Level 1 Processing Pipeline

Daniel Flath

For

Daniel Flath (Stanford),
Alex Schlessinger (SLAC)

Overview

- Requirements
  - Processing
  - Cataloguing
- How it works
- OPUS
  - Introduction & Desirable features
  - LAT-Specific Additions
- Implementation Status
  - OPUS
  - Add-Ons
- Goals
  - DC1
  - EM

Requirements

- **Processing**
  - Conversion of L0 (near-raw) to L1 data upon receipt from MOC
    - XFer of HSK data to OF
    - Reconstruction & Digitization
    - Must be complete before next downlink
    - XFer of diagnostics to OF
  - Reprocessing of L1 data as needed
  - Production of simulated data

- **Data Cataloguing**
  - Classification and storage of L1 data after processing
    - Summary information on each dataset
  - Tracking status and schedule of reprocessing

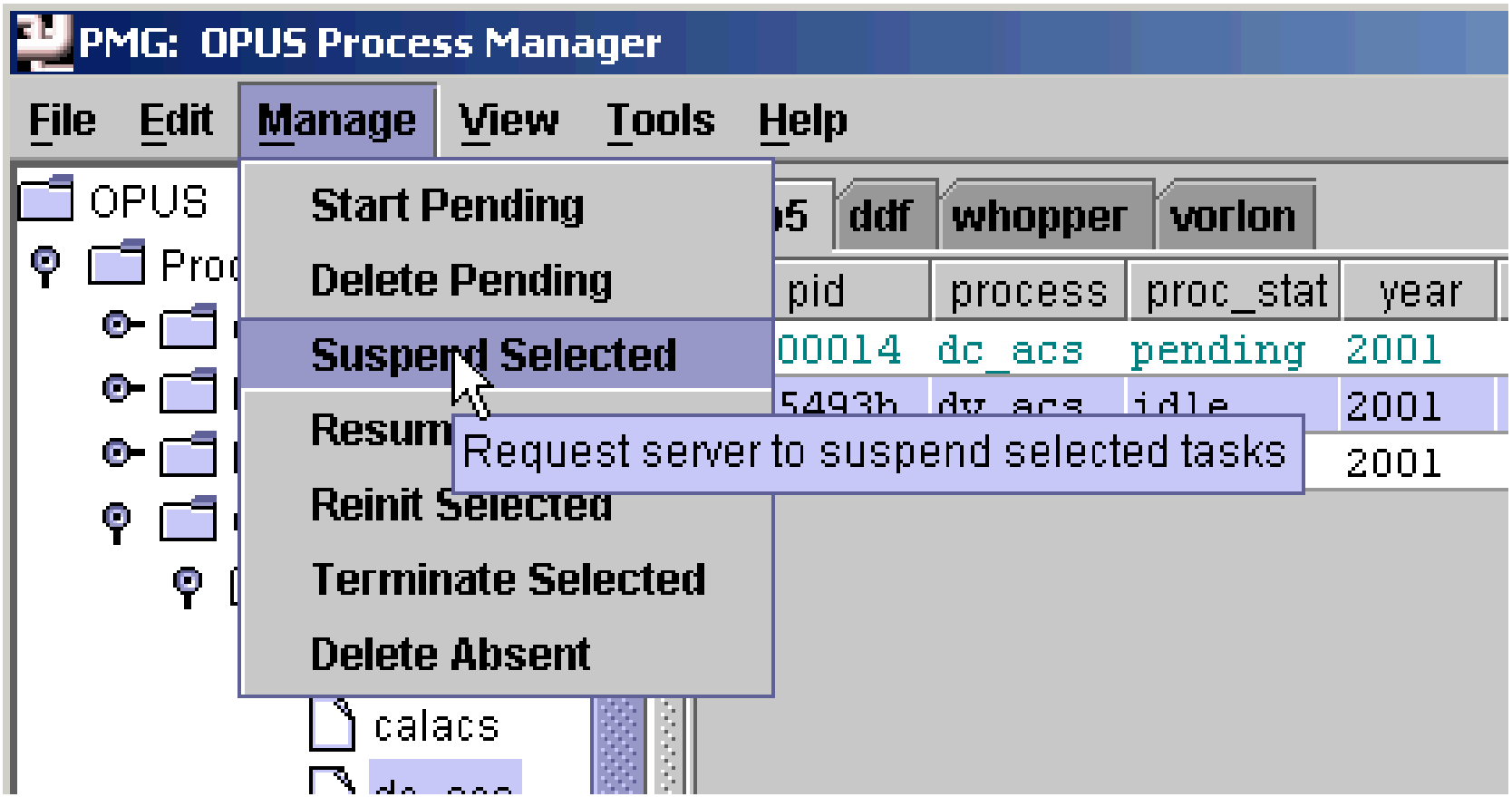
How it Works

- Pushes data through a sequence of processing steps
- Monitors status of a dataset as it proceeds through processing
- Provides notification upon failure of a processing stage
- Catalogues the result of processing for each dataset

