New Observing Concepts for ESO Survey Telescopes

T. Bierwirth*[a], T. Szeifert[b], D. Dorigo[a], P. Nunes[a], M. Rejkuba[a],
K. Baugh[c,a], M. klein Gebbinck[a], A. Manning[c,a], D. Muravov[d,a], I. Vera[a]

[a]European Southern Observatory, Karl-Schwarzschild-Str. 2, D-85748 Garching, Germany;
[b]European Southern Observatory, Casilla 19001, Santiago 19, Chile;
[c]Michael Bailey Associates GmbH, Prinzregentenstr. 52, D-80538 Munich, Germany;
[d]Gesellschaft fuer Technische Kommunikation, Rotebuehlstraße 64, D-70178 Stuttgart, Germany

ABSTRACT

The start of operations of the VISTA survey telescope will not only offer a new facility to the ESO community, but also a new way of observing. Survey observation programs typically observe large areas of the sky and might span several years, corresponding to the execution of hundreds of observations blocks (OBs) in service mode. However, the execution time of an individual survey OB will often be rather short. We expect that up to twelve OBs may be executed per hour, as opposed to about one OB per hour on ESO's Very Large Telescope (VLT). OBs of different programs are competing for observation time and must be executed with adequate priority. For these reasons, the scheduling of survey OBs is required to be almost fully automated. Two new key concepts are introduced to address these challenges: ESO's phase 2 proposal preparation tool P2PP allows PIs of survey programs to express advanced mid-term observing strategies using scheduling containers of OBs (groups, timelinks, concatenations). Telescope operators are provided with effective short-term decision support based on ranking observable OBs. The ranking takes into account both empirical probability distributions of various constraints and the observing strategy described by the scheduling containers. We introduce the three scheduling container types and describe how survey OBs are ranked. We demonstrate how the new concepts are implemented in the preparation and observing tools and give an overview of the end-to-end workflow.

Keywords: survey, observing, ranking, scheduling, container, observing block, service mode, VISTA

1. INTRODUCTION

The introduction of ESO's new *Visible and Infrared Survey Telescope for Astronomy* (VISTA) and the forthcoming *VLT Survey Telescope* (VST) impose new challenges on the preparation, review and execution of survey observations that require dedicated tool support. In order to make the most efficient use of the available observing time, and in particular to be able to guarantee that the most demanding observations are executed under requested weather constraints, ESO carries out the majority of its observations in *service mode*, implementing the following workflow: a principal investigator of an accepted observation program is granted observation time under requested weather constraints in terms of *observing runs*, each of which in turn consists of one or several *observation blocks* (OBs). An OB constitutes the smallest observational unit and contains all information necessary to execute sequentially and without interruption a set of correlated exposures, involving a single target, i.e. a single telescope preset. An OB is specified by the principal investigator using ESO's Phase 2 Proposal Preparation Tool (P2PP), subsequently checked into ESO's central OB database, verified and accepted by the User Support Department (USD), replicated to the telescope site at Cerro Paranal in Chile and finally executed by an ESO night astronomer or telescope instrument operator using the observing tool (OT).

In normal VLT service mode operations, OBs are considered largely independent of each other. Each OB has its own lifecycle and can be specified, checked into or out from the OB database without any dependency on other OBs. If the principal investigator wants to impose any constraints on the execution of several related OBs – such as execution order – these constraints have to be specified in a readme file associated to the observing run and have to be read, interpreted and manually implemented by the night astronomer prior to OB execution.

