Breakout Session Summary

Protocols/APIs for requesting observations, receiving data and scheduling

Moderator: Curtis McCully
Notes by: Rachel Street

Contributors:
Iair Arcavi, Rollin Thomas, Ken Smith, Matthew Graham, Bryan Miller,
Ben Weiner, Dave Young, Maria Patterson
Resource allocation

• How do we allocate observations to telescopes over a wider range of resources?

  Observatories publish their capabilities, and advertise their availability

  Observation requests define the capabilities they require

  Software matches the two
User story:
• What are my chances of getting my observations from telescope X tonight?
• Should I submit to multiple resources?

→ It matters if time allocations are charged by request submitted, or by requests completed.

→ But need to avoid gaming of the scheduling systems
e.g. by preventing constant deletion and re-submission
(“high frequency trading”)
Demo and Discussion of LCO API Interfaces

*Developer credit: Austin Riba, LCO Software Engineer*

Demo of the functionality of the new Valhalla system including:

- Tools indicating telescope availability recent history
- Visualizations for each observation request and parameters with related data
- Visualization of obs request converted to JSON
- Components and parameters of observation requests
Who/what decides which resource provides the observation?

Discussion of generalizing LCO model to other facilities, including:
   Enabling users to request classes of facility
   “Get me a spectrum of this target of R>R_{Min} with S/N_{Min} within T time window”
   ...across a range of diverse resources

Recognition that different users likely to have more or fewer constraints on the observation parameters

Discussion of complex observation requests:
   AND - requiring data from different resources
   Contingent – if A happened, then B
   Time variable priority weighting
Hard to implement, definition of 'success' is case-dependent
API Feedback

Essential.
Saves a lot of human effort/stress if users can find info for themselves.

[Just ask Nikolaus]

Recognition that 'the schedule' is a function of time.

Good visualization and status feedback tools are essential

Let users know immediately if observation is impossible

Pressure (overcontentention) plots for resources also useful but careful intepretation is needed

Also useful: breakdowns of observations by RA, etc.
LCO Scheduling
(The simplified version...)

Computes visibility windows for all requests
Has given metric for value of the request, then maximise the global score.
Still tuning the score metric.

Current weighting function is block length * TAC, IPP priority with small correction
factors for favoring early blocks + small random component to break ties

IPP system enables users to use 10% of time allocation at higher priority than TAC allocation.

Recommend developing a simulator to test optimization functions.

Adjustments/functionality considered:
• Weighting observations by airmass
• Balancing ‘value’ of longer blocks vs TAC priority an ongoing problem.

Discussion of how communications mechanisms between scheduler → each telescope

Useful to archive past states of the schedule, telescope availability
Need for Documentation

Good starting point for LCO to document existing protocols, terminologies etc, followed by community discussion to apply to other facilities

Including:
- Data model, observation request parameters
- What the scheduler submits to the POND
- What the instrument agent pulls from the POND
- Description of APIs

What parameters would telescopes need to advertise their availability?
→ Define a reverse API
Local Scheduling Overrides

“What if a telescope operator receives a viable observation request, but immediate conditions mean they decide not to do it?”

E.g. There’s localized cloud over that part of the sky...

Facilities may want a local veto option

Could be destabilizing if used too frequently

Enable facilities to revise their acceptance schema

Some observatories may prefer to have a ‘schedule freeze’ time each night.
Psycology / Sociology

Expectation of sociological change over time as observatories, TACs and users become accustomed to scheduling this way.

Understanding that implementation can be incremental, to allow for technological and policy development - and experimentation.
Next Steps

Publish documentation of existing APIs at LCO as a starting point.
→ Develop a template, circulate among observatories to see if their operations would fit into it.

Note Kafka, Avro supports development of schema and easy validation. Also allows aliasing between different users calling parameters by different names.

Some preference to simply list of parameters required. Investigation proposed of previous work/existing standards e.g. Astrogrid

→ Suggest we go back to the LCO software team for v1.0 schema, then circulate to workshop attendees.