From Start to Finish: Python for Space Missions

Paul Barrett[1], Ciprian Berghea[1], Chris Dieck[1], Scott Evans[2], Charles Poole[2], & Dan Veillette[1]

(1) U.S. Naval Observatory, Washington, DC (2) Computational Physics, Inc.

Abstract

The software development process for many space observatories is often disjoint and inefficient due to the use of multiple languages during the different phases of mission development. Code and algorithms that are often developed using an interactive, array language during the pathfinding efforts of Phase A are often rewritten in a non-interactive, compiled language for use in the production code for Phase C. This approach leads to inefficiency in both development time and cost and can introduce errors during the rewriting process. Python is one programming language that can be used as a high-level, array language and as an efficient, production language. This paper shows how Python will be used during the different phases of development of the Joint Milli-Arcsecond Pathfinder Survey (JMAPS) space mission with an emphasis on code and algorithm reuse from one phase to the next.

Joint Milli-Arcsecond Pathfinder Survey (JMAPS)

The Joint Milli-Arcsecond Pathfinder Survey (JMAPS) is a micro-satellite mission intended to update Hipparcos astrometry. With its single aperture 19 cm telescope, JMAPS will access not only the brightest stars observed by Hipparcos, but also extend Hipparcos level milli-arcsecond astrometry to 14th magnitude stars. Combining JMAPS and Hipparcos data will provide proper motion information at the level of a few tens of micro-arcseconds per year for stars brighter than 11 magnitude in the V band. Using a step-and-stare mode concept, JMAPS can integrate longer for specific fields on the sky, which allows the JMAPS star catalog to tie directly to version 2 of the International Celestial Reference Frame (ICRF2) by observing the brightest quasars in the optical. The JMAPS program is in the design and development stage, with an expected launch in mid-2013.

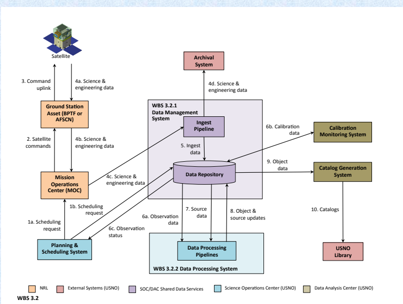

Figure 1 - Ground System Data Flow Diagram

Mission Success and Ground Processing

JMAPS is unusual for a space mission in that the timely processing of the data has a significant effect on the overall success of the mission. This is because all images are directly related during the global block adjustment (see Global Solution Simulator). In order to minimize global systematic errors, the results of the ground processing, specifically the global solution, are incorporated into the planning and scheduling system on a periodic basis. This creates a feedback loop among the Planning & Scheduling System, the Data Management System, and the Data Processing System.

Figure 1 shows the ground system data flow diagram. The core of the system is the Planning & Scheduling System (PSS) and Data Management System (DMS). These two systems form a feedback loop in order to optimize the long-term (~1 year) observing plan. The data flow begins with the PSS, where the weekly schedules are created. The schedules are sent to the Mission Operations Center for execution and also recorded in the DMS. When the science data is ingested into the DMS, it is checked against the schedule for discrepancies, which, if necessary, are included in the next long-term plan or weekly schedule.

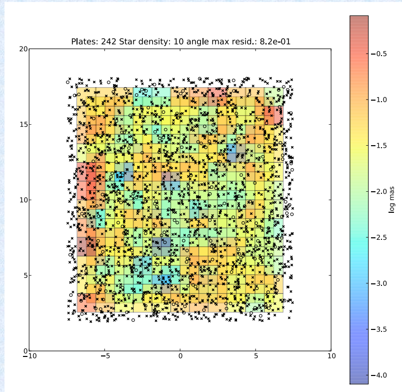

Figure 3 – A residual map from a global solution simulation. Squares show outline of images on the sky and small circles are positions of reference starts.

Code Reuse: From Pathfinding To Operations

As with all space missions, resource and schedule constraints place a strong emphasis on productivity. Reusing software is one way to improve productivity during mission design and development. In order to achieve this, a conducive environment for software development must be in place during the earliest stages of the mission. This is when most software development is being done by scientists and engineers, who do not to know, nor care, about software development methodology. However, much of the software that they develop during this mission phase can be reused in the operational system. Therefore, there is a strong desire to have a programming language and development environment that is easy to use during the design phase and that is also efficient enough for the operational phase.

