DEVELOPMENTS IN GEOMETRIC METADATA AND TOOLS AT THE PDS RING-MOON SYSTEMS NODE. M. R. Showalter, L. Ballard, R. S. French, M. K. Gordon, and M. S. Tiscareno, SETI Institute (189 Bernardo Ave., Mountain View, CA 94043, mshowalter@seti.org, lballard@seti.org, mgordon@seti.org, rfrench@seti.org, mtiscareno@seti.org).

Introduction: The scientific analysis of most planetary data products requires a detailed, pixel-level understanding of the product's geometric content and timing. For spacecraft-based data, most scientists carry out such calculations using SPICE tools. At the PDS Ring-Moon Systems (RMS) Node, we have developed an object-oriented overlay on the SPICE toolkit, written in Python, that vastly simplifies and speeds up most of these calculations. With fully object-oriented underpinnings, "Object-Oriented Python/SPICE", or "OOPS", makes it possible to determine the geometric content of any supported data product with just a few lines of code.

Applications: OOPS provides the infrastructure to support many RMS Node activities, including:

- generating representative metadata to accurately describe the entire geometric content of data products in OPUS, our search engine;
- automating the navigation of Cassini images;
- conducting a variety of research projects involving images and other data products from Cassini, New Horizons, Voyager, Galileo, and HST.

Software Design: OOPS is designed around a large set of abstract object classes, each with a variety of implementations. For example, the abstract `Path` class defines an API that can be used to determine the position of one object relative to another as a function of time. One extension/implementation is `SpicePath`, which uses the SPICE toolkit to return this information. Alternative implementations describe linear motion (`LinearPath`), orbital motion (`KeplerPath`), etc. Similarly, the abstract `Frame` class describes a (possibly time-variable) coordinate system, with implementations including `SpiceFrame`, `RotatingFrame`, and `Cmatrix`. With this design, adding a new capability to OOPS simply entails coding up a Python subclass that implements the API required by the base class.

Other abstract classes in OOPS describe `Instruments`, `Observations`, fields of view (`FOV`), Observation timing (`Cadence`), etc. Adding support for a new instrument or data type in OOPS is just a matter of programming to the well-defined APIs for these classes. Once implemented, a single Python program could be used to analyze remote sensing data products regardless of whether they are 2-D images (framing, pushbroom, or raster-scanned), cubes, occultation profiles, or virtually anything else.

All OOPS operations handle arbitrary N-dimensional arrays using the NumPy extension to Python. A single call can, for example, calculate the line-of-sight geometry of every pixel in an image. Array operations occur quickly and, by all appearances, in parallel. The SPICE toolkit is normally too slow to handle ~ 10^6 calls, but OOPS handles this by making a reduced number of calls and then interpolating. This vastly speeds up the process while retaining full precision.

We are in the process of releasing OOPS as open source to the community. Interested individuals can contact the author for pre-release access.

Metadata: Most of NASA's planetary data sets are far too large to be used effectively without a mechanism for product-level search. Our product-level search engine, OPUS ("Outer Planets Unified Search", pds-rings.seti.org/search), allows for a very fine degree of granularity in searches based on diverse geometric constraints. For example, a search on images that captured the south pole of Enceladus at a high phase angle immediately returns the highest-quality plume images. OPUS's search capabilities are made possible because we pre-calculate the geometric content of each supported product using OOPS, and use that information to populate our databases. Tables of this same metadata can be downloaded from the node at pds-rings.seti.org/viewmaster/metadata for use in other applications.

Automated C-smithing: We are now completing a project to "C-smith" the C kernels for the entire Cassini tour. Our C-smithing yields a new C-kernel that is accurate to sub-pixel precision for most of the Cassini images. Our technique has been to generate model images using OOPS, which we then correlate with the actual data. The correlation function enables us to determine the pixel offset accurately. We have developed a variety of techniques for fitting to images of solid bodies (with and without visible limbs), atmosphere-enshrouded bodies such as Titan, rings (including non-circular features), and background stars with and without smear. With minor modifications, these same techniques could be extended to other planetary systems, missions, and data sets. If PDS could routinely provide C-kernels with pixel-level precision, we could eliminate the need for most scientists to perform image navigation as an (often tedious) first step in their data analyses.