

Intelligent Scheduling Benefits for Condition Based Maintenance (CBM)

Thesis	3
Condition Based Maintenance: Potential	3
Air Force Maintenance, Repair, and Overhaul (MRO)	4
CBM and Intelligent Scheduling	4
Simple Scheduling Problem	5
Intelligent Scheduling and Stottler Henke's Aurora	6
Aurora Capabilities Apropos to CBM	7
Aurora Background and Details	8
Capabilities	10
Non-Concurrent or Exclusivity	13
Conflict Reports	15
Aurora Real World Scheduling Comparisons	16
Conclusions	18

Thesis

Intelligent scheduling is the enabler that condition based maintenance (CBM) needs to realize CBM's potential to improve fleet readiness, while reducing overall maintenance requirements and costs.

Condition Based Maintenance: Potential

Condition based maintenance (CBM) provides the opportunity to improve fleet readiness, while reducing overall maintenance requirements and costs if and only if the added knowledge provided by CBM can be managed and leveraged in an efficient manner. That is, even though the overall *required* workload for a fleet should be decreased thanks to CBM the predictability of the workload will decrease as it is not known a priori what maintenance and repairs will be required until much closer to when the actual maintenance needs to be performed compared to previously. So depending on the actual required maintenance the maintenance facility needs to adapt to the required maintenance that's actually coming in and then decide how best to leverage any extra unallocated time that otherwise would have been spent if the previous preventative/scheduled maintenance methodology was employed. If CBM is scheduled

properly the maintenance facility should be able to do more opportunistic but not required maintenance such as performing an upgrade sooner rather than later because now there is extra capacity. Therefore, the ability to intelligently schedule rapidly based on updating information based on the changing *conditions* is critical to the success of a CBM implementation.

Air Force Maintenance, Repair, and Overhaul (MRO)

The logistics activities associated with aircraft Maintenance, Repair, and Overhaul (MRO) aim to ensure the Air Force has the equipment ready to meet its demands. The USAF MRO environment deals with specialized aircraft and components, and major airframes as much as 40+ years old; it involves a range of complex, varied, and shifting technology and processes; it is distributed across far-flung commercial and military sites with very different kinds of ever-changing capabilities; and it must adapt to unpredictable shifts in resources and workloads as budget priorities change and forces surge.

CBM is a strategy that actively monitors the actual condition of the asset to perform maintenance only when it is needed and at the most opportune times. CBM dictates that maintenance should only be performed when certain indicators show signs of decreasing performance or upcoming failure. CBM was introduced to try to maintain the correct equipment at the right time, in order to reduce operating costs and increase the safety of assets requiring maintenance. Preventive/scheduled maintenance replaces parts before the end of their useful life. However, introducing CBM causes a major change in how maintenance is performed, and potentially to the whole maintenance organization, thus a paradigm shift must be accomplished to obtain the benefits of CBM, a major aspect of this shift is related to how scheduling is performed.

CBM and Intelligent Scheduling

CBM has the potential to improve maintenance agility and responsiveness, increase operational availability, and reduce life-cycle total ownership costs. However, the appropriate scheduling, and execution of maintenance operations based on the information provided by the CBM knowledge is critical to its successful implementation.

The scheduling of such operations presents a management challenge not only for the high number of components that need to be repaired but also for the diverse areas of expertise required for doing such repairs. For example, the 76th Commodities Maintenance Group (CMXG) is responsible for inspecting, repairing and testing 3,500 types of end items and 400 different secondary structures for jet engines, aircraft and cruise missiles. 76 CMXG delivers engine components for fuel systems, oil systems, electronics systems and hydraulic systems. Given the complexity of the maintenance, repair, and overhaul (MRO) operations, CBM initiatives' success relies on the use of enabling technologies that support the scheduling and execution tracking of such

operations.

