Observing at CFHT, evolving from Queued Service Observing to Automated Service Observing

1. General Information

1.1 Purpose
Canada France Hawaii Telescope (CFHT), with a proven track record in Queued Service Observing (QSO)\(^1\), should implement artificial intelligence (AI) algorithms and techniques to address specific functions within the operational event loop. Utilizing AI, planning, scheduling, data acquisition, monitoring, and grading of observations can be streamlined and even automated, greatly reducing the need for human management and operation of these business functions. The operation model evolves from QSO to Automated Service Observing (ASO).

1.2 Scope
This document outlines a draft proposal that encourages the use of artificial intelligence to improve all aspects of the CFHT observing environment. Specific functions within the current QSO scheme that should be automated will be addressed. AI techniques and algorithms will be presented as replacement functionality. It is necessary to highlight the impact on CFHT from an organizational point of view as well as the computing systems.

1.3 References
“The Queued Observing Project at CFHT”, SPIE 2002, Martin et al.

\(^1\) http://www.cfht.hawaii.edu/en/science/qso.php


“CFHT DIMM Data Analysis”, *Report to the SAC 09/2009*, Racine


“Naive Bayesian Classifier”, [Wikipedia](http://en.wikipedia.org/wiki/Naive_Bayes_classifier)


### 1.4 Acronyms and Abbreviations

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AI</td>
<td>Artificial Intelligence</td>
</tr>
<tr>
<td>ASO</td>
<td>Automated Service Observing</td>
</tr>
<tr>
<td>CFHT</td>
<td>Canada France Hawaii Telescope</td>
</tr>
<tr>
<td>DIMM</td>
<td>Differential Image Motion Monitor</td>
</tr>
<tr>
<td>GA</td>
<td>Genetic Algorithm</td>
</tr>
<tr>
<td>IQ</td>
<td>Image Quality</td>
</tr>
<tr>
<td>KNN</td>
<td>K-nearest neighborhood</td>
</tr>
<tr>
<td>MDP</td>
<td>Markov Decision Process</td>
</tr>
</tbody>
</table>

Automated Service Observing          September 23, 2010          Page 2
1.5 Contact

Billy Mahoney
mahoney@cfht.hawaii.edu

2. Current System Summary

2.1 Background

CFHT has successfully operated in QSO mode for almost a decade. Outside of new instrument integration and 'learn-from-experience” adjustments, the observing paradigm at CFHT has remained relatively the same since QSO became the de facto business method. This era defined by stable and improving Queue Efficiency (QE) has shown that QSO is an acceptable evolutionary step from classical observing.

The observatory is currently under a development project to allow for remote observing (RO) – the Observatory Automation Project (OAP). A valuable collection of tools have been deployed and are being developed that will provide observers real time site monitoring capabilities to assist with QSO and RO. An underlying goal of this development is to provide more input to the operators of the business system. This input can lead to making better long, short, and real-time decisions.

A potential problem exists with having more data to make better decisions. The increased complexity of the formulas can allow the operators to make better decisions. But will the permutation set of inputs reach a size that limits how efficient, thorough, and consistent these decisions will be? In CFHT's current manual operations of QSO, there is already variability as a function of ‘who' in each of the
phases of operations. AI deals with helping computers find solutions to complex problems in a human-like fashion. Traditional AI systems borrow characteristics from the human set of knowledge of the problem to be solved, and create algorithms that can be generated into computing systems. By recreating the human approaches to problem-solving, based on abstract thought, high-level deliberative and iterative reasoning and pattern recognition, AI can be used to streamline decision-making and enhance performance of business systems. If a problem can be formulated with all known inputs and desired (and undesired) outputs, and if the problem is repetitious in nature, an AI computing system can be employed to assist and eventually manage the decision-making process.

### 2.2 System Objectives and Current Functionality

To address the integration of AI methods within CFHT’s QSO mode, it is necessary to break down the business model into the affected functional units. These units should be further evaluated to determine operating complexity level, all inputs and outputs, and business rules (i.e. functional logic). Determining the value-add of enhancing these functions with AI will require an understanding of the new development costs, current and future operating costs.

#### 2.2.1 Planning

Although somewhat intertwined with long term scheduling, planning can be categorized as its own functional business unit. Planning can be considered as scheduling beyond a one semester or even one year time frame. Large programs, agency and global scientific goals and achievements, and funding influence the decision-making process that helps long term planning at CFHT.

#### 2.2.2 Scheduling

There are specific components in the QSO dataset that allow almost all scheduling to be created by fixed business rules. The most simple example is that a target will not be scheduled if it is not visible. The QC is empowered to be flexible interpreting target constraints and can push limits for the benefit of better program completion rates or other pressures that require the scheduling of the observations outside constraint definitions.

##### 2.2.2.1 Long term scheduler

Semester-based scheduling is taken on after the Phase I period is completed. The executives and QSO team will plan out instrument queue runs knowing the requested telescope times of each program, and collectively for each instrument.

##### 2.2.2.2 Mid range scheduler

Before each queue run, the QSO team will provide a summary of goals that need to be accomplished. Information that is brought to attention includes programs that have hard limits on certain constraints, such as airmass, sky background, proximity to the moon, and image quality (IQ). Any scheduling
constraints that exist are also highlighted. It is unknown what tools are being used to assist the QSO team with this mid range planning. Program grade, rank, target visibility, and agency balance are the minimum calculations that need to be considered before proposing a queue run schedule.

2.2.2.3. Short term scheduler

Each day prior to an observing night, the Queue Coordinator (QC) will create a set of schedules that will act as guides for the night of operations. These schedules are permutations of well-known environmental constraints. For an example observing night, the following queues are created:

<table>
<thead>
<tr>
<th>Queue</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>&lt; 0.65” Seeing, Photometric sky conditions</td>
</tr>
<tr>
<td>Q2</td>
<td>0.65” &lt; 0.80” Seeing, Photometric sky conditions</td>
</tr>
<tr>
<td>Q3</td>
<td>0.65” &lt; 0.80” Seeing, Thin Cirrus acceptable</td>
</tr>
<tr>
<td>Q4</td>
<td>0.80” &lt; 1.0” Seeing, Photometric</td>
</tr>
<tr>
<td>Q5</td>
<td>0.80” &lt; 1.0” Seeing, Cirrus acceptable</td>
</tr>
<tr>
<td>Q6</td>
<td>all other conditions</td>
</tr>
</tbody>
</table>

Table 1 - A Night's set of queues

A collection of observations and their constraints as specified by the PI and additional instructions from the QC will exist in each queue. Additional system calibration queues can be created and are available for observing throughout the night. Like the long term scheduler, there are a minimum number of calculations that need to be performed to properly allocate queues: program grade, rank, target visibility, and agency balance. Additionally, circumstances exist that can force entries into these queues and at specific times that don't follow standard formulas.

2.2.2.4. Real time scheduler

The real time scheduler at CFHT is the observing staff. The observer will utilize the various environmental sensors along with knowledge of the previous observations to determine from what queue they select the next observations from. Outside of system failures and weather events, this event loop is happening all night long. With extenuating circumstances, the QC can become involved in the real time scheduler decision making. In every situation, the metrics calculated during all other scheduling phases should be again calculated here for proper decision making. In addition to program grade, rank, target visibility, and agency balance, the observer must calculate such items as moon proximity and the effects of the instrument configuration – like filter selection, focus, etc.
2.2.3. Data Grading and Validation

Real-time processing specific for each instrument provides a set of metrics for each data acquisition. These metrics combined with visualization of the acquisition allows the observer to appropriately grade data and determine if repeating or aborting subsets of these queues is necessary. The following morning, the QC will validate the data triggering the queue cascade of calculations. Exposures, OGs, and ultimately the programs, will get updated completion and validation statistics, thus affecting forward looking agency balances. A modified set of available observations is available for the next session of short term scheduling.

It must be noted that within this function, the QC has the power to interpret the data *fuzzily*. A classic example is the image that is 10 percent within its IQ or Sky background constraints. The program is near 100 percent completion. And the target will not be visible again during the queue run. The QC could decide to validate this exposure with its shortcomings in IQ and Sky background. In a related scenario, the QC would not validate the exposure after the observing night. But, during a future planning and scheduling session, the QC could validate this historical exposure.

