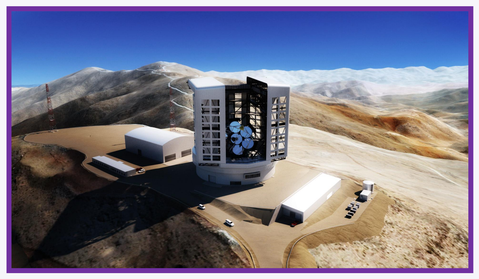
P6-9
Transforming Science Code into Maintainable Software
Insights into the GMT-Consortium Large Earth Finder (G-CLEF) Exposure Time Calculator (ETC)

Charles Paxson, Joseph B. Miller, Ian N. Evans, Janet D. Evans, Andrew Szentgyorgyi, Sagi Ben-Ami, and Jason Eastman
Harvard-Smithsonian Center for Astrophysics

Giant Magellan Telescope



Introduction: We explore a common workflow in research institutions where science code is transformed into robust, maintainable, and expandable code. We describe the process we took to develop requirements documentation and a web application from science code, and we describe some of the scientific and software highlights of the G-CLEF Exposure Time Calculator.

G-CLEF Spectrometer

Science Requirements -> Maintainable Software

Common Science Requirement Format

Executable science software from which software drives or creates: requirements document, background and theory, use case, test cases and validation results

Science Code Upgrades Aided by Evolving Requirements Documentation

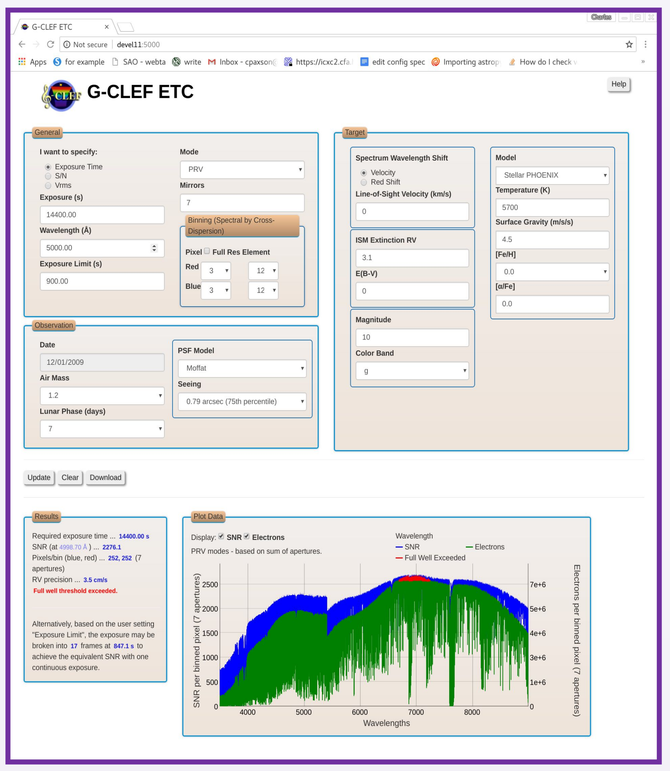
1. Read Noise added for each aperture
2. Transformed model spectra based on varying instrument resolution
3. Accounted for Spectral and Spatial profiles
4. Dark Noise
5. Added Primary and Secondary Mirror reflectances to total Throughput
6. Eliminated duplication of spectrum creation
7. Signal-to-Noise per binned pixel, per resolution element, per pixel
8. Angstrom per pixel, spectral & spatial factors, sky noise

Discussion

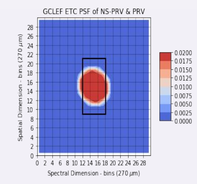
Without documentation or context, we needed to reverse engineer the algorithms. Magic numbers identified, improvements to the robustness of the code was necessary especially for corner cases, and bugs fixed. Specifically for the G-CLEF ETC, convenience factors such as angstrom per pixel, spatial factors, and more were difficult to trace to fundamental physics expressions. Units were omitted leaving uncertainty whether results were per pixel or resolution element. Science bugs such as read noise, omissions of loss components in optical path, omissions of the wavelength varying resolution needed attention.

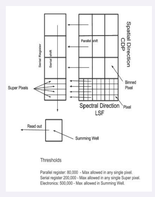
From this scrubbing of the first generation science code, a requirements document was iteratively assembled by software and science correspondence. Slowly the fundamental equations were identified, and rewritten for clarity. Software personnel documented the findings and scientists approved, clarified and were able to direct enhancements within the style of the code. The document ultimately allowed science and software to come to consensus on bug fixes and improvements.

G-CLEF ETC Web GUI



Example of ETC Back End Calibration Data and Algorithm





Observations

Delving into science code without documentation is time-consuming and costly work. Nevertheless, we persevere under these not so unusual circumstances and point out the benefits of building requirements docs:

- Writing explanations of algorithms and deriving equations reveals the code.
- The engineer becomes increasingly equipped to recommend improvements on the science.
- Documenting allows a scientist / software engineer a talking point. It reduces the level of sole responsibility on the engineer for scientific interpretation of the code.
- As the project progresses, and memories fade, the document serves as a resource to come back up to speed on the project.
- Future engineers and scientists will have a documented code base.
- New contributors are more easily brought into the project.
- The code has been transformed into a collective understanding, rather than the sole purview of the software engineer.

G-CLEF ETC Features

- Analysis results given goal exposure time, SNR or PRV
- Models: Stellar, Power Law, user defined
- MODTRAN atmospheric transmissivity + scattering
- ISM extinction
- Noise contributions: sky, dark, readout and square root signal
- Indicators if readout thresholds are exceeded
- Accurate Instrument resolution, proper convolution to convert from model grids to instrument grid

Validation

- Spectra validated in comparison to ESO Spectroscopic Standards

G-CLEF ETC Software Highlights

- Python, Flask, DiGraph, C (wavelength grid generation and convolution model -> instrument grid converter)
- Model-view-controller for our web application
- JSON user and static instrument inputs
- Calibration inputs: instrument and detector optical and CCD efficiencies, filters, spectral and spatial profiles
- Data model, graphical display, GUI text results, FITS file results
- Unit tests

This work has been supported by the GMTO Corporation, a non-profit organization operated on behalf of an international consortium of universities and institutions: Arizona State University, Astronomy Australia Ltd., the Australian National University, the Carnegie Institution for Science, Harvard University, the Korea Astronomy and Space Science Institute, the Sao Paulo Research Foundation, the Smithsonian Institution, the University of Texas at Austin, Texas A&M University, the University of Arizona, and the University of Chicago.

Giant Magellan Telescope image courtesy of GMTO.