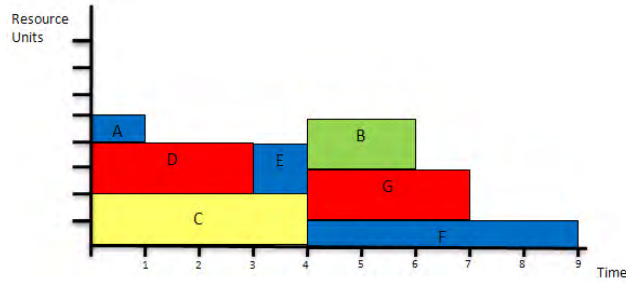
Unfortunately, most scheduling systems are neither able to produce optimal schedules for the high number of tasks and resources that need to be scheduled, nor are they designed to represent the special constraints needed to properly model PDM activities (spatial constraints, complex occupation and skill combinations, specialized equipment including armaments, non-concurrent constraints, and security concerns), nor can they schedule the large models wall-clock fast enough to support the number of what-if analyses desired for hypothetical situations based on user-defined metrics. The scheduling problem is a well-known computational hard problem and most project management tools, such as Primavera P6 and Microsoft Project, only use “resource leveling”, a simple technique that virtually always results in inefficient schedules.

All depot maintenance can benefit from intelligent scheduling, especially if the scheduling activity itself can be completed in short wall-clock times, however, the even more dynamic nature of CDM *needs* the intelligent scheduling to be performed quickly so that many what-ifs can be performed to realize the maximum benefits. Aurora can perform the intelligent scheduling at least an order of magnitude faster than the inferior resource leveling in Microsoft Project.

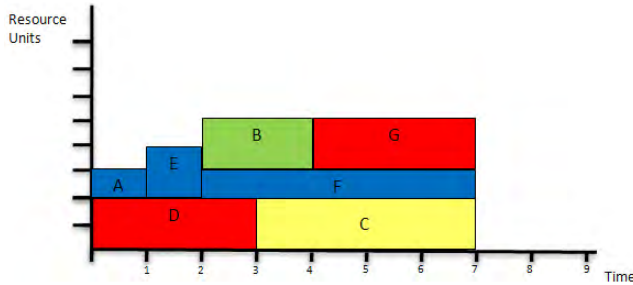
Returning to the actual scheduling results, the example in the next section and other results presented later, illustrate the reality that the duration of the schedule varies significantly depending on the scheduling algorithms.

Simple Scheduling Problem

Consider the following very simple project, shown in Figure 1, where the arrows represent finish-to-start links. The task name is shown inside the squares. The four paths can be worked in parallel as shown in the upper left box in the figure; the height of each indicates its resource requirement, which is also indicated by the number located at the top left corner of the activity. The number at the lower right corner indicates the time duration. Activity D, for example, uses two resource units and its duration is 3 units of time. There is the implicit desire to accomplish all these tasks in the shortest possible time. Microsoft Project produces the answer shown below in the upper right box of Figure 1, which requires 9 units of time. Primavera P6 produces the schedule below in the lower-left box, requiring 8 units of time. There is also an optimum result, shown in the lower-right box, requiring 7 units of time. Note that this problem only dealt with one type of resource and included only 7 tasks, and already this problem is challenging. High-quality scheduling decisions can get more tasks done in less time with the same resources. PDM planners need the appropriate scheduling tools to generate schedules with shorter turnaround times than otherwise required.



Tasks Flows	Schedule 1
-------------	------------



Schedule 2	Schedule 3
------------	------------

Figure 1. Different schedules for the same tasks flows. Schedule 3 has the optimal shortest time.

Intelligent Scheduling and Stottler Henke’s Aurora

Stottler Henke has studied human expert schedulers making scheduling decisions in critical applications for 25+ years and has implemented these decision-making processes as separate, composable methods in a software architecture known as *Aurora*. *Aurora* has been applied to a wide variety of domains, showing its generality, and in every domain where a comparison between *Aurora* and another scheduling tool was performed; *Aurora* has always been shown to generate more optimal schedules. Most of these comparisons were made by the clients themselves in order to pick the

tool that performed best. In the case of Boeing, after they performed a worldwide search for applicable scheduling tools, Aurora beat all competitors, including their own internally developed Timepiece software, which had been specifically optimized and tuned to aircraft manufacturing scheduling over a 20-year period. The use of Aurora for scheduling has typically meant that 10% to 40% more tasks can be accomplished with the same resources in the same amount of time (or the same tasks accomplished in 10% to 40% less time) compared to other scheduling methods.

Aurora Capabilities Apropos to CBM

Aurora provides a unique set of capabilities that CBM can leverage to increase efficiency across a wide a range of operations.