2.3 Current Methods and Procedures

2.3.1. Systems

The software systems described below are the primary tools used in the QSO operating model. It is necessary to demonstrate all known software, data, and interactions for insight into the size of the combinatorial space of inputs, desired outputs, and their interactions. As these are user-operated tools, the scope of the tool and dataset can seem overwhelming especially to those not trained with skills to use and evaluate these systems.

2.3.1.1. Northstar/Phase I

Northstar is the Phase I tool employed by CFHT. Twice a year, astronomers/principal investigators (PI) around the world are invited to submit proposals for telescope time. They will login to Northstar and submit relevant information for their proposal. This information will include the instrument(s), instrument configuration, the targets of their observations, any constraints on those observations including scheduling constraints, and the amount of total telescope time they are requesting.

Northstar has been used since 2009. It is expected to be a long term solution to fulfill proposal submission for Phase I. It is operated exclusive of the Phase II tool and other down line QSO software. The use of Northstar to handle Target Of Opportunity (TOO) requests will require integration into the Phase II/QSO software systems and databases.

The following images are screen shots of the Northstar web application\(^2\). A few of the important pieces of data to assist with planning and scheduling are components of these images.

\(^2\) http://jcdbs1.cfht.hawaii.edu:8180/proposal
In Figure 1, two required data inputs are specified by the PI: the instrument(s) to be used, and the total amount of telescope time they are requesting. This data is important as it allows the long term planner to determine if it is possible to fit all the requested proposals within one semester. Studying the justification, including science and technical information and relating to the instruments of choice aids in calculating proposal suitability.

Figure 1 - Northstar Observing Request - Instrument, Total Time
Figure 2 shows what an instrument configuration can be defined as. In addition to filter selection, the PI specifies scheduling constraints and even a general seeing range. Predefined scheduling constraints typically are forced into queues during operations. They primarily exist in their position in the queue based on the requested schedule. Higher ranked programs can supersede their observation, but the influence of predefined schedule constraints seems great.
Within Northstar, the PI can specify additional target-specific constraints. A large set of information from Northstar does not get integrated into down line QSO software.

### 2.3.1.2. Phase II

The CFHT Phase II is used by the PI once their Phase 1 proposal has been accepted by CFHT. Within this tool, strict constraints and general instructions can be defined for their observations. These constraints are evaluated by the QSO team while scheduling observations. It is the central data entry tool linking the PI and the QSO team\(^3\).

---

The PI has a great deal of flexibility in setting up their observing program. They can define an unlimited number of constraints. The constraint definition at this level is specific to the instrument. Seeing, sky background and airmass conditions are typically specified here (Figure 4). These constraints are then associated to target observation definitions (Figure 5). At the Observation Group (OG) level, the PI ties together the instrument configuration, acquisition constraints, and the target. Notice the instructions in the comments of Figure 5 – scheduling criteria in a free form text field.

Additional web applications exist as part of the PH2 tool – the Program Viewer and Night Reports provide different perspectives on the data that exists within PH2 and QSO Tools.
2.3.1.3. QSO Tools

QSO Tools is the software used by the QSO Team to plan and schedule observations as well as perform, grade, and validate those observations. Figure 6 shows the Queue Wizard software providing the QC with a great number of combinations to build queues with. In Figure 8 – Figure 10, a sample result set from executing the Queue Wizard is provided. Three screen shots are necessary to highlight the scope of information that is considered in making these queues – and three screen shots are not enough to show all the information available. A typical semester can have many thousands of targets per instrument. As observations are validated and the semester wanes, the result set will be reduced accordingly. From this result set, the QC will match pending observations with the queues based on the observing constraints. The QC will work with some metric to filter, prioritize, and schedule observations for the night. Statistics for agency balances, instruments, and individual programs are considered in this metric (Figure 7). While queue-building, the QC can view a Night Graph (Figure 11), to see how the night can theoretically play out.
Figure 6 - Queue Wizard

Figure 7 - QSO Statistics
Figure 8 - Queue Wizard Results
Figure 9 - Queue Wizard Results

<table>
<thead>
<tr>
<th>Status</th>
<th>OB(s)</th>
<th>OB Status</th>
<th>OB I-Times</th>
<th>Target Name</th>
<th>Target RA</th>
<th>Target Dec</th>
</tr>
</thead>
<tbody>
<tr>
<td>1ATED</td>
<td>OB1</td>
<td>VALIDATED</td>
<td>5.724</td>
<td>RC5023158</td>
<td>02:21:58</td>
<td>-08:49:55</td>
</tr>
<tr>
<td>2ATED</td>
<td>OB2</td>
<td>VALIDATED</td>
<td>5.724</td>
<td>RC5023158</td>
<td>02:21:58</td>
<td>-08:49:55</td>
</tr>
<tr>
<td>3ATED</td>
<td>OB3</td>
<td>VALIDATED</td>
<td>5.724</td>
<td>RC5023158</td>
<td>02:21:58</td>
<td>-08:49:55</td>
</tr>
<tr>
<td>5ATED</td>
<td>OB5</td>
<td>VALIDATED</td>
<td>5.724</td>
<td>RC5023158</td>
<td>02:21:58</td>
<td>-08:49:55</td>
</tr>
<tr>
<td>6ATED</td>
<td>OB6</td>
<td>VALIDATED</td>
<td>5.724</td>
<td>RC5023158</td>
<td>02:21:58</td>
<td>-08:49:55</td>
</tr>
<tr>
<td>7ATED</td>
<td>OB7</td>
<td>VALIDATED</td>
<td>5.724</td>
<td>RC5023158</td>
<td>02:21:58</td>
<td>-08:49:55</td>
</tr>
<tr>
<td>8ATED</td>
<td>OB8</td>
<td>VALIDATED</td>
<td>5.724</td>
<td>RC5023158</td>
<td>02:21:58</td>
<td>-08:49:55</td>
</tr>
<tr>
<td>9ATED</td>
<td>OB9</td>
<td>VALIDATED</td>
<td>5.724</td>
<td>RC5023158</td>
<td>02:21:58</td>
<td>-08:49:55</td>
</tr>
<tr>
<td>10ATED</td>
<td>OB10</td>
<td>VALIDATED</td>
<td>5.724</td>
<td>RC5023158</td>
<td>02:21:58</td>
<td>-08:49:55</td>
</tr>
<tr>
<td>12ATED</td>
<td>OB12</td>
<td>VALIDATED</td>
<td>5.724</td>
<td>RC5023158</td>
<td>02:21:58</td>
<td>-08:49:55</td>
</tr>
<tr>
<td>16ATED</td>
<td>OB16</td>
<td>VALIDATED</td>
<td>5.724</td>
<td>RC5023158</td>
<td>02:21:58</td>
<td>-08:49:55</td>
</tr>
<tr>
<td>17ATED</td>
<td>OB17</td>
<td>VALIDATED</td>
<td>5.724</td>
<td>RC5023158</td>
<td>02:21:58</td>
<td>-08:49:55</td>
</tr>
<tr>
<td>18ATED</td>
<td>OB18</td>
<td>VALIDATED</td>
<td>5.724</td>
<td>RC5023158</td>
<td>02:21:58</td>
<td>-08:49:55</td>
</tr>
<tr>
<td>19ATED</td>
<td>OB19</td>
<td>VALIDATED</td>
<td>5.724</td>
<td>RC5023158</td>
<td>02:21:58</td>
<td>-08:49:55</td>
</tr>
<tr>
<td>20ATED</td>
<td>OB20</td>
<td>VALIDATED</td>
<td>5.724</td>
<td>RC5023158</td>
<td>02:21:58</td>
<td>-08:49:55</td>
</tr>
<tr>
<td>22ATED</td>
<td>OB22</td>
<td>VALIDATED</td>
<td>5.724</td>
<td>RC5023158</td>
<td>02:21:58</td>
<td>-08:49:55</td>
</tr>
</tbody>
</table>

