LSST Data Management Working Group Report
Data Quality Assessment

Our group had 5 members:

- Stuart Marshall (SLAC, chair)
- Zeljko Ivezic (UW, Princeton)
- Jacques Sebag (NOAO)
- Dick Shaw (NOAO)
- Steve Kahn (SLAC)

We met during the afternoon of May 10 and briefly on the morning of May 11. We started with a relatively free ranging discussion of data quality assessment (DQA) to try to get a handle on how to structure the discussion and our report. It was clear that DQA overlaps each of the other discussion areas.

Some of the important items from our initial discussion are listed here. We need to build in the ability to fold in new things learned during operation (Shaw). Can we divide up the subject into DQA for control, calibration, DQA for data products? (Kahn). Each module should deliver a QA of some sort (Ivezic). For example, if the output of the photometric measurements in a field don't make sense, the module should flag this in real time. What about the bootstrapping phase – before we have enough operational data to compare with (Shaw). In SDSS, a per-field image grade (1 – 4) was implemented with ~10% of images getting the grade of 1 (excellent). The majority of problems were weather and calibration. Contributions were seeing, astrometric accuracy, photometric accuracy, cloud cover, and transperancy. It seems we need a nearly fully functioning DQA system ~2 years before first light (Kahn). Is there already a calibration working group? What are they doing? Consider the distinction between trends and events, (eg. slow changes of a calibration constant vs. a dust speck that moves on a detector).

We decided to organize our discussion and report into 4 sections with some initial agreement on principles we could apply to the discussion.

0. principles or generic concepts
1. Data quality for control/operation (eg feedback)
2. calibration
3. DQA for downstream data products
4. QA requirements for software (verification and validation)

Principles

- DQA support required in all software modules: either a self assessment or an output suitable for use by an external system
- Intelligent processing: modules/functions should flag & raise alarm if outputs don't make sense (somewhat redundant with previous point)
- All softwaredevelopment will obey strict configuration control. The “paper trail” should be complete: for a given piece of data, we must be able to explain how it went from raw to processed showing all code versions, calibration file sources, etc.

- DQA info is archived and available to query (eg. the papertrail)

Control/Operations

We broke this topic up into three areas that would need to be defined:

A) Define parameters characterizing an observation. This set of parameters is then used in feedback during operation to maintain quality and downstream to grade observations.
B) A "data-capture" system computes derived quantities for feedback to the control system.
C) feedback definition

- At the end of the R&D phase where should we be?
  - Item A should be completed to include all known parameters with a provision for adding others during the life of the system.
  - Item B should be designed in terms of functionality, accuracy, speed, frequency algorithms. Again, the design should include ability to extend. The inputs here are from the facitlity status system, calibration constants, and images with outputs to the scheduler/control system.
  - Item C should have interfaces defined (eg. what gets feedback to which system)
- Unresolved Issues: coupling to a facility status system (awaits system software arch)
- Teams, steps: these are mostly design documents which must be written
  - Team for items A,C: Camera, Telescope, DM subgroup. **Sebag** has agreed to supervise the formation of subgroups (doesn't mean he will lead or be a member)
  - Team for item B, **Dick Shaw** (NOAO) volunteered to work on this, also the group suggested **Thaler and Plante** at NCSA to be involved in forming a team.

The Calibration System

We need documents:

A) Calibration Plan (philosophy, steps,...) -- should be written in 12 months
B) Calibration Pipeline Software design, -- written in 18-24 months
C) Initial LSST catalog – (pre-lsst survey, use panSTARRS, others) 18-24 months
D) QA of calibration (handle monitoring events, trends, re-calibration alerts, this might be part of B above)

Need a calibration group with representation from telescope, camera, and data management. Again **Sebag** has volunteered so push the formation of a group. **Stubbs** was suggested as a second person to work with Sebag on the group asignment. We suggested the neccessity of working with other groups and existing catalogs as input to our initial catalog (bootstrapping). We discussed the scope of calibration, should it be a single isolated system or distributed amoung the various items being calibrated. Also we discussed the use of auxillary systems: outrigger telescopes both visible and IR, to assist

in calibration.

Data Quality Assessment for Downstream Data Products

- Primarily an issue of catalog QA
  - Naturally included in static or variable pipelines, starting with the QA info that was generated for control/operations
  - Statistical tests (color-color width,...)
  - Standardize input from calibrations
  - Automated “learning”
  - Artifact injection for efficiencies and completeness tests, this is not generic and probably should reside in the various pipelines. So main goal is to facilitate this feature for pipelines by having a standard set of tools for adding artifacts.
- Form groups responsible for defining what pipelines need as input and common tools
- Definition documents, generic parts and pipeline groups define their parts.
- who: **Ivezic** (UW), **Cook** (LLNL) suggested to further define this task, Ivezic volunteered.

Quality Assurance Requirements for Software

- Definition of verification and validation test suites
- Schedule time for v&v program
- Accuracy, scientific quality measurements of algorithms (not just that they don't crash)
- stress tests by independent teams (separation of code/test, can we afford this in practice?)
- who: **Axelrod** and **Kantor** should assign and supervise a group