- Large multi-project support
 - Support for 100,000+ tasks per project
- Multiple-pass intelligent resource-constrained scheduling
 - Generates shorter project duration & shorter remaining project durations during execution
- Mixed-mode scheduling
 - Providing both as-soon-as-possible (ASAP) and as-late-as-possible (ALAP) scheduling, available on a task-by-task basis.
- Schedule Rationale: Aurora provides a rationale for each task on why it was schedule where it was scheduled
- Supports More Types of Constraints than other software, e.g.,
 - Resource constraints
 - Resource Sets – job can be performed by 2 different specialists or (1 generalist and 1 specialist) or 2 generalists.
 - Spatial constraints – e.g.,
 - task requires a certain location or type of space;
 - two elements should (or should not) be next to each other
 - Concurrent, Non-concurrent / Hazardous constraints
 - Ergonomic constraints – individual limitations on work conditions
 - Skills / Certifications in addition to Occupations
 - E.g., Mechanic (occupation) with 4 additional skills or certifications
 - Shift based constraints
 - Task needs to be completed during single shift
 - Do not start task unless x% of time left in shift
- Integrates with Enterprise Software, e.g., Oracle, SAP, and proprietary systems.

All this functionality is available on any computer users which to access it. That is there are stand-alone application versions available for Windows, Linux and the Mac. Furthermore, Aurora is available via the Web. The web server can be hosted off site, or internally.

Aurora Background and Details

Stottler Henke started working with NASA to improve the efficiency of its projects and other scheduling challenges since the 1990s. For example, Stottler Henke provided the scheduling software for the MRO of NASA's Space Shuttle. Experienced mission planners were studied to identify relevant scheduling techniques, heuristics, and data. Their knowledge was captured using a combination of rules and object-oriented representations. Techniques were developed to automate the MRO scheduling process. A full-scale Automated Manifest Planner tool (AMP) was in daily use from the mid 1990s through the end of the Space Shuttle era to schedule the MROs and perform advanced "what-if" studies.

During the 1990s Stottler Henke enjoyed further success with various other scheduling-related projects, many for NASA. After building independent scheduling solutions, it was decided that it would be wise to re-architect our scheduling software so that it would be easy to modify in the future. That is how *Aurora* came to be. The Aurora architecture was created in such a way that every *decision point* that could be changed in a scheduling system, is very easy to modify. Figure 2 shows a high-level representation of Aurora's Architecture.