Total I-Time: 20 hr, 37 min, 12 sec
<table>
<thead>
<tr>
<th>Needs Acc PT</th>
<th>Filter</th>
<th>ExpTime</th>
<th>OG - Time</th>
<th>OP</th>
<th>IQ</th>
<th>SkyBG</th>
<th>Needs Pilot?</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>D</td>
<td>307.0</td>
<td>4.60</td>
<td></td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>D</td>
<td>307.0</td>
<td>4.60</td>
<td></td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>D</td>
<td>307.0</td>
<td>4.60</td>
<td></td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>D</td>
<td>307.0</td>
<td>4.60</td>
<td></td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>D</td>
<td>307.0</td>
<td>4.60</td>
<td></td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>D</td>
<td>307.0</td>
<td>4.60</td>
<td></td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>D</td>
<td>307.0</td>
<td>4.60</td>
<td></td>
<td>8</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 10 - Queue Wizard Results
With queues available, and when the night is up, the observer takes over operations. Using the various queues, preceding observations, and real time environmental information, the observer will choose observations and send them through the remaining software chain. From the Observing Tool (Figure 12), the observer will select a queue to operate from based on the current environmental conditions. Once the queue has been selected (Figure 13), the observer will send the observations to be performed. The next bit of data actions will occur within the Director tool – this is discussed below. But, as data is acquired the observer will have visibility within the QSO Tools Logbook (Figure 14). For each acquired image, the observer is responsible for providing an initial grade and any relevant commentary.
<table>
<thead>
<tr>
<th>Name</th>
<th>Instrument</th>
<th>Date</th>
<th>Type</th>
<th>Comment</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q8</td>
<td>WIRCAM</td>
<td>2010-09-14</td>
<td>Obs</td>
<td>Liu OGS for the run</td>
<td>CREATED</td>
</tr>
<tr>
<td>Q9</td>
<td>WIRCAM</td>
<td>2010-09-14</td>
<td>Obs</td>
<td>Snapshots</td>
<td>CREATED</td>
</tr>
<tr>
<td>Q10</td>
<td>WIRCAM</td>
<td>2010-09-14</td>
<td>Obs</td>
<td>Q3 IQ &lt; 1.2&quot;</td>
<td>CREATED</td>
</tr>
<tr>
<td>Q11</td>
<td>WIRCAM</td>
<td>2010-09-14</td>
<td>Obs</td>
<td>Q3 IQ &lt; 0.8&quot;</td>
<td>CREATED</td>
</tr>
<tr>
<td>Q12</td>
<td>WIRCAM</td>
<td>2010-09-14</td>
<td>Obs</td>
<td>Q2 IQ &lt; 0.65&quot;</td>
<td>CREATED</td>
</tr>
<tr>
<td>Q13</td>
<td>WIRCAM</td>
<td>2010-09-14</td>
<td>Obs</td>
<td>Q1 IQ &lt; 0.8&quot;</td>
<td>CREATED</td>
</tr>
</tbody>
</table>

**Figure 12 - Observing Tool (Select queue)**

**Figure 13 - Observing Tool (Observe from a queue)**
At the end of the night, the QC will again take helm of QSO operations. All data acquired in the previous night must get their final grades and be validated (or not). Within the Logbook software, there is a validation tools (Figure 15). Additionally, all the night's environmental information is available to the QC from within QSO Tools or various web applications. Using real time data processing information (addressed below), environmental information (addressed below), observer's grades and comments, and sometimes pushing PI's original observing criteria, the QC will go through a rigorous and repetitious process of data validation. The workload varies throughout the semester and per instrument.
### Director

The Director software provides the main data communications bus for all the intra related software and instruments (scientific and environmental). Figure 10 shows the director console as it is acquiring Dark exposures for Wiracam. Notice some pertinent status information in bottom gray section.

**Figure 15 - Validation Tool**
2.3.1.5. Real time processing (Elixir, Upena, IIWI)

Upon data acquisition in QSO, real time processing systems evaluate the data providing information to the observer for real time image grading. This information can include the derived IQ and sky background. The data fed back into the QSO database from real-time processing is instrument specific.

<table>
<thead>
<tr>
<th>Instrument</th>
<th>Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEGACAM</td>
<td>IQ, Sky background</td>
</tr>
<tr>
<td>WIRCAM</td>
<td>IQ, Sky background, absorption</td>
</tr>
<tr>
<td>ESPaDOnS</td>
<td>?</td>
</tr>
</tbody>
</table>

The observer will grade the image accordingly. The QC will validate the image based on the database information which includes the the real time processing, environmental conditions, and the observer's grades and comments.

2.3.1.6. DIMM

CFHT maintains the operation of two DIMMs. The Mauna Kea Atmosphere Monitor (MKAM) provides MASS and DIMM data. The Dome DIMM is a seeing monitor operated from inside the CFHT dome. It provides real time seeing measurements as affected by the CFT dome itself. With the data collected from this instrument since 2009, formulas have been devised that can provide real time seeing estimates anywhere in the sky from the CFHT dome. These formulas use meteo data such as wind speed, direction, and various temperatures that are all available in real time.4 In Figure 17, the

---

4 “CFHT DIMM Data Analysis”, Report to the SAC 09/2009, Racine
data from both DIMMs for one night of observations is presented.

![Figure 17 - DIMM data](image)

### 2.3.1.7. Skyprobe

Skyprobe is an instrument that measures atmospheric attenuation. Photometricity can be derived using data collected and information calculated by Skyprobe. Skyprobe uses the field the CFHT telescope is pointing at and information is useful in determining whether science observations are taken under photometric conditions. This determination is useful for grading acquired data and queue selection.

An example night of non-photometric conditions is presented in Figure 18. If there are operating conditions that affect skyprobe data, they should be addressed. Hearsay describes dome and telescope motion as a debaser of Skyprobe's data.

---

The Cloudsensor is a commercial device from the Canadian company Boltwood. A micro bolometer with a 90° FOV, it operates continuously even under harsh winter conditions. It can give generic

2.3.1.8. Cloudsensor

The Cloudsensor is a commercial device from the Canadian company Boltwood. A micro bolometer with a 90° FOV, it operates continuously even under harsh winter conditions. It can give generic
cloudy/no cloud indications for the area it targets. It is not useful for cirrus cloud detection.

2.3.1.9. MKAM Weather Tower

Environmental conditions are monitored continuously at the observatory. Certain components of the meteo data set are fundamental in real time scheduling considerations. Wind speed, wind direction, and temperature measurements from the MKAM weather tower and inside the dome should be continuously available.
2.3.2. Opportunities

There are no indications that the current QSO operating model is not sound. A somewhat large internal infrastructure exists to provide external and internal support, perform the scheduling tasks, and operate the telescope and supporting instruments. As the observatory moves into a remote observing model, a
great number of hardware and software tools are being developed to assist with handling short term and real time observation scheduling, handling telescope operations, site monitoring, and data grading. These modifications can only improve on the standards that CFHT has established as a productive observatory operating in QSO mode. And CFHT can probably operate throughout the remainder of its life in a human-controlled QSO mode.

With a long successful history of QSO operations, a great deal of knowledge has been acquired and applied as the business logic. Unfortunately, a large part of this knowledge remains outside of the computing systems. If the logic exists in software, the software remains under operator control – providing verbatim sets of information to work with. QSO team members must interpret the results and decide on actions to take. Arduous training program is used to educate new QSO team members of proper strategies to employ while performing their job functions. For existing team members, regular evaluations of their past performance is continuous – (emphasis on points of improvement.)

The opportunity exists to incorporate as many business rules as possible into the computing systems. Errors can be minimized, each job function can be performed more consistently, and a smaller set of the operating model components is individually experted. “A major challenge in operating in queue mode is the decision making process during an observing night.” This statement easily expands on time, from the now, to the observing night, to the queue run, to the semester.

3. Proposed Methods and Procedures

3.1 Summary of Improvements

3.1.1 Planning and Scheduling

If the data entry applications and databases can be modified to allow relational storage of all observation constraints from the PIs and any fuzzy enhancements from the observing staff, the QSO operational model with phases of scheduling can be morphed into one real-time scheduler. The best observation for the now can be determined from the whole collection of observing targets and constraints, matched with the moment's environmental conditions, in real time. The need for observing queues based on permutations of priorities and environmental conditions is nonexistent. The fundamental change proposed herein is automated real-time scheduling.