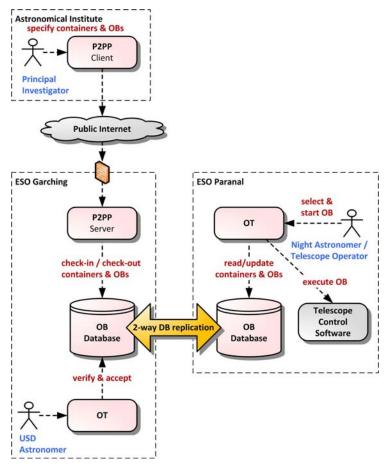*tbierwir@eso.org; phone +49 89 3200 6648; fax +49 89 320 2362; www.eso.org

Figure 1. OB specification, review and execution workflow

2. CHALLENGES OF SURVEY OBSERVATIONS

Survey observation programs typically observe large areas of the sky and might span several years, corresponding to the execution of many hundreds of OBs in service mode. However – depending on the depth of the observations and on the survey strategy – the execution time of an individual OB will often be rather short. On VISTA, we expect that up to twelve OBs may be scheduled per hour, as opposed to about one OB per hour on the VLT. The large number of OBs and their shorter duration increases the decision taking pressure on the telescope instrument operators, who have to decide from a pool of many hundreds of available OBs which one to execute next. We developed survey-specific tool support with the following major goals

- Execute survey observations highly efficiently by making optimal usage of the available telescope time while guaranteeing fair competition of survey programs

- Allow principal investigators to express complex, long-term observation strategies and to have the possibility for necessary revision of OBs after the observations have already started

- Allow principal investigators to prepare many similar OBs in a semi-automatic way

- Increase OB execution automation by providing the telescope instrument operators with an effective ranking engine that suggests the next OB to be executed

- Minimize operational overhead by integrating OB execution status reports

3. SCHEDULING CONTAINERS

The means to allow principal investigators to express more complex observation strategies is the introduction of an additional abstraction on top of individual OBs that allows expressing dependencies between them. Initially, we introduce three types of OB scheduling containers.

3.1 Time Link

A *time link* defines an ordered sequence of OBs with minimum and maximum execution time delay between them, referred to as *earliest after previous* and *latest after previous*. With the exception of the first OB in the time link, every OB has a *relative* time constraint window with respect to the execution time of its predecessor. *Open* time link OBs, i.e. OBs for which *latest after previous* is unspecified, are allowed.
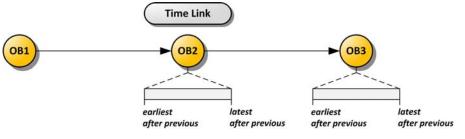
Figure 2. Time Link scheduling container

The first OB of a time link can be executed at any suitable time. After execution of the first OB, the *relative* time constraint window of the next OB is converted to an *absolute* time constraint window. If the next time link OB to be executed fails to meet its absolute time constraint window, a hypothetical execution time in the middle of its constraint window is assumed and the execution continues to the next OB.
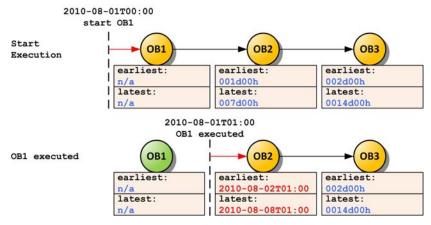
Figure 3. Execution of Time Link Container

3.2 Concatenation

A *concatenation* defines an unordered set of OBs to be executed with no break, i.e. back to back. Execution order of the OBs is not specified. A concatenation can be thought of as one super OB to be executed atomically. If any OB of the concatenation fails to execute successfully, the entire concatenation is considered as failed and has to be reexecuted. A typical example of a concatenation is the execution of a calibration OB and a science OB.
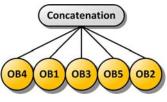
Figure 4. Concatenation Container

3.3 Group

A group allows principal investigators to express their preference to execute several OBs "close to each other". The constraint on execution preference is only desirable, not mandatory. This is achieved through a *group score* defined in units of percent defaulting to 0% at the start of observations. Every OB belonging to a group is assigned a *group contribution* in percent, indicating how much the group score is increased by sucessfully executing the OB.
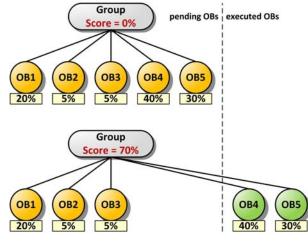

Figure 5. Group Container

If many similar OBs are observable, preference is given to OBs belonging to groups with the highest group score. In case of equal group scores of several OBs, preference is given to the OB with highest group contribution. We use an example to illustrate how group score and group contribution are evaluated: Suppose we start with two identical groups, each containing three OBs with group contributions 50%, 20% and 30% respectively.