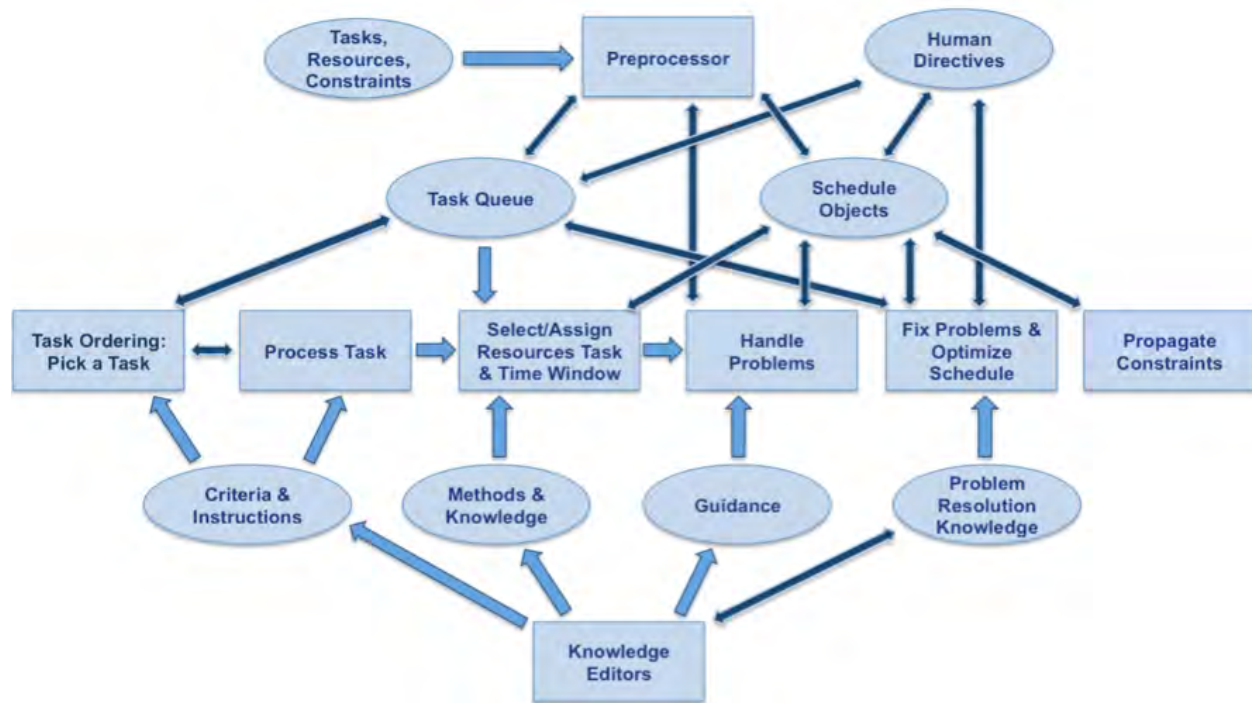


Figure 2. High-Level Aurora Architecture

To achieve this flexibility, we designed Aurora to have a number of components that could be plugged in and modified to gain varied results. The scheduling system permits

arbitrary flexibility by allowing a developer to specify what code libraries to use for different parts of scheduling. Each of the pluggable components must extend the corresponding general base class that defines the entry-point methods. This allows the objects that are integral to Aurora to interact with them successfully. The libraries may make use of any of the Aurora objects (such as activities and resources) that pass through the interface. These objects provide support for additional attribute caching, permitting domains to make use of custom properties in the scheduling heuristics. The primary pluggable components include a preprocessor; a scheduling queue prioritizer; the actual scheduler, which usually applies several scheduling methods; a conflict solution manager; and a postprocessor. See Figure 3 for a more detailed breakdown of configurable operations.

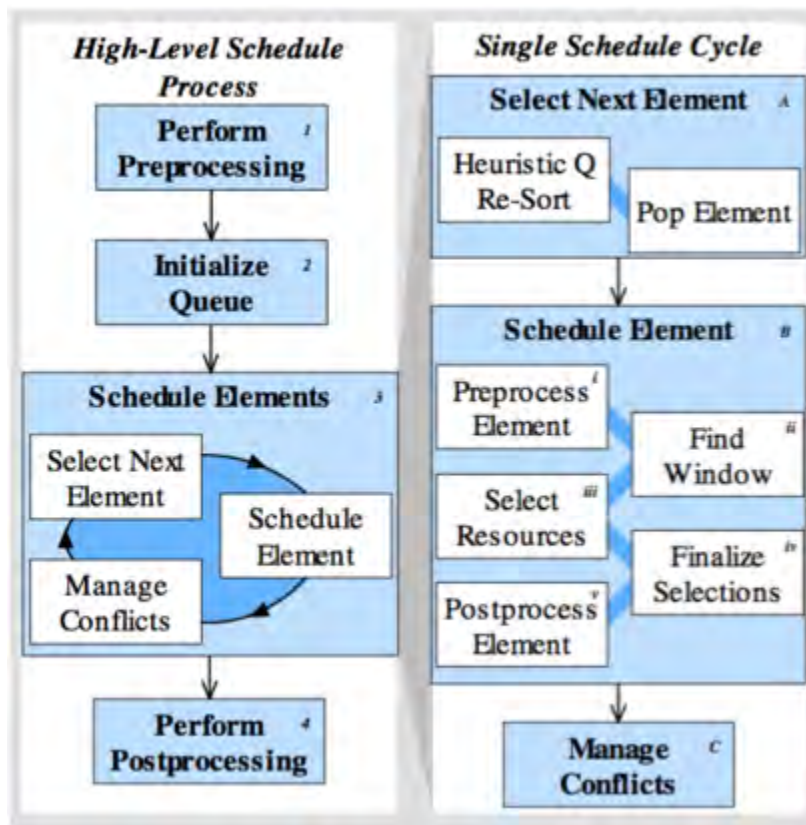


Figure 3. Aurora's reconfigurable scheduling system process breakdown.

From this new architecture, we have been able to build quite varied complex and successful scheduling systems; accomplishments range from scheduling the downlinks of US Air Force satellites & scheduling related to space debris tracking, to scheduling medical residents during their education at Harvard's Medical School, to scheduling the final assembly of the Boeing 787 jetliner and various other aircraft for Boeing, as well as similar operations for Bombardier and Learjet, to combining intelligent scheduling with

Critical Chain Project Management (CCPM), to scheduling the manufacturing facilities of pharmaceutical production for Pfizer.