Figure 21 shows how at any given moment it is possible to select the best observation to perform. Outside some of the missing environmental conditions tools, the QC and observers use this methodology regularly. The targets and constraints database can number in the thousands. For semester 2010B, over 600 ESPaDOnS, 900 MEGACAM, and 2600 WIRCAM observations have been defined by the PIs. For each observing moment, the observer is required to go through a permutation set based on these numbers, filter the observations against current environmental conditions, and then select the best observation based on the magical best-fit metric. It should be noted that a current operational benefit of QSO is that the QC has greatly reduced and effectively categorized the working set of targets and constraints to manageable numbers.

6 The Queued Service Observing Project at CFHT, Section 6.3, SPIE 2002

Automated Service Observing September 23, 2010 Page 25
There is historical precedence for implementation of automated scheduling software in business and industry. Within astronomy, in particular observatories, the Space Telescope Science Institute developed and successfully used SPIKE for intelligent scheduling of the Hubble Space Telescope\(^7\). A stochastic heuristic search is employed using suitability functions to solve their constraint satisfaction problem. In the early 1990s, CFHT evaluated using SPIKE as an automated long term scheduler for classical observing. Currently, SPIKE is being evaluated at CFHT for applicability as a long and short term scheduler. SOFIA is an airborne astronomy observatory. The scheduling and operational constraints to be used in developing an automated scheduling system included affects of being flight-based. Again a heuristic search is employed to perform the task. The specific algorithm in use is the Squeaky Wheel Optimization local search.\(^8\) Both the SPIKE and SOFIA solutions are recursive algorithms that prune large permutation sets based on constraint satisfaction and self-correct over use and time quickly generating acceptable, but quite possibly not the best, schedules.

A significant amount of knowledge engineering is still required for understanding of the QSO operational model. From this exercise, the development of constraint satisfaction or mismatch business rules can be developed. System and data deficiencies can also be identified and remedied. Ultimately, the knowledge engineering will enable the construction of momentary and evolved suitability functions. The suitability or best-fit function is undetermined although many of the inputs and desired outputs are currently known. The evaluation of algorithmic solutions should further consider, and in much more detail, these aforementioned heuristic searches, the development of genetic algorithms (GA) and/or Markov decision processes (MDP) as theoretical solutions to real time scheduling.

“GAs are well suited to solving scheduling problems, because unlike simpler heuristic methods genetic algorithms operate on a population of solutions rather than a single solution.”\(^9\) In the case of CFHT's observation scheduling problem, there are many solutions what need optimizing – agency balance, program completion statistics, data validation and loss statistics, shutter and idle time statistics. Additionally fuzzy matching of observing constraints and environmental conditions affect the large population of solutions. Section 5.1.1 provides more information on GAs.

“Markov decision processes, named after Andrey Markov, provide a mathematical framework for modeling decision-making in situations where outcomes are partly random and partly under the control of a decision maker.”\(^10\) Additionally, the ability to work better with large state spaces compared to other stochastic techniques is of particular interest. Upon completion of the knowledge engineering, the size of the state space will be determined, allowing refinement in the selection of the automated scheduling algorithms. Section 5.1.2 provides more information on MDPs.

Brute force computing techniques can also be employed as automated scheduling solution. But, the result of a demonstrative effort showed the shortcomings of using such techniques due to the size of the population of targets, constraints, and acceptable solutions.

---

\(^7\) SPIKE: Intelligent Scheduling of Hubble Space Telescope Observations

\(^8\) SOFIA's Challenge: Automated Scheduling of Airborne Astronomy Observations


Computing processes that will generate some of the required inputs for any automated scheduler system are depicted as 'predefined processes' in the flowchart of Figure 21. The process defined as 'select best-fit observation' will be executed using the AI algorithm selected to perform the ultimate function of selecting the best observation for any particular moment. Feeding this process will be a process that creates the randomized population set of targets and their constraints. This particular function is complete, and is used in today's functionality. Quite simply, the process is an extraction of data from the database that meets static criteria. More simple database extractions and calculations will comprise the predefined processes that 'get operational statistics'. The 'all sky seeing' process will provide estimations of seeing anywhere in the sky based on various temperature and wind values, and knowledge of the current seeing at a particular location in the sky. Determining 'all sky photometry' will require the deployment of scientific instruments currently in development, ASIVA, along with Skyprobe, and the possible use of more all sky cameras.

An initial population set of thousands of targets and their constraints can be reduced effectively by solving the non-random constraint satisfaction problems. Environmental conditions such as moon-related data, calculated photometric conditions, and calculated seeing conditions will prune the population to those a smaller finite set of potential observations. Additional pruning of the population will occur with another level of constraint satisfaction – hard or soft scheduling, must-do status, etc. Once the statically determined working population set is determined, the AI processes can work through the permutations of using the observations. A large tree of results based on the selection of an
observation at a particular moment impacts the weighted statistics of agency balances and program completion and validation rates. Evaluating the tree's branches will again prune the population set as conflicts with future constraints, or not meeting expected results of the operating business model eliminate candidates. Increased computing times based on traversing permutations on depths of search, contribute to the best selected observation at a particular moment. It is necessary to develop suitability and fitness functions for each traveled level of search satisfaction to operate on the whole populate set rather than the real-time observation.

At any moment in time, the initial working state or node in the solution tree can look like the diagram in Figure 22.

![Figure 22 - A Working State](image)

A large population of available observations and static constraints is the working set. Environmental conditions, constraint satisfaction, and operational overrides (saying something has to happen at a certain time regardless of the rules – in essence a constraint) are used to prune the population to a suitable working set. Iterating on this new population, fitness values based on affected agency balances and operating statistics (validation rates, program completion rates effectively the current Queue Efficiency statistic) can be calculated for each observation (see Figure 23). Continuing in a what if manner by depth filling the observation schedule from this new working state, any block of time can have a suitability function calculated, backward affecting the fitness value of the top node in the tree. This block of time can be the now moment, a night, an instrument observing run, or an entire semester – as the standard examples. In Figure 24, the makeup of each node is the content depicted in Figure 22. As time increases, the available and pruned observation sets will decrease in size accordingly. Each now state becomes (now + Σitime) with the summation running over the depth of the built observation tree.

It is necessary to develop and model the effectiveness of the algorithms proposed to determine the best performing solution as a function of computing time to create acceptable schedules at anytime for any time frame. Knowledge engineering on the operating observation model must yield the suitability and fitness functions before the most acceptable solution can be determined. Deriving these functions will highlight development requirements in all the supporting software and computing systems. It does not seem that the numeric load of the system will be prohibitive to any solution. Limited modeling has shown that the working population sizes are all quickly processed with today’s computing power. An unoptimized breadth-first approach has yielded the ability to select and prune 2000 possible observations to ~10% of its original size and further determine the most suitable observation from this set in a very rough seven seconds (the database is slow, and better metric points should be established).
3.1.2. Data Evaluation

During and post acquisition, the observer is responsible to verify the data. Going through the verification process will determine if system failures exist, how the data should be graded, image quality, and any environmental conditions that might have contributed to the grading. If constraint limits were pushed, awareness to this is also recorded. It is possible that environmental conditions will change during the observations, rendering the data useless. In this scenario, the observation can be
aborted and possibly retried. It could be necessary to find new observations that will meet the current environmental conditions. The flowchart for monitoring data acquisition is shown in Figure 25. For the constraint violation check, fuzzy conditions as specified by the PI or observing staff should come in play.

Grading and validating the exposure involves using real-time processing metrics and environmental conditions during the exposure along with looking at the actual data. CFHT has acquired hundreds of thousands of images and their associated evaluations and observing conditions. This data along with the original observing constraints is a great training set for teaching an image classifying neural net to
be able to grade and validate future data.