OB Name	Group Name	Group Score	Group Contribution
OB_A	Group 1	00%	50%
OB_B	Group 1	00%	20%
OB_C	Group 1	00%	30%
OB_D	Group 2	00%	50%
OB_E	Group 2	00%	20%
OB_F	Group 2	00%	30%

First, OB_A is executed, raising group 1's score to 50%.

OB Name	Group Name	Group Score	Group Contribution
OB_B	Group 1	50%	20%
OB_C	Group 1	50%	30%
OB_D	Group 2	00%	50%
OB_E	Group 2	00%	20%
OB_F	Group 2	00%	30%

Next, OB_C is executed, raising group 1's score to 80%.

OB Name	Group Name	Group Score	Group Contribution
OB_B	Group 1	**80%**	20%
OB_D	Group 2	00%	50%
OB_E	Group 2	00%	20%
OB_F	Group 2	00%	30%

Now suppose that for some other reason, OB_B becomes non-observable.

OB Name	Group Name	Group Score	Group Contribution
~~OB_B~~	~~Group 1~~	~~80%~~	~~20%~~
OB_D	Group 2	00%	50%
OB_E	Group 2	00%	20%
OB_F	Group 2	00%	30%

OB_D is executed, raising group 2's score to 50%.

OB Name	Group Name	Group Score	Group Contribution
~~OB_B~~	~~Group 1~~	~~80%~~	~~20%~~
OB_E	Group 2	**50%**	20%
OB_F	Group 2	**50%**	30%

Now suppose that OB_B becomes observable again.

OB Name	Group Name	Group Score	Group Contribution
OB_B	Group 1	80%	20%
OB_E	Group 2	50%	20%
OB_F	Group 2	50%	30%

OB_B is executed, thereby finishing the execution of group 1.

OB Name	Group Name	Group Score	Group Contribution
OB_E	Group 2	50%	20%
OB_F	Group 2	50%	30%

OB_F is executed, raising group 2's score to 80%.

OB Name	Group Name	Group Score	Group Contribution
OB_E	Group 2	**80%**	20%

Finally, OB_E is executed, completing the execution of group 2. The example illustrates, that although we have two identical groups to begin with, once execution of group 1 is started, the algorithm assigns higher priority to OBs in that group. Only when – due to other constraints - group 1 has no more observable OBs to offer, execution switches to group 2, but returns to group 1 as soon as observable OBs are available again. This behaviour can be used by a principal

investigator to express different observing strategies. As shown in figure 6, suppose the principal investigator has divided a survey area into one hundred tiles, each of which shall be observed in three different bands.
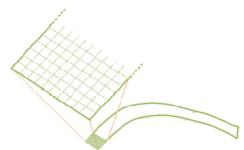

Figure 6. Dividing a survey area into tiles, each of which corresponds to a single telescope preset

Depending on the science goal, the principal investigator might favor different observing strategies as follows

- **Strategy 1 – a tile should be observed in all three bands before moving to the next tile**
  → For every tile, a group of three OBs shall be defined corresponding to the three filters, i.e. 100 groups with 3 OBs

- **Strategy 2 – the complete survey area should be observed in one band first, before switching to the next band**
  → For every band, a group of one hundred OBs shall be defined, i.e. 3 groups with 100 OBs

4. TIME CONSTRAINTS

In addition to the new concept of scheduling containers, we also allow principal investigators to formally specify one or several absolute time constraint intervals for an OB in which it has to be observed in order to constitute a successful observation. These time constraint intervals can be imposed on "loose", i.e. non-container OBs, on the first OB of a time link, on a single OB of a concatenation or on any OB of a group.