Capabilities

One of the unique and powerful capabilities in Aurora is the explanation facility. For any task Aurora can explain why the task is scheduled where it is, this is a powerful capability that provides transparency into the why the schedule is scheduled the way it is, and builds trust by the users. Figure 4 shows an example of an explanation.

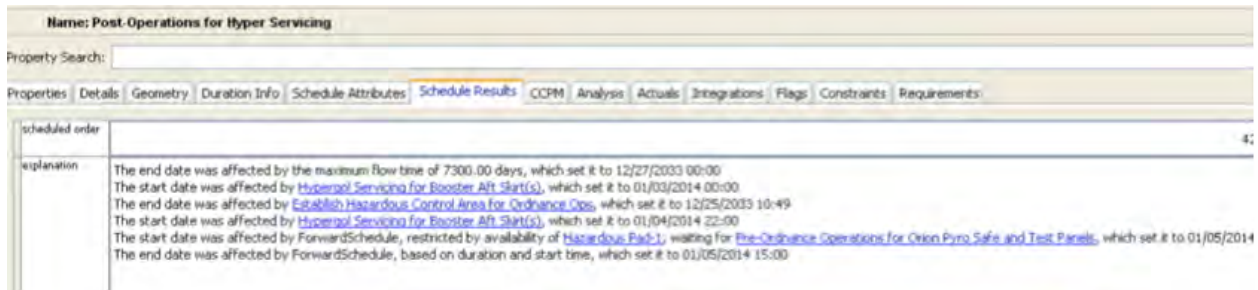


Figure 4. Automatically generated explanation

Aurora also has a histogram plot that reveals what tasks are actually consuming the resources at any slice in time by the user clicking at the time of interest. Figure 5 shows this functionality where the user clicked at the point in time represented by the vertical dashed line.

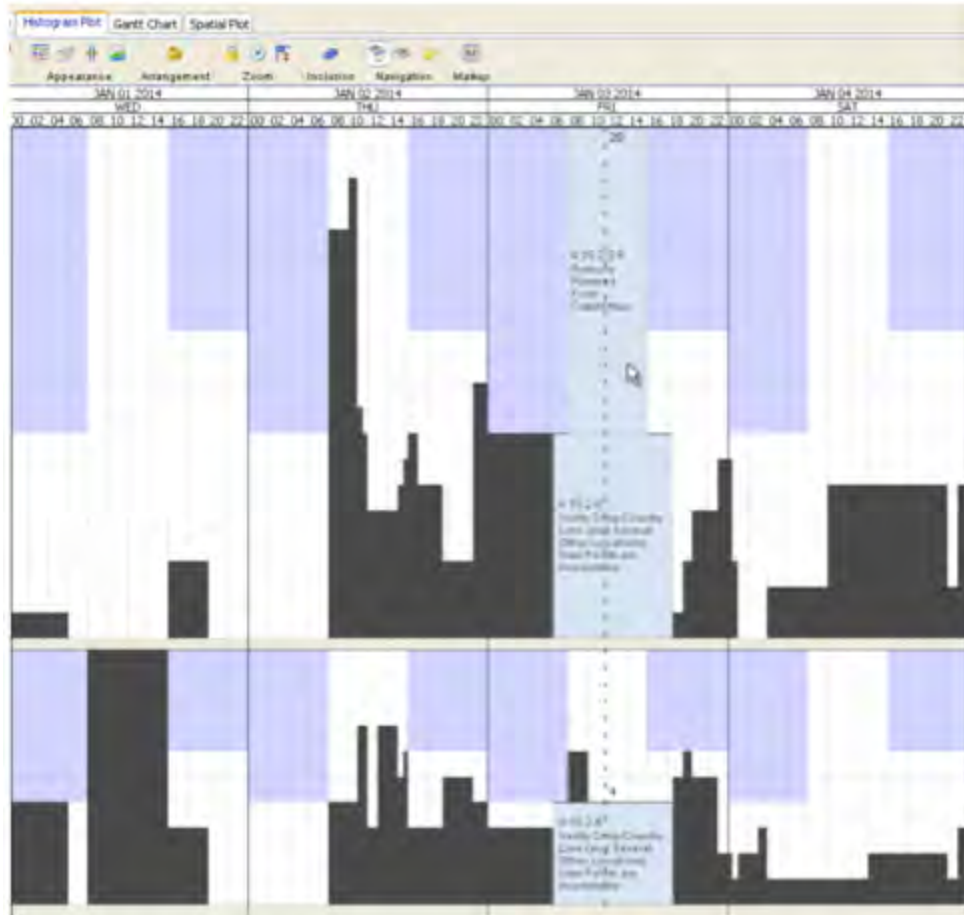


Figure 5. Manpower histogram, showing activities constituting manpower need for one time instance