In QSO, grading and validation are separate processes in the operating model. Grading is done by the observer after an observation has been acquired. Validation happens after the observing night, and is performed by the QC. It seems the fundamental criteria used to validate an exposure is also available at the time the data is graded. In ASO, grading and validating can happen in real time, upon the completion of the data acquisition. The monitoring flowchart can be extended to show that another round of constraint satisfaction based on real-time processing information can provide the necessary input to assist in automated validation (or not) of data (see Figure 26). In addition to the post-acquisition processing for automated grading and validation, the chart shows some *fuzzy enhancements* to the event loop. If the PI or the observing staff have indicated that constraints can be pushed beyond their limits, this modified event loop indicates a processing step to handle that.

In QSO, beyond constraint satisfaction, there exists some *magic* to validating data that requires a human to perform that duty. Knowledge engineering on this business practice will yield the fundamental logic and rules to this magic. It is unknown whether or not simple boolean logic is enough for a computing system to perform data grading and validation. Maybe there is a need for an AI-based software system. Neural nets have already been applied to the problems of star and galaxy detection. Sextractor is a well-known astronomy software application that utilizes neural nets. The knowledge engineering is incomplete for this business function. A future addendum to this document can include refined suggestions for deploying computing systems that can perform the function of data grading and validation. For the ASO model to be complete, it is a process that is required to be automated.
Figure 26 - Data acquisition flow chart
3.2 Summary of Impacts

The development of ASO can happen exclusive of all other operational development. Its development goals do not conflict with the QSO or OAP projects. The result of unit development, hardware and software, required to operate under ASO contributes to more effective and efficient QSO and RO operations. The development of ASO applies only to the business components currently carried on by QSO. The development cycle can also be constructed so that the software units necessary for ASO actually reproduce existing QSO functions without altering the operating procedures. For each business process that exists, the automated systems do core work, but the human operator of those functions is still responsible for the execution. It is anticipated that changes will be implemented in Phase I/Northstar, Phase II, QSO Tools, and other NEO software.

3.3 Assumptions and Constraints

There are other functional QSO processes that have not been addressed within this document. The scheduling, acquisition, and grading of calibration is an example of a function that needs to be researched more thoroughly and folded into ASO.

4. Detailed Characteristics

A great deal of knowledge engineering must be completed to gather all the requirements for a formal design and development program to be proposed. The result of knowledge engineering must:

1. Identify all known observational constraint databases – those not integrated into QSO systems.
2. Recommend the changes to Phase I and Phase II tools that can incorporate the greatest set of constraints including fuzziness of those constraints without too negatively affecting the flexibility of the existing system.
3. Like the existing observational definitions, the maintenance of these new and modified constraints should be left to the burden of the PIs.
4. Formulate the all sky seeing estimator.
5. Formulate the all sky photometricity calculator.
6. Formulate target pressure.
7. Formulate fuzzy constraint satisfaction.
8. Formulate best-fit functions for observations with various time intervals.
9. Identify all required human overrides.
10. Provide a mechanism for constraint definitions of human overrides.
11. Formulate data verification, grading, and validation.
5. Design Considerations

Algorithms and techniques mentioned have been so because of historical use and reference. Case studies, software examples, and development tools exist for any algorithm or technique mentioned. The largest amount of focus of this documentation has been on scheduling systems. More study is needed on AI algorithms and techniques that apply to data evaluation, in particular image processing.

5.1 Scheduling

5.1.1. Genetic Algorithm (GA)

Encoded in genes of an organism are the rules that define its growth and interaction with the environment. Connections of genes are called chromosomes. Each gene represents a specific trait of the organism. The process of recombination is creating a new organism by passing genes from two selected different organisms. The new organism will inherit half of its genes from each of the two selected parent organisms. Occasionally, organisms that are the result of recombination have genes that are mutated. In Darwin's evolution, the selection of organisms for recombination is done by fitness – survival of the fittest. This process continues until a termination condition occurs.

From Wikipedia, and injecting scheduling terminology, a simple generational GA:

1. Choose the initial population of randomized schedules based on now
2. Evaluate the fitness of each schedule based on now
3. Repeat on this generation until termination: (time limit, sufficient fitness achieved, etc.)
   1. Select the two best-fit schedules for recombination
   2. Breed new schedules through crossover and mutation operations to give birth to offspring. In the case of scheduling, crossover may not be possible because the scheduling problem requires ordered chromosomes.
   3. Evaluate the individual fitness of the new schedules
   4. Replace least-fit population with new schedules

A possible way to look at the CFHT real time scheduling problem (RTSP) is as the Traveling Salesperson Problem (TSP). Solving the TSP is a classic exercise for all computer scientists. The TSP problem is this: Find the cheapest way to travel to every city on a list. To similarly represent CFHT's scheduling problem: Find the most efficient way to schedule an observing period with an acceptable list of observations from a database. In the TSP, the genome is represented as ordered list of cities visited. In the RTSP, the genome is similar, an ordered list of observations and their costs. For recombination, take a set of N-schedules, swap the next observations from each in the selected set, calculate the new schedules' fitness, and replace previous least-fit schedules with the new schedules. Repeat this process until the best or acceptable fitness of the next observation in the schedule has been determined, and repeat the process with the next observations. Once an entire scheduling period has been filled, and the best-fit schedules have been determined a simple selection process can be done to
select use the generated schedule.

A more analogous way to address RTSP is as the Knapsack problem. In the TSP, a finite number of cities must be visited at the lowest cost. In the Knapsack problem, there are a finite number objects that can be packed, but there is a limit to what the knapsack can hold, and there are invidivual values to the items being packed. Solving the problem requires maximizing the value of the items in the knapsack. Injecting CFHT scheduling terminology, a knapsack could be a scheduling night. The limit on this night is the amount of time available. The items we are looking to pack are observations. Each of these observations have a value (the fitness value impacting agency balance, program completion rates, etc.) The observations weight can be the time required to make this observation. This is a very simple model and not quite the problem. RTSP has multiple constraints. Fortunately, research to solving this multiconstraint Knapsack problem offers some solutions. Brute-force and dynamic programming techniques including the use of GAs\(^{11}\) have been used to solve the Knapsack problem.

Building schedules and populations of schedules in theory becomes very expensive if the size of the pruned observations set is not effectively reduced through the constraint satisfaction process. There are best-fit considerations at the observation level for moments in time, as well as the described GA's best-fit considerations for completed schedules candidates. Randomized brute-force traversal through the set of observations and incremental best-fit filtering will create the initial population of schedules to work against. The scheduling period can be a any period of time: semester, instrument run, observing night, or simply the next available period.

### 5.1.2. Markov Decision Process (MDP)

A Markov Decision Process provides a model for the decision making process that considers the immediate and long term effects of its action. “If you can model the problem as an MDP, then there are a number of algorithms that will allow you to automatically solve the decision problem”\(^{12}\). Modeling of the MDP requires defining these components: states, actions, transitions, and immediate rewards.

#### 5.1.2.1. States

What is the current state of scheduling? Statistics for agency balances, program completion and validation rates, and observing efficiency can be used to define this state. How does performing an action affect the statistics? Business rules dictate how we want actions to affect statistics – better QE, better agency balance, observe as much as possible. A set of states is every possible condition that could exist at any moment for any scheduling period.

#### 5.1.2.2. Actions

The action is the selection of an observation for a scheduling period. What observations are acceptable for the state of this scheduling period?


5.1.2.3. Transitions

The action of selecting an observation for a particular scheduling period will immediately impact the state defined as the statistics for observing efficiency and agency balances. These are the transitions.

5.1.2.4. Immediate rewards

The value of performing an action must be determined so that a comparator can be derived. This would be similar to having suitability and fitness functions in other algorithms.

5.1.2.5. Solving the MDP

To solve the MDP, policies must be defined as the best action for each of the states. For agency balance, a policy will be specified so that the selection of an observation will best keep the agency statistics as balanced as possible (without consideration of the other states). Maybe a policy will exist to prioritize the observations of a program that is near 100% completion (without consideration of the other states). Such policies are actions defined for various states. Policies can be derived from value functions that assign numerical computations to states. It is the trick to solve within MDPs as fitness functions are in GAs.

The theory and applicability of MDPs is still under active learning and investigation. No prototyping of MDPs has occurred yet.