Figure 7. Defining groups, timelinks and concatenations in P2PP for Surveys

5. OB FILTERING

After the OBs have been specified by the principal investigator using P2PP, checked-in into the OB database at the ESO headquarters in Garching, verified and accepted by the User Support Department and replicated to the OB database at Cerro Paranal they are finally ready for observation execution. The telescope instrument operator uses the *Observing Tool for Surveys* (OT for Surveys) to analyze the pool of accepted OBs and to decide which OB to execute next. When the execution of the previous OB has finished, the operator initiates a new OB ranking action and OT for Surveys presents an ordered list of OBs to suggest the next OBs to be executed. However, at any moment in time during the night, a significant amount of OBs from the pool of accepted OBs will not be observable, because one or several of their requested constraints cannot be met. Consequently, prior to applying a ranking algorithm to the OBs in the pool, the non-observable OBs have to be filtered out.
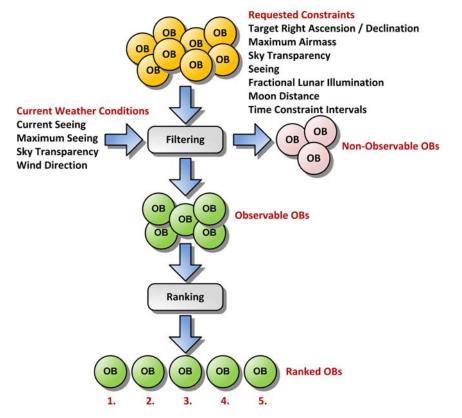

Figure 8. OB Filtering and Ranking

Thus, prior to ranking the operator has to input the following parameters and constraints, many of which are optional

Basic Parameters

Start Date	The date and time for which the OB filtering (and ranking) shall be computed
Duration	The "look ahead" time interval considered for filtering OBs (1h | 2h | 4h | all night)
Step Interval	The resolution indicating for how many steps in the duration interval the visibility is computed

In the first filtering step, all OBs are discarded whose requested weather constraints cannot be met.

Current Weather Conditions

Current Seeing	The current seeing at the observatory for $\lambda = 600nm$ at zenith, i.e. airmass $z = 1$

Max. Seeing	The maximum accepted seeing for $\lambda = 600nm$ at zenith, i.e. airmass $z = 1$
	Filter Condition: Current Seeing ≤ Requested Seeing ≤ Max. Seeing
Current Sky Transparency	The current sky transparency at the observatory: Photometric (1) | Clear (2) | Thin Clouds (3) | Thick Clouds (4)
	Filter Condition: Requested Sky Transparency ≤ Current Sky Transparency
Current Wind Direction	To make sure the telescope is pointing at least 90 degrees away from the current wind direction
	Filter Condition: 90° ≤ Target's Angular Distance from Wind Direction ≤ 270°
	(this condition is only applied if wind speed is above the telescope pointing limit)

All filtering options (seeing, sky transparency, wind direction) can be separately enabled or disabled

The operator enters the current seeing for $\lambda = 600nm$ at zenith, i.e. airmass $z = 1$. However, the seeing depends on the OB's filter wavelength λ and on the target's current airmass $z_{current}$. Prior to applying the seeing constraint, the OB's requested seeing is therefore normalized into a comparable reference seeing at $\lambda = 600nm$ and zenith. For instance, with a requested seeing constraint of $s_{requested} = 1.0$ arcseconds using the R-Filter ($\lambda = 650nm$) at airmass $z_{current} = 2$, one would require a normalized seeing of $s_{requested,\ normalized} = 0.77$, while one can still observe at $s_{requested,\ normalized} = 0.98$ in the K-band at $\lambda = 2200nm$. If the OB specifies several filters, we normalize using the bluest filter λ_{bluest}.

$$s_{requested,normalized}(z=1, \lambda=600nm) = s_{requested}\ z_{current}^{-0.4}(\lambda_{bluest}/600nm)^{0.2} \qquad (1)$$

In the second filtering step, piecewise binary functions (0 | 1) are calculated for the following visibility constraints for every step in the specified duration interval to indicate whether the constraint is fulfilled.