Aurora also provides a split view so this histogram time slice can be combined with other views for more insight, for example it can be combined with a Gantt view to show more of the other tasks occurring around the same time span. Figure 6 shows a split view showing a Gantt chart.

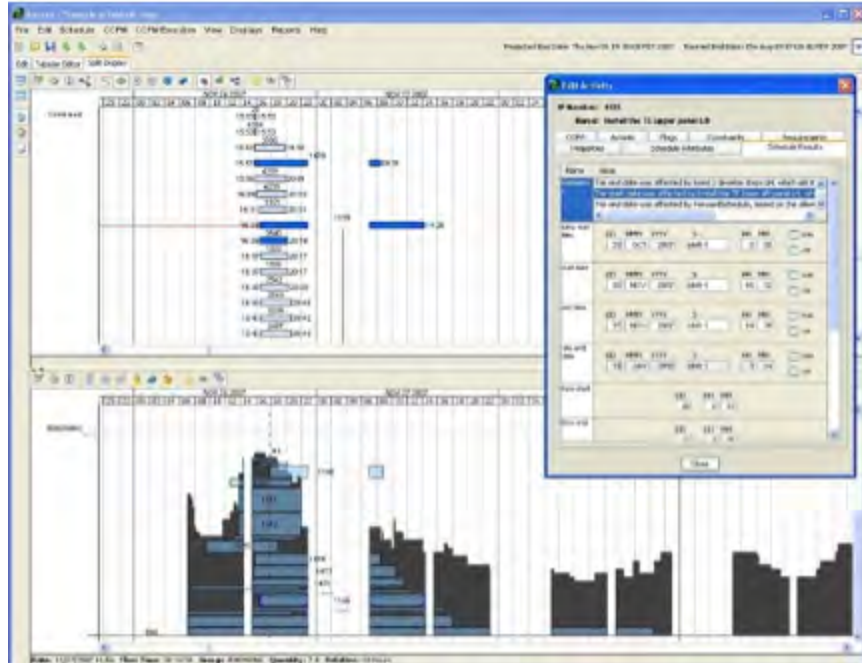


Figure 6. Split view showing Gantt chart same time slice as histogram, showing activities constituting resource need for one time instance

Finally another view is the single-element view. This is necessary since the size and complexity of the model make it usually impossible to see all the relationships between tasks in a global view. The single-element view shows a task in its own window, showing only the element, and all the tasks that it is related to, including predecessors, successors, resource links, etc. Figure 7 shows a network diagram and a single-element view shown in a separate window.

