

A Framework-Based Approach to Science Software Development

The use of software frameworks to increase reuse, and reduce cost and product development cycles is becoming increasingly prevalent in the software industry. This paper describes the application of the framework-based approach to the development of science data processing software for a NASA Earth Observing System (EOS) mission.

The Tropospheric Emission Spectrometer (TES) is a Fourier Transform interferometer scheduled to fly on the EOS Chem-1 spacecraft in December 2002. The project is managed by NASA's Jet Propulsion Laboratory (JPL). In its five-year mission, the TES instrument will provide the world's first three-dimensional global data set of tropospheric ozone and its precursors. The instrument will produce over 6TB of raw data per year. TES is an unprecedented instrument in terms of its combined performance, resolution and operational capabilities. Evolution of the algorithms, and the production software that implements them, is expected to continue throughout the mission, and accelerate in the first months after launch.

Production data processing will be performed by the EOS Data and Information System (EOSDIS) using software developed by the TES project. The production environment is a highly automated batch-oriented system that places a number of significant constraints on the implementation and management of the processing software. Data processing activities are planned to continue through December 2011. A more flexible version of the software will be used at JPL support ongoing algorithm development, research and operations.

Adopting a framework-based approach was a major strategic decision. Developing a framework is considerably harder than simply developing a one-off system. The costs of doing so had to be weighed against the expected benefits.

The decision to use a framework-based approach to TES was made for several reasons. Given the long lifetime of the project, maintenance was expected to be a significant cost factor. The trend towards faster, cheaper missions is anticipated to continue, with increasing pressure on operational projects to do more with less.

The TES framework comprises the following elements: Data Access, Basic Data Types, Basic Algorithmic Processing, Process Control, Exception Handling, User Parameters, and Log Files. The data access package is required to insulate application code from the details of file organization, format and location. The basic data types and algorithm packages provide classes which implement linear algebra (vectors, matrices and related operations), high performance discrete Fourier transform, wavelet transforms, non-linear least-squares algorithms, and other specialized computational tasks. These packages are based on existing third-party software. The process control package insulates applications from the details of the operational job control system, provides UNIX signal handling facilities, and provides support for parallel processing. The framework as a whole must insulate applications from differences in the runtime environments of the EOSDIS and the JPL facilities.

Development of the framework is being treated as a project-within-a-project. Development will be evolutionary and incremental, and driven strongly by the needs of supported subsystems. The Unified Modeling Language (UML) and Rational Software Corp.'s Rose C++ CASE tool is being used to document design.