Visibility Constraints

Airmass	**Filter Condition**: Current Airmass ≤ Requested Airmass
Moon Angular Distance	**Filter Condition**: (Moon is down) or (Moon is up and (Current Moon Distance ≤ Requested Moon Distance))
Fractional Lunar Illumination (FLI)	**Filter Condition**: Current FLI ≤ Requested FLI
Zenith Avoidance	**Filter Condition**: Telescope points at least 5° away from zenith
Time Intervals	**Filter Condition**: Requested time intervals allow full OB execution at the given point in time

Airmass and time intervals are mandatory constraints, moon distance, FLI and zenith avoidance are optional constraints

In order to compute the overall visibility, all binary functions are multiplied and only those OBs are returned, for which the start date specified by the operator is within one of their time constraint intervals (if any). If any OB of a concatenation turns is non-observable, all OBs of that concatenation are considered non-observable. The computation of the piecewise binary functions for the entire specified duration is not strictly necessary for the OB filtering, but it allows OT for Surveys to produce graphical visibility plots that help the operator or night astronomer to visualize the visibility of an OB for the rest of the night and thus to manually schedule OB execution should this be required.

6. OB RANKING

The ranking algorithm applied to VISTA OBs is not an evolution of the VLT algorithm but has been developed from scratch by ESO's Science Operations department based both on theoretical considerations as well as on simulations and practical experience with the short term scheduling of VLT service mode observations. The algorithm will be further evolved and fine-tuned as we gain more experience during the execution of the first six public surveys on VISTA. The ranking of OBs takes into account several cumulative probability functions, absolute time constraints, the score and contribution of group OBs and the relative ranking of different approved observing runs.

6.1 Seeing Probability P_{seeing}

In order to derive a suitable value for the seeing probability requested by an OB, we use a similar approach as in the OB filtering step described in equation (1). However, in this case we do not normalize to the airmass at zenith ($z = 1$) but

rather to the minimal airmass a target can reach, which is given by the difference between the target's declination δ and the observatory's latitude $lat_{observatory}$

$$z_{Minimal} \approx \frac{1}{\cos(lat_{observatory} - \delta)} \tag{2}$$

$$s_{requested,normalized}(z_{Minimal}, \lambda = 600nm) = s_{requested} \; z_{Minimal}^{-0.4} \; (\lambda_{bluest}/600nm)^{0.2} \tag{3}$$

The cumulative probability distribution of the seeing at Cerro Paranal for $\lambda = 600nm$, taking into account VISTA's instrument resolution (~0.5 arcsec) has been measured as follows

Cumulative probability of seeing at Cerro Paranal for $\lambda = 600nm$

Seeing [arcseconds]	< 0.4	< 0.5	< 0.6	< 0.8	< 1.0	< 1.2	< 2.0
Cumulative Probability	10%	20%	30%	50%	90%	95%	100%

Using the normalized seeing at the minimal airmass the target can reach, we simply interpolate in the cumulative probability distribution of the seeing to retrieve the OB's P_{seeing}.
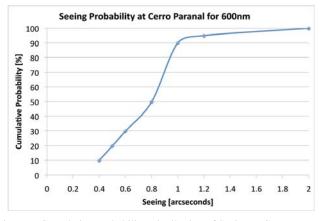