5.2 Data monitoring, verification, validation

5.2.1. Neural Nets for Image Classification, Pattern Recognition

A neural network (NN) is an artificial intelligence concept that attempts to model how the brain processes information and learns. A large number of interconnected processing elements called neurons work together to solve specific problems. Like people, neural nets learn by example, adjusting synaptic connections tying neurons together. Neural networks, can be used to identify patterns and detect trends that are too complex or too subtle to be noticed by either humans or other computer techniques. A trained neural network is an expert in its field.

An important application of neural networks is pattern recognition. Pattern recognition can be implemented by using a trained neural network. After the network has been trained to associate outputs with input patterns, the network can identify input patterns and its associated output. If the input pattern has no known associated output, the resulting output will be the output associated to the input pattern with the least difference to the given pattern.

There are various NN algorithms that can be used to tackle the same problems. More investigation into these techniques is necessary before development commitment. These AI concepts will be applicable to solving all sky photometric determination and possibly real-time observational verification and validation. Specific AI techniques being considered are back propagation NN, K-nearest neighborhood (KNN), self organizing maps (SOM), and Bayesian classifier variants.
Backpropagation is a supervised learning method that trains feed forward neural networks. The problem solving technique is best applied when:

- It is difficult to derive relationships between large amounts of input and output data.
- There is a clear solution that cannot be solved by traditional flowchart software.
- There are known acceptable solutions that be used as training material.
- Outputs can be "fuzzy", or non-numeric.

Pattern recognition is a problem that is associated with these multilayer back propagation neural network solutions. It has been shown to be useful in optical character recognition (OCR) applications.

The k-nearest neighbor algorithm is amongst the simplest of all machine learning algorithms: an object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its k nearest neighbors.\(^\text{13}\)

A Self-Organizing Map (SOM), or Kohonen network, is a NN that utilizes unsupervised learning to produce neighbor-influence maps of training data.

Bayesian techniques are currently being used in computing environments as SPAM email classifiers.

6. Environment

6.1 Equipment Requirements

The current set of functional environmental monitors must be operational and provide real time data that will be readily available for the AI systems to use in their processing. Real time sky and cloud monitoring must be made possible through instruments such as the DIMM, Skyprobe, Cloudsensor, and ASIVA. Meteo information must be available in real time to assist with estimation functions for all sky seeing and photometricity estimates.

6.1.1. ASIVA

When deployed, and if dependable, the ASIVA instrument will provide CFHT with knowledge of sky photometricity. As an all sky camera, the instrument ultimately should provide per pixel evaluation of cloud conditions\(^\text{14}\). Per pixel cloud condition determination is not currently available or required functionality from the ASIVA project. Artificial intelligence techniques utilizing neural nets and self organizing maps is proposed to gain this functionality. This functionality is a requirement for ASO operations.

6.2 Software Environment

The CFHT operating software environment does not drastically change. All software can be developed with the existing infrastructure in mind. The operating system, development languages, and software


subsystems in use such as real time processing, environmental data monitors, and databases are not affected except by slight alterations for more or finer data.

6.3 Summary of Impacts

There are obvious organizational and therefore operational impacts to automating job functions commanded by humans. At CFHT, the QSO team is one of the largest organizations in the company. At the time QSO ends and ASO begins, there is no QSO team. An ASO group will not have the same human resource needs as was required with QSO.

The development cycle can be very long. The techniques, even with historical precedence, are experimental and novel. The CFHT environment has distinct features that might not fit as well into these solutions. The scope of data inputs and complexity of desired outputs could be prohibitive to the presented algorithms.

6.4 Failure Contingencies

All required subsystems contributing to the ASO business model must exhibit great levels of reliability and offer redundancy. Any failure, causes a complete breakdown of the operating model. A development requirement of allowing operator intervention at all processing nodes will allow for reverting to an altered task list of manual QSO operations.

7. Conclusion

Evolving CFHT into Automated Service Observing is a great but doable challenge Manual operations over many years provides a knowledge base and rule set for standardized observing practices within CFHT. Variability in the results of all business nodes occurs because of human control operations and lack of computing systems to minimize that variability. The science fiction of artificial intelligence lends certain techniques to allow for machine control of certain aspects of operations. Is the CFHT observation event loop so well defined that we can train machines to do the job? At this point in the research, indications are yes. The executives and managers of CFHT, with knowledge of operational and development cost metrics can determine if a long term, primarily software development project with great organizational and operational implications is worth executing.

8. Appendices

8.1 Appendix 1, output of unoptimized, one-generation GA

The following output shows system execution and results of automated short term observation scheduling. The fitness functions do not consider real time environmental conditions, instead selecting observations based on early implementations of target pressure, agency balance, and observing efficiency fitness functions.

Initial population set:

Available targets: 1809

10BP22(A,2), OG585(Medium, OB585/1OB(1)), F 1.531140
10BP22(A,2), OG586(Medium, OB586/1OB(1)), F 1.529977
10BP22(A,2), OG852(Medium, OB852/1OB(1)), F 1.586756
10BC26(A,2), OG7(Medium, OB6/MOB(1)), FT2 1.132189
10BH08(B,3), OG77(Medium, OB6/1OB(1)), FT6 16.328113
10BH17(A,2), OG4(Medium, OB4/MOB(1)), FT4 4.491475
10BH17(A,2), OG12(Medium, OB12/MOB(1)), FT1 -1.196004
10BH17(A,2), OG13(Medium, OB13/MOB(1)), FT1 -2.358066
10BH17(A,2), OG15(Medium, OB15/MOB(1)), FT1 2.157614
10BH17(A,2), OG16(Medium, OB17/MOB(1)), FT1 -1.257407
...

(There is no need to reproduce ~2000 observations in this documentation)

... 

10BH94(B,5), OG1(Medium, OB1/1OB(1)), FT1 -1.444320
10BH94(B,5), OG2(Medium, OB2/1OB(1)), FT2 -1.443831
10BH92(C,1), OG1(High, OB1/1OB(1)), FT1 -1.434618
10BH92(C,1), OG2(High, OB2/1OB(1)), FT2 -1.454215
10BH92(C,1), OG3(High, OB3/1OB(1)), FT3 -1.434346
10BH92(C,1), OG4(Low, OB4/1OB(1)), FT4 -1.453937
10BH92(C,1), OG5(Medium, OB5/1OB(1)), FT5 -1.434135
10BH92(C,1), OG6(Low, OB6/1OB(1)), FT6 -1.453720
10BH54(B,1), OG3(Medium, OB3/1OB(1)), ET1 -47.295915
10BH54(B,1), OG2(High, OB2/MOB(1)), ET2 -4.110259
10BH54(B,1), OG1(High, OB1/MOB(1)), ET3 -2.314451

