Implementation of the Standard Analysis Environment (SAE)

James Peachey

(HEASARC/GLAST SSC—GSFC/L3)

Overview of the SAE

- SAE is the set of software which will be available to guest investigators for analyzing GLAST data
- LAT and GSSC teams are working together to develop SAE
- Two communities will be served by a single software analysis environment
  - Astronomy community:
    - Prefer well defined tools, scripts, cookbooks
    - Familiar with Ftools, Xspec, etc.
    - Interested in multi-mission analysis
  - High energy community:
    - Prefer tool kit from which to write own custom tools
    - Familiar with CLHEP, Root, etc.
    - Prefer object-oriented development frameworks
  - Both:
    - Want to reuse familiar tools
    - Want new tools to behave similarly to familiar tools

SAE Development Goals

- The goal is to develop a software system which...
  - Is scientifically valid and complete
  - Meets the needs of its users
  - Is supportable and maintainable by GSSC and HEASARC
  - Is of high quality
  - Is delivered on time

Development Timetable

- The GLAST mission timetable includes three “mock data challenges” prior to the launch:
  - Data Challenge 1 took place December 2003 - March 2004
    - SAE prototypes evaluated
    - Requirements refined
  - Data Challenge 2 planned for spring, 2005
    - Software will be provided in mature, albeit incomplete form
  - Data Challenge 3 planned for spring, 2006
    - Software will be very close to final form (beta release)
  - Launch will be February, 2007
    - First public release available
- This suggests two major development cycles of about one year each followed by a shorter cycle for refinements

Meeting Needs of Users

- Support for analyzing GLAST data with existing tools
  - Conventional, OGIP compliant file formats
- Support for analyzing other missions with GLAST software
  - Whenever possible, software will be multi-mission
- Look and feel of SAE applications will be similar to existing tools
  - Ballistic Ftools-like interface and behavior whenever possible and appropriate
- Documentation
  - Developers are writing documentation as they go
  - LAT team has professional technical writers
- Ease of use
  - LAT team working on easy installation procedures
  - Plan for GUIs and data visualization capabilities layered on top of applications

SAE Technologies

- Languages
  - C++ (ANSI/ISO standard compliant)
  - Python (Scripting, GUIs)
- Software Packages
  - Cfitsio (FITS file access)
  - PIL (SAO host-conforming parameter interface for user input)
  - WCS (Coordinate transformations)
  - HEADAS (FITS utilities, support libraries, and container for the above libraries)
  - Root (Data visualization, GUIs)
  - CLHEP (Mathematical utilities)
- Development Platforms
  - Intel Linux, GNU compilers
  - MS Windows, Visual Studio compiler
- Supported Platforms
  - Planned support for same Unix platforms as HEASARC software

Development Methodologies

- Short, iterative build cycles
  - Ensures that most important features are added first
  - Allows flexibility in schedule for unanticipated issues
  - Provides natural points for internal test releases
  - Frequent feedback keeps development on course
- Unit tests developed first in each build cycle
  - Provides metric for progress
  - Promotes more robust software
  - Allows changes to be made with confidence
- Modular, object oriented design
  - Individual applications are small, consisting of well-defined interactions between a small number of loosely coupled objects

Quality Assurance

- Testing
  - Unit tests developed concomitantly with the code
  - Time allotted prior to each data challenge and launch for system and integration testing
- Coding standards have been established by the LAT team
- Code reviews are being organized by the LAT team
  - Identifies discrepancies between requirements and actual code behavior
  - Checks for adherence to coding standards
  - Provides feedback regarding usability

Summary of State of SAE

- High level design phase of SAE was completed, now entering implementation phase
- A realistic development schedule which meets the requirements in a timely manner has been created and is being followed
- Ample resources are being deployed to ensure software fulfills its requirements

Appendix: Summary of SAE Development Schedule

	DC1	DC2	DC3	Launch
Likelihood Analysis	Unbinned analysis	Some binned analysis	Full binned, unbinned analyses	Refinements and integration
Pulsar Analysis	-	Write timing info, basic period search	Full period search	Refinements and integration
GRB Analysis	Prototypes	Support for standard analysis	Advanced multi-dimensional analysis	Refinements and integration
Catalog Analysis	-	Framework and common cats, bkgnd model	All required catalogs, refinements	Refinements and integration
Obs Simulation	Simple sources, bursts	Pulsars, AGN	Refinements	Refinements and integration
User Interface	-	Basic data visualization, basic GUIs	Scripting, advanced visual/GUIs	Refinements and integration
General Utilities	Common database access	More DB access, data selection	Refinements	Refinements and integration