Figure 9. Cumulative Probability Distribution of Seeing at Cerro Paranal

6.2 Airmass Probability P_z

The OB specifies a maximum airmass $z_{requested}$ which is tolerable for the observation. Using this value and the target's declination δ one can calculate the hour angle h_z when the target rises above the airmass limit. The probability for the fulfillment of the airmass constraint is proportional to the length of the tolerable airmass interval, which is twice that hour angle

$$h_z = 24/360 * a\cos\left(\frac{\sin(alt) - \sin(lat_{observatory})\sin(\delta)}{\cos(lat_{observatory})\cos(\delta)}\right) \tag{4}$$

where $lat_{observatory}$ is given by the latitude of the observatory and alt can be calculated from the request airmass

$$alt = a\sin(\frac{1}{z_{requested}}) \tag{5}$$

We normalize the airmass probability P_z by the hour angle reached for an airmass limit of 1.7 and $\delta = lat_{observatory}$.

$$P_z = \frac{h_z}{h_z(z_{requested} = 1.7, \delta = lat_{observatory})} \tag{6}$$

6.3 Sky Transparency Probability P_{sky}

The cumulative probability distribution of the sky transparency at Cerro Paranal has been measured as follows

Cumulative probability of sky transparency at Cerro Paranal

Sky Transparency	Photometric	Clear	Thin	Thick
Cumulative Probability	50%	80%	90%	95%

The probability P_{sky} is simply interpolated using the OB's requested sky transparency.
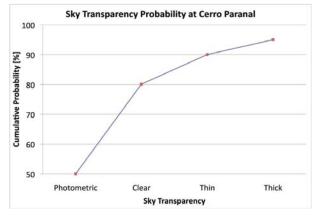

Figure 10. Cumulative Probability Distribution of Sky Transparency at Cerro Paranal

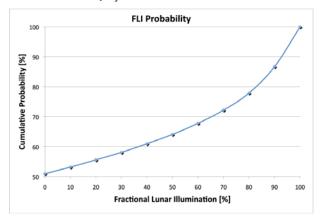## 6.4 Fractional Lunar Illumination Probability P_{fli}

Figure 11. Cumulative Probability Distribution of Fractional Lunar Illumination

The cumulative probability of the fractional lunar illumination is not dependent on any target properties or constraints. We use a simple lookup table to interpolate the propability using the OB's specified FLI constraint. We do not currently take into account the angular distance between the target and the moon for OB ranking.

6.5 Setting Target Probability $P_{setting}$

The probability to successfully execute an OB is reduced for targets which are already close to setting at the start of night. We promote such OBs in the overall ranking to give them higher execution priority. The target's hour angle h_{dusk} at the beginning of night is a function of the target's right ascension α and the local sidereal time at dusk ST_{dusk}.

$$h_{dusk} = \alpha - ST_{dusk} \quad \text{(add 24h if } h_{dusk} < \text{-12, subtract 24h if } h_{dusk} > 12) \tag{7}$$

The target's hour angle h_z when it rises above the requested airmass limit $z_{requested}$ is given by equation (4). The target's remaining *tracking time* at the beginning of the night is then given by

$$t = h_z + h_{dusk} \quad (if \ h_z > h_{dusk})$$
$$t = 2 * h_z \quad\quad (if \ h_z \leq h_{dusk})$$

(8)

And the normalized probability function P_{set} is calculated as follows

$$P_{setting} = \frac{h_z + h_{dusk}}{2 * h_z} \quad (if \ h_z > h_{dusk})$$
$$P_{setting} = 1.0 \quad\quad (if \ h_z \leq h_{dusk})$$

(9)

This concludes the discussion on deriving suitable probabilities for observability constraints. The next sections will discuss additional constraints and how the overall OB ranking is computed.