Pruned population set, constraint satisfaction

Targets: 188
<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10BP22(A,2),OG585(Medium,OB585/1OB(1)),F, 22 00 31.29 +01 54 30.9 0.650000 0.000000 0.200000 0.000000 170.000000 0.000000 -8.21533 0.000000 21.13594 60.409357 -7.76812</td>
</tr>
<tr>
<td>2</td>
<td>10BP22(A,2),OG852(Medium,OB852/1OB(1)),F, 22 10 51.28 +03 31 28.1 0.650000 0.000000 0.200000 0.000000 170.000000 1.000000 -8.45614 0.000000 21.13466 60.409357 -6.75882</td>
</tr>
<tr>
<td>3</td>
<td>10BP22(A,2),OG1149(Medium,OB1149/1OB(1)),F, 22 10 50.96 +04 27 26.8 0.650000 0.000000 0.200000 0.000000 170.000000 1.000000 -8.47143 0.000000 21.13594 60.409357 -6.74130</td>
</tr>
<tr>
<td>4</td>
<td>10BP22(A,2),OG861(Medium,OB861/1OB(1)),F, 22 10 50.96 +04 07 00.0 0.650000 0.000000 0.200000 0.000000 170.000000 1.000000 -8.46586 0.000000 21.13159 60.409357 -6.73572</td>
</tr>
<tr>
<td>5</td>
<td>10BP22(A,2),OG1152(Medium,OB1152/1OB(1)), 22 10 50.96 +04 36 33.2 0.650000 0.000000 0.200000 0.000000 170.000000 1.000000 -8.46024 0.000000 21.12739 60.409357 -6.73018</td>
</tr>
<tr>
<td>6</td>
<td>10BP22(A,2),OG864(Medium,OB864/1OB(1)),F, 22 10 50.96 +04 07 00.0 0.650000 0.000000 0.200000 0.000000 170.000000 1.000000 -8.46586 0.000000 21.13159 60.409357 -6.73572</td>
</tr>
<tr>
<td>7</td>
<td>10BP22(A,2),OG1155(Medium,OB1155/1OB(1)), 22 10 50.96 +04 46 33.2 0.650000 0.000000 0.200000 0.000000 170.000000 1.000000 -8.46024 0.000000 21.12739 60.409357 -6.73018</td>
</tr>
<tr>
<td>8</td>
<td>10BP22(A,2),OG867(Medium,OB867/1OB(1)),F, 22 10 50.96 +04 07 00.0 0.650000 0.000000 0.200000 0.000000 170.000000 1.000000 -8.46586 0.000000 21.13159 60.409357 -6.73572</td>
</tr>
<tr>
<td>9</td>
<td>10BP22(A,2),OG1140(Medium,OB1140/1OB(1)), 22 10 50.96 +04 31 28.1 0.650000 0.000000 0.200000 0.000000 170.000000 1.000000 -8.46586 0.000000 21.12739 60.409357 -6.73018</td>
</tr>
<tr>
<td>10</td>
<td>10BP22(A,2),OG1150(Medium,OB1150/1OB(1)), 22 09 29.17 +04 27 26.8 0.650000 0.000000 0.200000 0.000000 170.000000 1.000000 -8.45571 0.000000 21.13790 60.409357 -6.72558</td>
</tr>
</tbody>
</table>

(Treehuggery requires truncation of the results.)
The suitability function defined for a moment's selection of an observation in this case selects item 0 as
the observation to be executed now. The process loop will increment the starting time as the last time +
time of the selected observation. The search will be repeated using the new constraint on time. This
procedure is used to build a complete night's schedule. The statistics displayed with each observation
are various calculations for constraint satisfaction, target pressure, and observing efficiency impact.
They are very elementary and possibly invalid. The point of the exercise is to show that automation of
the scheduling is possible, and fulfilling development requirements will produce the applicable formulas.

8.2 Appendix 2 – Output of simple Neural network to automatically validate exposures

An attempt was made to develop a simple back propagation neural network to do automatic validation
using a portion of the validation history of Megacam. Four inputs consisting of requested IQ limit,
requested airmass limit, real time processed IQ, and actual airmass were defined. With each input
pattern, a known output is 1 for Validated, -1 for not. At this point, the NN training error is not
converging to 0. Therefore, the derived NN output for patterns post-training are in a very strange state.
This could very well be an indication that the delta's derived from this input set are not enough
information to create an appropriate NN. The selection of a back propagation NN could also be
incorrect. The training set needs careful consideration, and there is a significant amount of looseness in
the test set for this example; refinement to a set of validated data that has no dramatic fuzziness could
help. Additionally, remodeling the training set to be binned into states can improve the results. There
are a few other NN algorithms that can be tried to solve this problem – and more time is needed to
implement and evaluate their suitability.

Weight = 0.042964
Weight = 0.049784
Weight = 0.091770
Weight = 0.084869

initialising data

cycle = 0 RMS Error = 0.823090

cycle = 1 RMS Error = 0.825673

cycle = 2 RMS Error = 0.842168

cycle = 3 RMS Error = 0.818626

cycle = 4 RMS Error = 0.820444

cycle = 5 RMS Error = 0.84482
cycle = 6 RMS Error = 0.812117

... (to 500 training cycles)

... cycle = 491 RMS Error = 0.819472
cycle = 492 RMS Error = 0.822018
cycle = 493 RMS Error = 0.828070
cycle = 494 RMS Error = 0.839052
cycle = 495 RMS Error = 0.845383
cycle = 496 RMS Error = 0.839001
cycle = 497 RMS Error = 0.820267
cycle = 498 RMS Error = 0.846761
cycle = 499 RMS Error = 0.822038
cycle = 500 RMS Error = 0.826505

If the NN is properly implemented, the training cycles will influence the RMS Error to eventually approach zero. This NN is modeled incorrectly. The incorrect learning forces known data to be interpreted incorrectly – can’t validate as no outputs from the NN approach -1 or 1.

Reevaluating the training patterns to see how the neural net treats these inputs as new:

pattern 1 actual = 1 neural model = 0.476667
pattern 2 actual = 1 neural model = 0.476950
pattern 3 actual = 1 neural model = 0.476744
pattern 4 actual = 1 neural model = 0.472021
pattern 5 actual = 1 neural model = 0.420438
pattern 6 actual = 1 neural model = 0.455696
pattern 7 actual = 1 neural model = 0.469107
pattern 8 actual = 1 neural model = 0.474928
pattern 9 actual = 1 neural model = 0.474901
pattern 10 actual = 1 neural model = 0.475014
...
pattern 43174 actual = -1 neural model = 0.477160
pattern 43175 actual = -1 neural model = 0.477160
pattern 43176 actual = -1 neural model = 0.477160
pattern 43177 actual = -1 neural model = 0.477160
8.3 Appendix 3 – Output of K-nearest neighbor technique applied to data validation

Using a K-nearest neighbor technique, automated validation of observations with very loose rules is proven. Using a very small subset of known observations and their validation status, it is possible train the KNN system to validate subsequent observations.

Table 2 shows the training data. A set of some the permutations of conditions are indicated. (There are many more conditions and rules, and what-if's to incorporate.) Once the KNN has been trained, we can query the system using new observations and its related constraints.

An obvious test is to query the system when all conditions have been met.

Observation Constraints:
Airmass = Met, Sky BG = Met, IQ = Met, Photometric = Yes

The KNN will retrieve the 3 closest neighbors to the queried data set – Table 3. Determining the maximum condition of those three neighbors is VALIDATED.

The next obvious test to make is making sure bad conditional matches will result in no validation.

Observation Constraints:
Airmass = +20%, Sky BG = +10%, IQ = +.05%, Photometric = No

The result is the closest neighbors all have UNVALIDATED results, the maximum condition of those are UNVALIDATED.
### Table 2 - Training data

<table>
<thead>
<tr>
<th>Observation</th>
<th>Airmass</th>
<th>Sky BG</th>
<th>IQ</th>
<th>Photometric</th>
<th>Validate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>Conditions Met</td>
<td>+ 5%</td>
<td>Conditions Met</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>3</td>
<td>Conditions Met</td>
<td>+ 10%</td>
<td>Conditions Met</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>4</td>
<td>Conditions Met</td>
<td>+ 5%</td>
<td>Conditions Met</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>5</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>+ 0.5%</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>6</td>
<td>Conditions Met</td>
<td>+ 10%</td>
<td>Conditions Met</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>7</td>
<td>Conditions Met</td>
<td>+ 10%</td>
<td>Conditions Met</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>8</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>+ 0.5%</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>9</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>+ 0.5%</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>10</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>+ 0.5%</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>11</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>+ 0.5%</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>12</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>+ 0.5%</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>13</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>+ 0.5%</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>14</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>+ 0.5%</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>15</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>+ 0.5%</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>16</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>+ 0.5%</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>17</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>+ 0.5%</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>18</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>+ 0.5%</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>19</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>+ 0.5%</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>20</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>+ 0.5%</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>21</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>+ 0.5%</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

### Table 3 - Validated Match

<table>
<thead>
<tr>
<th>Observation</th>
<th>Airmass</th>
<th>Sky BG</th>
<th>IQ</th>
<th>Photometric</th>
<th>Validate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>3</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>4</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>5</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>6</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>7</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>8</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>9</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>10</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>11</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>12</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>13</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>14</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>15</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>16</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>17</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>18</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>19</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>20</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>21</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>
8.4 Appendix 4 – Examples of known Queue Observing Constraints that should be folded into PH1/PH2/QSOTools

The following examples are cases of data and software that exist outside software driven by the QSO databases. This is probably compounded by personal communications, like emails, that could be harboring required scheduling and constraint information.

