
- Photometry and astrometry and shape measurement of single sources (stars and galaxies), crowded fields, and deep fields.
- Single-spectrum analysis (velocity, line strengths & widths, etc.), multiple-spectrum analysis, 3-d data-cube analysis.
- A convenient method for editing and storing parameters.
- Logging, command-recall, and tools to simplify creating scripts.
- Tools to sort, filter and edit, and otherwise manipulate images, spectra, and tabular data.
- Tools to generate artificial data
- Synthetic photometry and tools to estimate exposure times
- Calibration routines. Easy access to well-documented calibration reference files.

Some near-term objectives

- Improve IRAF/STSDAS/Pyraf distribution
  - One-click installation on standard platforms
  - Requires some coordination with NOAO
- Improve training and documentation for PyRAF and Python development
  - Tutorials at STScI? AAS? Summer school?
- External libraries
  - E.g. scipy?
- Standards development/adoption
  - Support VO standards
  - Evolving FITS support
  - Interprocess communication
  - Metadata for catalogs

Issues for STUC

- Use cases
  - Capabilities not present in current software that would help your science?
- Thoughts on the evolution of IRAF/Pyraf/STSDAS?
  - Are there things we should be doing that we haven't mentioned?
  - Are there things we are doing that we shouldn't?
  - What is better done by the general community?
- How important is this infrastructure to the community?
- User community needs normally get lowest priority if not needed for mission operations.
  - Examples:
    - E.g., unified software distribution
    - Training and documentation
    - General analysis tools
  - Unless users clearly voice their needs to raise the priority

XXX

– XXX

XXX

– XXX