6.6 Time Rank R_{time}

The time rank has to take into account both absolute time constraints of time link OBs as well as the additional time constraint interval applied to all sorts of OBs. For every OB, the *total remaining constraint time* $\Delta t_{remainingConstraint}$ is the sum of all remaining time constraint intervals, thereby giving a measure for the OB's time rank R_{time}. We give execution preference to OBs whose remaining time constraint time expires within about one lunar cycle, therefore we normalize the time rank by 30 days.

$$R_{time} = \min(1, \Delta t_{remainingConstraint} / 30d)$$

(10)

6.7 Observability Class

We can now combine the five observability probabilities and the time rank to an overall observability class

$$obs\_class = 10 * NINT \left[10 * P_{seeing} * P_z * P_{sky} * P_{fli} * P_{setting} * R_{time} \right]$$

(11)

We express the resulting *obs_class* as a 3-digit string of the form "000", "010", "020", … "100". The rounding to the nearest integer (NINT) is strictly required, since otherwise the observability class would be slightly different for every OB, and other ranking criteria such as run rank class, user priority group score and contribution would never take effect.

6.8 Run Rank Class

The observing runs that are granted observing time are prioritzed by the observing programmes selection committee. Typically three rank classes "A", "B" and "C" are used, with "A" indicating the highest priority and "C" the lowest.

$$run\_rank\_class = A \,|B \,| \,C$$

(12)

6.9 User Priority

The principal investigator assigns a user priority to each loose OB or to scheduling containers. All OBs belonging to a scheduling container have the same user priority, which they inherit from their container. The user priority has a range from [1…10] and is expressed as a 2-digit string of the form "01", "02", "03", … "10", where the default of "01" corresponds to the highest priority.

$$user\_priority = 01 \,| \,02 \,| \,03 \,| \,04 \,| \,05 \,| \,06 \,| \,07 \,| \,08 \,| \,09 \,| \,10$$

(13)

6.10 Group Score and Contribution

As discussed in section 3.3., higher group scores and contributions of OBs should increase their execution priority. In order to express these properties as strings where lower values correspond to higher execution priority, we invert the two properties and express them in units of percent as a string of the form "xxx.xx"

$$inv\_group\_score = 100 - group\_score$$

(14)

$$inv\_group\_contrib = 100 - group\_contrib$$

6.11 Overall OB Rank

We can now combine all discussed observability probabilities, the time rank, the run rank class, the user priority and the group score and group contribution into a final, overall OB rank. The OB rank is computed for each OB in terms of a rank string concatenating all criteria. The strings are then sorted alphabetically. Strings with a lower lexicographical value correspond to higher execution priority.

$$OB\ rank = <obs\_class>\_<run\_rank\_class>\_<user\_priority>\_<inv\_group\_score><inv\_group\_contribution> \qquad (15)$$

The string "000_A_01_040.00_095.00" would correspond to a group OB, which should be executed with high priority due to a very low observability probability, which has a run rank class "A", a default user priority of 1, a group score of 60% and a group contribution of 5%.

7. OTHER ASPECTS

Scheduling containers create dependencies between several OBs resulting in significant operational side effects as well as implementation complexity. Scheduling containers that were already partially executed might need to be modified by the principal investigator, requiring a partial checkout of the container from the OB database, followed by local editing of OBs and containers and an update into the database. Care must be taken that after re-checkin, the scheduling containers are still in a consistent state and execution can continue. During execution, it might be required that some OBs of a time link or group have to be re-executed. Therefore, comprehensive container-level rollback scenarios have to be implemented. The consistency of the two-way synchronization between the databases at Cerro Paranal and ESO headquarters is crucially important. A very important basis for the implementation of the survey tools is a carefully designed OB state machine, which describes allowed OB state transitions both in Garching and Paranal and their side effects on other OBs. OT for Surveys changes the status of time-constrained OBs that terminally failed to meet any of their absolute time constraint intervals to "Failed", such that these OBs will be ruled out from future ranking. Should the failed OB belong to a time link container, the time constraint of the next OB in the time link has to be updated accordingly, which might result in a cascade of failures and updates, if several OBs in the time link have failed to meet their time constraints.
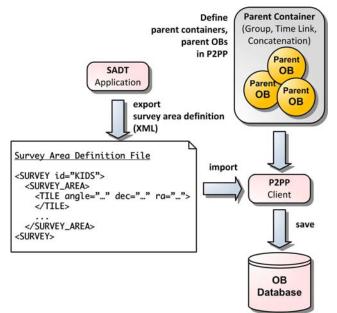