One aspect of a conducive development environment is a high level programming language. This language should provide an easy interface to multi-dimensional arrays, so that scientists and engineers can easily develop code for their analysis. This language should also be object-oriented, so that the shared libraries and utilities are well defined, robust, and easy to use. If properly designed, tested, and documented, most of the shared libraries and utilities can be reused in the operational code. The Python programming language meets these criteria. In addition, there is copious evidence it increases productively by several hundred percent.

Object Modeling and Shared Code

The JMAPS program has made an effort early in the mission to develop code using Agile programming methodology. An important aspect of this approach is to begin object modeling as soon as possible by identifying and defining class interfaces that are common to the various pathfinding projects. Currently, the three pathfinding projects are progressing somewhat independently, because they have different near-term objectives. In the future, the goal is to begin integrating these projects into a larger system in order to perform more accurate simulations of the mission. In particular, there is a close association between the Planning & Scheduling Simulator and the Global Solution Simulator. This integration is most easily accomplished if each project is using a common set of classes. Examples of the common classes that have been identified to date are: Catalog (DBCatalog, FileCatalog), Satellite, Bus, Instrument, Telescope, Detector, and Scan (MeridianScan, LongitudinalScan, etc.).

JMAPS Focal Plane Simulator (Jsim)

The **JMAPS Focal Plane Simulator** (Jsim) is a Python application that is currently being develop to support requirements analysis associated with the focal plane array. Known characteristics of the detectors are included in order to understand their effect on the astrometric and photometric sensitivity of the mission. Read and photon noise, variable point-spread functions, satellite jitter, and variations of instrument parameters such as focal length and orientation have been tested.

Planning & Scheduling Simulator

The JMAPS observing concept differs from other astrometric satellites by using a step-and-stare observing mode, whereas Hipparcos and GAIA are scanning missions. The greater flexibility of this mode has its costs in that ~10000 images must be scheduled over an extended period (~1 year). The Planning & Scheduling Simulator is an effort to identify the best observing methods and to evaluate various planning and scheduling applications (both free and commercial).

Global Solution Simulator

The JMAPS astrometric telescope has a field of view of 1.56 square degrees, so >26000 images are required to cover the entire sky once. JMAPS will observe the entire sky at least 72 times during the 3 year mission. For precise astrometry, >500000 reference stars uniformly covering the sky will be used to calculate the stellar and image parameters, i.e., the positions, proper motions, and parallaxes of the reference stars and the position, orientation, and scale of the images. This involves solving for about 25 million parameters (~5 million for stars and ~20 million for images). One approach to calculating these parameters is called Global Block Adjustment (GBA). All parameters are calculated simultaneously by solving a large sparse linear system. Figure 2 is a an example sparse array containing ~400 parameters and ~2500 images.

There are several methods for solving this type of sparse system using direct and iterative (Krylov) solvers. The **Global Solution Simulator** is one pathfinding effort to investigate the best approach to solving this problem. Figure 3 is a plot of a simulation showing the variation of the residual error on number of images and the reference star positions. During operations, the results of the Global Solution will be used for long-term (~1 year) planning and weekly scheduling of the satellite.

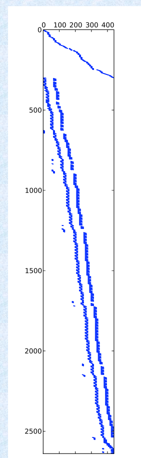

Figure 2 – Plot showing non-zero elements (blue) of a global block adjustment using the algorithm of Stock.

Summary

For this software development approach to be fully successful requires buy-in by scientists and engineers early in the program, since they are partly laying the groundwork for later software development. The concepts of object-oriented design and development often take time to grasp fully. In addition, classes that have an intuitive interface and are fully functional, take time to develop, test, and document (though the long-term return usually outweighs the initial investment). This time may be better spent on more immediate issues.

For the USNO ground segment portion of the JMAPS program, we obviously believe that the approach to software development outlined in this poster is preferred, since a large portion of the software development must be done to support the extensive simulations of the instrument and bus to assure mission success. It is therefore reasonable to reuse this code in the operational system to increase productivity and minimize risk, since the core components of the operational system must be fully functional or fully integrated into the mission by launch.