OPUS – Introduction & Features

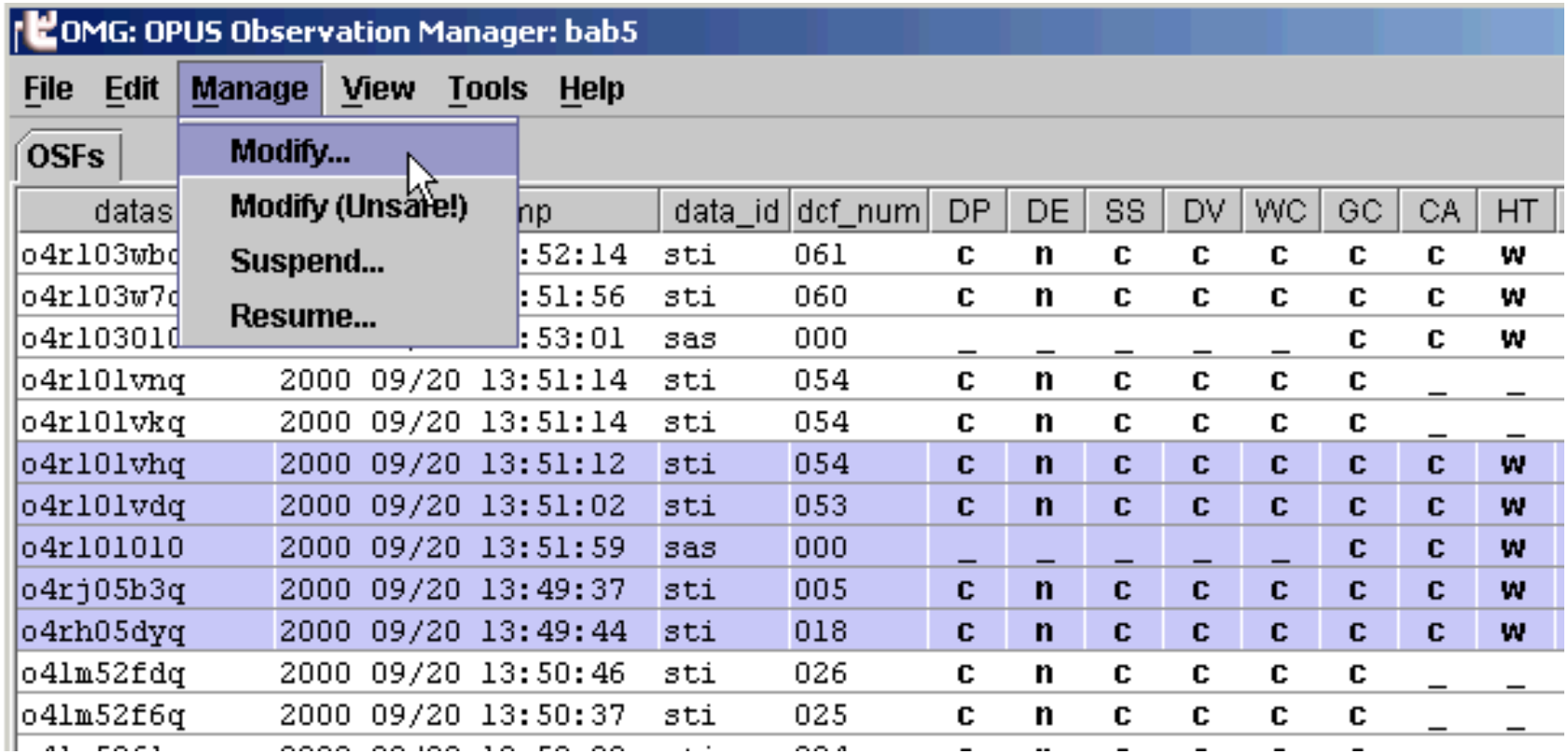
- Developed by AURA for the Space Telescope
- In use by several NASA experiments
- Does almost everything we need
  - Will run any *program* that can be "wrapped" by a shell script
  - Provides hooks for trapping processing errors
  - Distributes processing over a network of machines
  - Will run multiple processing sequences simultaneously
- Supports Extension of Functionality
  - Provides a C++ API to develop "OPUS-Aware" applications that have access to OPUS state information
- Displays live processing status (see next slide)
  - User can modify or override individual jobs & statuses

OPUS in Action (PMG)



(Image Courtesy of OPUS manual)

OPUS in Action (OMG)



(Image Courtesy of OPUS manual)

LAT-Specific Additions

- Support for the SLAC LSF Batch Processing Farm (Alex)
  - OPUS supports processing over multiple nodes using R/SSH
  - Alex is developing an interface layer that will extend this to the Batch Farm at SLAC
- Support for the ORACLE processing DB
  - OPUS saves log files containing the status of each processing stage
  - Dan is developing a set of scripts to wrap the processing database
  - At completion of processing for a particular dataset, these will catalog the status, location, and a summary of the dataset

Implementation Status

- OPUS
  - Sample pipeline has been developed
    - Runs GLEAM (Sim/Digi/Recon)
    - Runs ROOT macros to verify the output
  - Support garnered from development team

- Database & LSF
  - Database routines about 50% complete
  - LSF layer is researched and ready to be developed
  - Awaiting OPUS source code & documentation
    - Upon receipt, can develop the LSF layer and the Cataloguing program (that will use the DB routines)

GOALS

- **DC1**
  - **Stress-test Pipeline components (OPUS, LSF-Layer, DB-Layer) by running GLEAM on many machines**
    - **Full implementation contingent on getting source code from STSC**
    - **Kludge implementation possible failing this in short order**

- **EM**
  - **Test a toy version of what will be the L1P using data as it comes off the instrument**