Figure 12. Definition of a Survey Area using SADT and import into P2PP

In addition to the core tools P2PP and OT for Surveys, the VISTA consortium has developed a Survey Area Definition Tool (SADT) that allows principal investigators to define the survey area as a collection of individual tiles with user-

defined overlaps between adjacent tiles. The tool not only computes the target coordinates, but also provides the list of suitable guide stars and stars necessary for active optics corrections. The result is exported in terms of a survey area definition file in XML format, which can be imported by P2PP in several ways, such that tiles can be flexibly associated to OBs of scheduling containers and a large number of similar scheduling containers and associated OBs can be easily generated. Finally, in order to support data handling and reduction, we propagate the unique container ID as well as the container type into the produced raw frames as additional FITS keywords to allow reassociation of the original OBs and containers.

8. RESULTS

The first regular VISTA observation period has started in April 2010 and runs until the end of September 2010. Although initially there are only six scheduled public surveys, a total of 4800 OBs were submitted, approved and scheduled for observations in this first period. To put these numbers into context, note that the total number of OBs submitted for a VLT unit telescope over a 6-month observing period is between 1300-2800. While these VLT numbers vary from period to period, depending on the type of observations being approved and on the usage of optical versus infrared instruments, they are almost a factor of two lower than for VISTA. So far all VISTA public survey programs used the scheduling containers extensively. Observations are mostly carried out following the suggestions of the ranking algorithm described above. If manual intervention is required, the telescope instrument operators can simply modify filtering and ranking constraints or execute other OBs following their own decision. As far as scheduling containers are concerned we have found only one case which is not covered by the concept: If time critical observations require to concatenate a calibrator or a sky field observation it would be required to have time links of scheduling containers, which is not yet implemented.

The ranking of executable observations is very powerful, yet sometimes also yields surprising results. Setting targets or time critical observations may appear on top of the list which even experienced astronomers may have easily missed in a large pool of observations. Another strong aspect is that the fast Java-based implementation provides – within seconds – a new plan for the night if the weather conditions change, or if the pointing is suddenly limited to only one direction on the sky due to strong winds. Ranking of the observations is a complex decision making process which is often based on contradicting constraints. Observation A may be the most difficult in terms of the observation conditions, but observation B may have received the highest evaluation of the review committees and observation C may have the highest priority observation for a particular user. This decision is typically based on experience and equally often on prejudices. The ranking as explained above provides a very good guess if – during a photometric night with good seeing and dark sky – the observer should give preference to an observation which requires only one of these constraints, or to an observation which requires any other possible combination in the parameter space. During the first three months we have progressed more quickly with the more demanding observations having tight constraints rather then with the observations with very loose constraints. Observations with loose constraints will be smoothly pushed to the top of the ranking at the end of the visibility period of the targets. This will guarantee not only that a minimum of observations are missed during a given period, but also that a possible shortage of observations at the end of the period is avoided.

9. OUTLOOK

The introduction of the new concept of scheduling containers in combination with a powerful OB ranking algorithm is not only attractive to survey telescopes. Our next step will be to implement the scheduling container concept also on the VLT unit telescopes and to adapt the ranking algorithms to address specific needs of particular instruments. In the future, if nested scheduling containers such as time links of concatenations are supported, even more advanced observing strategies could be expressed.

REFERENCES

[1] VISTA – Visible and Infrared Survey Telescope for Astronomy (ESO Website)
http://www.eso.info/sci/facilities/paranal/instruments/vista/
[2] SADT – The Survey Area Definition Tool (ESO Website)
http://www.eso.info/sci/observing/phase2/VIRCAM/SADT.html
[3] Rejkuba, M., "Phase 2 Proposal Preparation Tool for Surveys – User Manual", ESO,
http://www.eso.info/sci/observing/phase2/P2PP/P2PP3Documentation.html