Roadmap 2021

Herein, we present the ASC fiscal year 2021 (October 1, 2020 - September 30, 2021) software roadmap. The themes and associated tasks are ASC funded work to be completed during FY21. The enumerated tasks are those that ASC is committing to execute, but are not a complete listing of the work that could be undertaken. As an open source project, we are actively soliciting contributions from the community.

We have labeled some tasks as investigations to identify those places where we need additional information before we can estimate the full complexity and scope of a task. These tasks can lead to future work, a decision to defer, or a decision that additional information is required.

Tasks on this roadmap are living. They may move ahead to future years, be identified as being completed, or be identified as not needed. Our ultimate goal is to engage the user community to identify where we can add value.

Tasks status is identified using the following shorthand:

<table>
<thead>
<tr>
<th>Mark</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>bullet</td>
<td>work not yet started</td>
</tr>
<tr>
<td>check</td>
<td>work completed</td>
</tr>
<tr>
<td>🏋️</td>
<td>on-going work</td>
</tr>
<tr>
<td>💪</td>
<td>stretch goal</td>
</tr>
</tbody>
</table>

Some number of these tasks will also include an associate Request For Comment (RFC). We have primarily used RFCs for the ISIS project. The RFCs are added to the ISIS wiki and old RFCs are available there for reference.

Themes
Funded development for ASC software has been categorized into three very broad thematic areas: (1) Maintenance and Maintainability, (2) Value Adding, and (3) Research. In this document, we describe lower level thematic areas that ASC software will focus on. This roadmap is thematically organized because of the amount of synergy between projects. In FY21 (October 1, 2020 - September 30, 2021), we are working in the following thematic areas:

- ISIS modularization
- Continue to improve the build, test, deploy trains
- Grow and support a planetary science software community
- Begin a transition to support entirely remote SPICE
- Improve bundle adjustment capabilities and control processes

**ISIS Modularization 🏃**

- Extraction of PVL (label parsing within ISIS and general Parameter Value Language parser for other applications) and BLOB (traditionally ephemeris data stored on ISIS cube files) capabilities from ISIS into standalone libraries
- SPICE modularization to extract ISIS kernel selection into a stand alone library with a C++ interface
- Interface for reading/writing kernels
  
  🌲 Associated C++ interfaces for programmatic access to interior and exterior orientations
  
  💚 Python bindings to the PVL and BLOB libraries

**ISIS Application Conversions**

- Translate ISIS applications to support functional testing using GTest
  
  🍚 Reduce total test data volume to < 1GB and get under version control

**Release Train 🚄**

- Four ISIS releases (pending ongoing dicussion with the ISIS TC about release cadence)
- Five releases of other ASC software packages, e.g., knoten, usgscsm, plio, pyhat, autocnet, pyhat-gui, etc.
Community Growth and Support

- Centralize and standardize documentation across the software portfolio
- Publish documentation on SPICE kernel lifecycle, i.e., which kernels are updated at what cadence for active missions
- Maintain communication channels established during FY20
- Improve documentation for external contributions
- One week per quarter of mission specific support (Git submitted issues and public prioritization meetings) One week per quarter of ASC product development support (Git submitted issues and public prioritization meetings)
- Reduction of total test data volumes to support external contribution (see above)
- ISIS3 to ISISX conversion cleanup to remove all ISIS3 references

SPICE Calibration

- Update calibration routines to support remote SPICE kernels
  - Determine additional applications needing changes with the long-term goal of supporting 100% remote SPICE

Sensor Models (CSM/ISIS)

- Integrate CSM into ISIS
  - Implement two additional sensor models

Control and Bundle Adjustment

- Improve AutoCNet documentation to support use for analysis of control networks
- Operationalize FY20 efforts for network analysis including identification of regions of low control density and use of analysis methods (e.g., data snooping)
  - Implement data snooping method inside of the bundle adjustment code

PDS4 Support
• Develop a PDS4 support solution to add to the ASC software portfolio

**Additional Efforts by non-ASC developers**

• Integrate the Community Sensor Model (CSM) into the Ames Stereo Pipeline (ASP)
• Develop as easy as installation process as possible to have ASP and ISIS installed in the same conda environment
• Open Source a HiRISE de-jitter pipeline

**Change Detection**

• Fully operationalize the prototype HiRISE change detection pipeline built in FY20
• Implement 1+ additional change detection algorithms

**Summary**

In FY21 we are continuing to transition our software to a standard open source model. In FY20 we have benefited from contributions from users outside the ASC staff. We are exceptionally appreciative of these efforts and continue to seek to build a vibrant community of planetary software developers and users. This focus means we are: (1) improving the portability of our tests so that external contributors can more easily participate in a standard development cycle and so that our users benefit from improved stability; (2) improving documentation for how contributors can update documentation and tutorials thereby supporting all users of our software offerings; (3) modularizing the software into libraries to improve the developer experience and allow use by external developers in their applications; and (4) supporting the existing user community with timely releases with community driven bug fixes and software enhancements. Additionally, we are reaching out to other planetary software package maintainers with the goal of building a planetary software stack (similar in concept to the ecosystem that exists around the python scientific computing stack). This effort supports the user community by seeking to unify the software delivery and installation tools that can frequently be a hurdle for new users. We are also focusing on our core competencies, standards support and generation of the highest quality planetary data products. This includes continued development of CSM sensors, tools for assessing control network quality, and the development of support for the PDS4 format. Finally, we continue to research methods and develop tools for hard science questions such as those associated with change detection.