<table>
<thead>
<tr>
<th>Observation</th>
<th>Airmass</th>
<th>Sky BG</th>
<th>I0</th>
<th>Photometric</th>
<th>Validate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Observation 1</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Observation 2</td>
<td>Conditions Met</td>
<td>+5%</td>
<td>Conditions Met</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Observation 3</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>+0.05%</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Observation 4</td>
<td>Conditions Met</td>
<td>+10%</td>
<td>Conditions Met</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Observation 5</td>
<td>Conditions Met</td>
<td>+5%</td>
<td>Conditions Met</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Observation 6</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>+0.05%</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Observation 7</td>
<td>Conditions Met</td>
<td>+10%</td>
<td>+0.05%</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Observation 8</td>
<td>+10%</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Observation 9</td>
<td>+10%</td>
<td>+5%</td>
<td>Conditions Met</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Observation 10</td>
<td>+10%</td>
<td>Conditions Met</td>
<td>+0.05%</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Observation 11</td>
<td>+10%</td>
<td>+10%</td>
<td>Conditions Met</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Observation 12</td>
<td>+10%</td>
<td>+5%</td>
<td>Conditions Met</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Observation 13</td>
<td>+10%</td>
<td>Conditions Met</td>
<td>+0.05%</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Observation 14</td>
<td>+10%</td>
<td>+10%</td>
<td>+0.05%</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Observation 15</td>
<td>+20%</td>
<td>Conditions Met</td>
<td>Conditions Met</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Observation 16</td>
<td>+20%</td>
<td>+5%</td>
<td>Conditions Met</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Observation 17</td>
<td>+20%</td>
<td>Conditions Met</td>
<td>+0.05%</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Observation 18</td>
<td>+20%</td>
<td>+10%</td>
<td>Conditions Met</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Observation 19</td>
<td>+20%</td>
<td>+5%</td>
<td>Conditions Met</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Observation 20</td>
<td>+20%</td>
<td>Conditions Met</td>
<td>+0.05%</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Observation 21</td>
<td>+20%</td>
<td>+10%</td>
<td>+0.05%</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>
### Liu 2010B

Added by Todd Burdullis, last edited by Todd Burdullis on Sep 17, 2010  |  [view change]

Liu 2010B August run

10BH17 Liu A program (all OGs just need one iteration)

<table>
<thead>
<tr>
<th>OG</th>
<th>Date</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>O4</td>
<td>Sep14</td>
<td></td>
</tr>
<tr>
<td>O8</td>
<td>Sep16</td>
<td></td>
</tr>
<tr>
<td>O9</td>
<td>Sep16</td>
<td></td>
</tr>
<tr>
<td>O10</td>
<td>Sep16</td>
<td></td>
</tr>
<tr>
<td>O12</td>
<td>Sep15</td>
<td></td>
</tr>
<tr>
<td>O13</td>
<td>Sep14</td>
<td></td>
</tr>
<tr>
<td>O14</td>
<td>Sep15</td>
<td></td>
</tr>
<tr>
<td>O15</td>
<td>Sep14</td>
<td></td>
</tr>
<tr>
<td>O16</td>
<td>Sep14</td>
<td></td>
</tr>
<tr>
<td>O17</td>
<td>Sep15</td>
<td></td>
</tr>
<tr>
<td>O18</td>
<td>Sep14</td>
<td></td>
</tr>
<tr>
<td>O19</td>
<td>Sep14</td>
<td></td>
</tr>
<tr>
<td>O22</td>
<td>Sep14</td>
<td></td>
</tr>
</tbody>
</table>

*Figure 27 - Scheduling Constraints outside of the QSO databases*
Figure 28 - Previous Constraints should be defined here
OBSERVING CONSTRAINTS AT A GLANCE

(please see the "Details" page for more information). NOTE: THESE ARE STRICT CONSTRAINTS AND MUST NOT BE EXCEEDED

**Seeing:**
- $u < 1.1$ arcsec
- $g, r$ and $z < 1.0$ arcsec
- $i < 0.6$ arcsec

**Moon Illumination:**
- $u$ and $g$: < 20%
- $r$: < 40%
- $i$: > 20%
- $z$: ANY

**Airmass:**
- $u$ and $i$: < 1.5
- $g, r$ and $z$: < 2.0

**Cloud Coverage:** Photometric conditions are preferred. However, for the entire program, if the moon illumination is less than 10% and the moon is farther than 40 degrees from the fields, we can allow up to 0.1 mag of REAL extinction, or 0.25 mag as measured by the Sky Probe.

**Twilight Data:** no data should be taken while or before crossing the 12 degree twilight mark.

Figure 29 - NGVS Observing Constraints
June 30, 2010 -- PROGRESS ON 09B/10A DATA ACQUISITION

Partially completed exposures:
- 10AP03 OG12 needs OB83 and OB84 from DP1, and all of DP2-DP5 (OBs 78-84) (i-band)
- 10AP03 OG9 needs OB60 for DP2 and OB63 for DP3 (i-band; this is a repeat)
- 09BP03 OG1 needs OB6 for DP2 and OB6 for DP4 (u-band; this is a repeat)
- 09AP03 OG1 needs OB1 for DP1, OB4, OB5 and OB6 for DP11 (u-band, this is a repeat)

IN ADDITION, THE FOLLOWING OGs REMAIN TO BE DONE:
- 09AP04 OG1 (single r-band exposure on background field)
- 09AP04 OG16-51 and OG62 (all i-band short)
- 10AP03 OG69-103 (i-band, short)

Completed OGs:

Summary: All u, g and z-band data, both long and short, have been obtained:
- 09AP03 OG2 and OG3 (i-band, long)
- 09BP03 OG2 (g-band, long)
- 09BP04 OG2 (g-band, long), OG3 (g-band, long), OG4 (z-band, long)
- 10AP03 OG1 and OG2 (u-band, long), OG3 to OG7 (g-band, long); OG8, OG10 and OG11 (z-band, long)
- 09BP03 OG3-OG9 (u-band, short)
- 09BP04 OG5-OG15 (g-band, short); OG52 to OG61 (z-band, short)
- 10AP03 OG20 to OG33 (u-band, short); OG34 to OG68 (g-band, short); OG104 to OG144 (z-band, short)

CLICK HERE FOR A COMPLETE SUMMARY OF OBSERVATIONS FROM THE 2MASS PROJECT.

Feb 24, 2009. IMPORTANT NOTE ON EXTINCTION: for the entire program.
<table>
<thead>
<tr>
<th>Time</th>
<th>Duration</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aug 18 2010 6:56:42:390PM</td>
<td>704</td>
<td>W1.4-3+2.E2.8</td>
</tr>
<tr>
<td>Aug 19 2010 6:56:09:766PM</td>
<td>705</td>
<td>W1.4+3+2.E2.7</td>
</tr>
<tr>
<td>Aug 19 2010 6:55:42:139PM</td>
<td>699</td>
<td>W1.4+3+1.E2.4</td>
</tr>
<tr>
<td>Aug 19 2010 6:55:07:120PM</td>
<td>700</td>
<td>W1.4+3+1.E2.3</td>
</tr>
<tr>
<td>Aug 18 2010 6:54:01:523PM</td>
<td>693</td>
<td>W1.4+2+3.E2.1</td>
</tr>
<tr>
<td>Aug 18 2010 6:54:14:120PM</td>
<td>694</td>
<td>W1.4+3+1.E2.9</td>
</tr>
<tr>
<td>Aug 19 2010 6:55:15:673PM</td>
<td>695</td>
<td>W1.4+3+1.E2.8</td>
</tr>
<tr>
<td>Aug 19 2010 6:45:29:890PM</td>
<td>723</td>
<td>W1.4+4+1.E2.7</td>
</tr>
</tbody>
</table>

Figure 31 - Scheduling Constraints for P22
8.5 Appendix 5 – Examples of proposed development that should be folded into PH1/PH2/QSO Tools development

Figure 32 - Proposed Software Development