Stottler Henke
Smarter Software Solutions

Aurora Key Features

This document discusses some of the things that differentiate Aurora from other scheduling software.

Modeling Accuracy

Aurora supplies all of the “standard” modeling features (precedence constraints, resource requirements, etc.), but it also supplies additional functionality to allow the user to quickly and easily model things that may be difficult or impossible to capture with other scheduling software. The list below covers some but not all specialized modeling support:

- Concurrent Constraints – These allow the user to specify that two jobs need to happen at the same time in the schedule.
- Exclusivities – These allow the user to specify that a job cannot happen at the same time as another job or class of jobs.
- Preferred Resources – This allows the user to specify a preference order when defining a set of resources that are mostly interchangeable.
- Alternative Resource Combinations – This allows the user to specify different combinations of requirements that could be used to complete a task, including variants with different durations.
- Variable Duration Jobs – This allows the user to specify that a job could use more people and get done more quickly, or fewer people and get done more slowly.
- Special Manufacturing/Shift Control Properties – This is a set of properties that allows the user to control how jobs interact with shift breaks (can it go between shifts, can it go from one day to another, does it have to complete a certain length of time before a shift ends, etc.)
- Capacity Change Constraints – This allows the user to specify a relationship between a task and a resource. Some tasks may make a resource available (e.g. adding a space zone that can subsequently be used for work), others may make a resource unavailable (e.g. installing panels that block access to a space zone).
- Jig Support – This is specialized support that ensures that a jig is assigned to a series of work and will be retained throughout the work statement.

Domain-Tailored Optimization

Aurora’s algorithms have been fine-tuned for sub-assembly and final-assembly scheduling. There are additional settings that allow the user to adjust some of the logic and use Aurora outside of those key target domains, but the special tailored optimization make Aurora perform very well on the dense networks and high resource contention situations common in sub-assembly and final-assembly scheduling. Some of the special support includes:

- Special downstream work analysis that will tend to get work started if more/more complex downstream work depends on it.
- Giving the option to level labor utilization within the bounds determined by a preliminary scheduling round. This evens out utilization peaks without impacting the overall schedule.

- Schedule optimization that takes the preliminary schedule, analyzes its critical chain or tent-pole, and then reschedules iteratively based on that to try to shorten the schedule.

Execution and Iteration Support

An operational scheduling system needs to work well for planning ahead (scheduling in the future), execution (scheduling something that is in progress), and iteration (taking execution information from earlier lines and applying that to later lines). Aurora provides special support to help respond to critical model updates while preventing churn, and tools to help maintain data across line numbers.

- Stable Schedule Mode – Aurora provides a mode that will prefer to retain the previous schedule, with the minimum updates necessary to incorporate emergent work or updated resource availability. This permits updates to take new data into account while avoiding churn.
- Critical Chain Analysis – The user can analyze the critical chain (project tent pole) of their whole schedule or a subset of their schedule to better understand what is preventing their schedule from being shorter. This is both a valuable execution and a valuable analytic tool.
- Actuals Analysis – The user can analyze the actuals (what actually happened when) against the model to mark constraints that were not satisfied (e.g. a successor started before a predecessor was completed). This makes it easier to update models for greater accuracy in the future.
- Flow-Based Offsets – Some groups try to maintain a high degree of consistency in terms of what jobs should happen when from one line to the next. Flow-based offsets make this easy by recomputing how far into a schedule a job should fall based on the start of a given line. This makes it much easier to achieve cross-line consistency, if desired.

Analytics

Aurora provides a variety of additional analytic features intended to help IEs better understand their model and schedule.

- Upstream/Downstream Task Analysis – These analyses start with a given job or jobs, and walk up/down the network to find the jobs it is dependent on, or the jobs that are dependent on it. The upstream analysis can help in understanding a key task and what it is dependent on; the downstream analysis can help in understanding a key task and what is impacted by it.
- Point-to-Point Analysis – This finds the path through the network from the first task to the second task (if there is such a path). It can be valuable for analyzing a subset of the network that is connected (e.g. all the work linking Milestone 1 and Milestone 2).
- Monte Carlo Simulation – This takes advantage of duration distribution information to simulate multiple executions of the schedule to show how things are likely to play out. This gives the IEs insight into how brittle their schedule is, how likely it will be to run late, etc. – and from there, try to make their schedule more robust and stable.
- Sensitivity Analysis – This takes advantage of duration distribution information to explore what would happen to the schedule if certain key jobs are faster than expected or slower than expected.

- Critical Chain Analysis – The critical chain analysis finds the limiting chain through the schedule or a subset of the schedule that will prevent the project from being completed more quickly. Aurora offers both conventional critical chain analysis (often used for CCPM but also just to better understand a project) and analysis that can be applied to various subsets of data in a schedule (e.g. positions) so that the group managing that subset of the schedule can better understand their limiting factors.
- Schedule Explanations – In the course of scheduling, Aurora captures a variety of information about why things scheduled the way they did. This can be very valuable in understanding why a project cannot be completed more quickly. This information can then be used to determine whether reformulating some aspects of the model may permit a faster schedule.
- Long-Range Planning – Aurora provides multi-line utilities intended to allow the user to run a model across many line numbers to see how the schedule performs through time.
- API – Aurora provides a programmatic API that can be used to load an Aurora model or programmatically load data, and then perform a range of operations and analysis. Aurora provides built-in access to analytics available via the UI, but because it is a programmatic API, the implementing group can also define their own manipulations and analytics.

Visualization

Aurora provides a variety of displays that are intended to help the IEs quickly and easily interact with and analyze their model and schedule.

Many of the visualizations provided by Aurora are standard, in that any decent scheduling system should have a PERT Chart, Gantt Chart, resource utilization plots, etc. However, Aurora's charts can readily be configured to filter the data, color things based on some criteria, label with a variety of information, etc. The PERT and results displays are plotter compatible and are often printed out for discussion.

Data Management

Stottler Henke understands that getting data into and out of Aurora is one of the key challenges for any scheduling team. Aurora provides customized integrations with key systems such as CMAD, as well as more flexible integrations and data management tools that can be used more generally.

- Configurable Database Import – Aurora will allow the user to import model and schedule information from an Oracle or Microsoft SQL Server database. The user can map columns from the source database to data columns in Aurora. Aurora can then parse a variety of information for robust and flexible data import.
- Configurable Database Export – Aurora will allow the user to export model and schedule information to an Oracle or Microsoft SQL Server database. The user can map columns from the destination database to data columns in Aurora.
- Tabular Editor – Aurora provides an Excel-style view into the model and schedule data, allowing the user to easily filter, analyze, extract, and enter data. Aurora also provides a "referenced paste" feature that effectively performs a VLookup cross-referencing incoming data with data in Aurora on key matching criteria to robustly transfer data without perfect row alignment.

- API – Aurora provides a programmatic API that can be used to load an Aurora model, programmatically load or update data, save an Aurora model, etc. This can be used to define a data interaction layer between Aurora and any client-controlled system without Stottler Henke's intervention.

Responsiveness to Change

Perhaps the most important thing Aurora has to offer is not a program feature: it is mindset and a group of personnel who have worked on customizing Aurora for years and – in some cases – decades.

For Stottler Henke, providing the users with what they need is the primary goal. As needs change and new considerations come into play, Stottler Henke is committed to nimbly and effectively updating Aurora's feature set to cover those emergent needs, while still supporting legacy users.