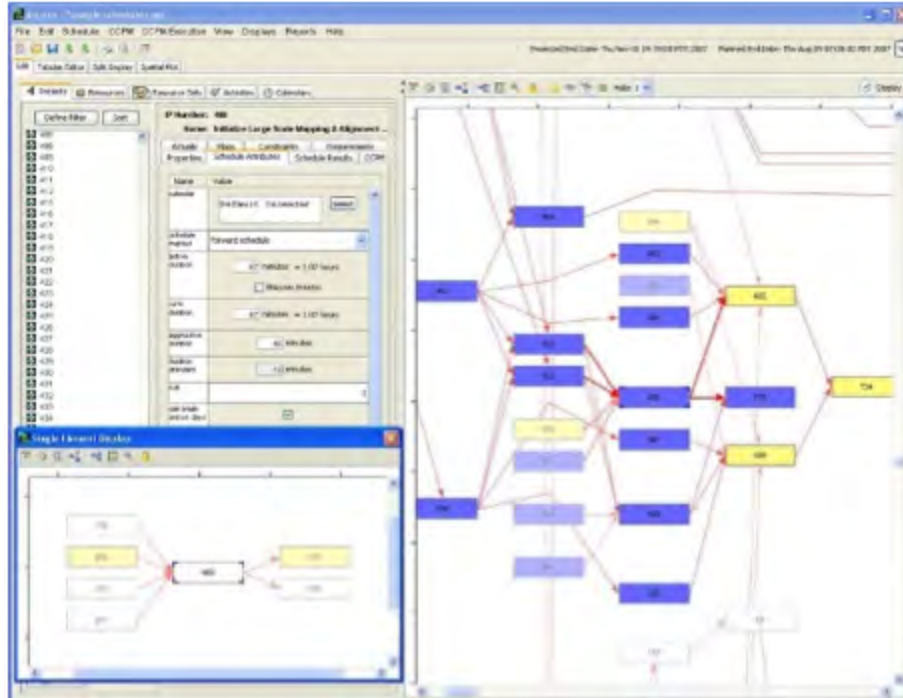


Figure 7. Network diagram showing single-element view option

Non-Concurrent or Exclusivity

Aurora provides a flexible non-concurrent or exclusivity capability, exclusivity indicates that a task cannot have any overlap with a set of other tasks. The set of other tasks is defined by one of several criteria. Before scheduling, Aurora compiles a list of all the tasks that are exclusive with each other. For example one option is to create a non-concurrent constraint between two tasks to prevent their temporal allocations from overlapping. "Exclusivity" provides a means of setting up this relationship from one task to a larger set of tasks. There are 5 types of exclusivity relationships. All of them, with the exception of "universal", have a target value. This target values refer to properties of the other tasks -- the tasks that this task is potentially exclusive with.

- | | |
|----------------|--|
| universal | This task will not schedule concurrently with any other task |
| user attribute | Specify a specific attribute/value pair. This task will not schedule concurrently with any task that has this attribute/value pair in its list of "user attributes". |
| resource type | specify a resource type ("Labor", "Zone", etc.). This task will not schedule concurrently with any task using any resource of this type. |
| resource name | specify a resource by name. This task will not schedule concurrently with any task using a resource with this name. |
| task type | (e.g., "Milestone"). This task will not schedule concurrently with any task with this type. |

Figure 9 below shows the network diagram with the non-concurrent constraints as red arrows, emanating from the highlighted task that is non-concurrent to the tasks at the end of the red arrows.

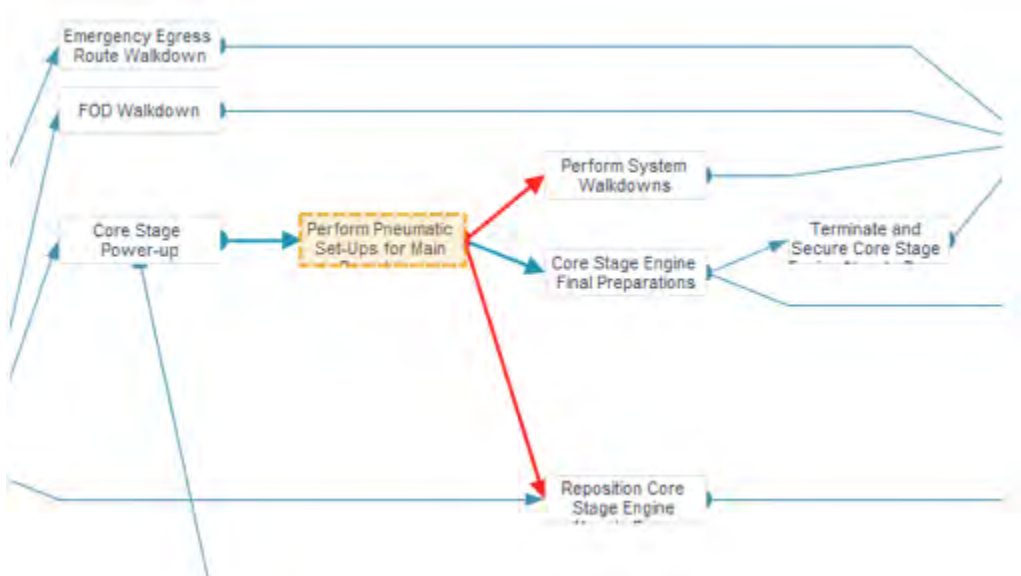


Figure 9. Non-concurrent constraints shown with red arrows

Aurora includes various filters that can be used to signify different
Figure 10 below shows the network diagram with the non-concurrent constraints as red arrows, emanating from the highlighted task that is non-concurrent to the tasks at the end of the red arrows.

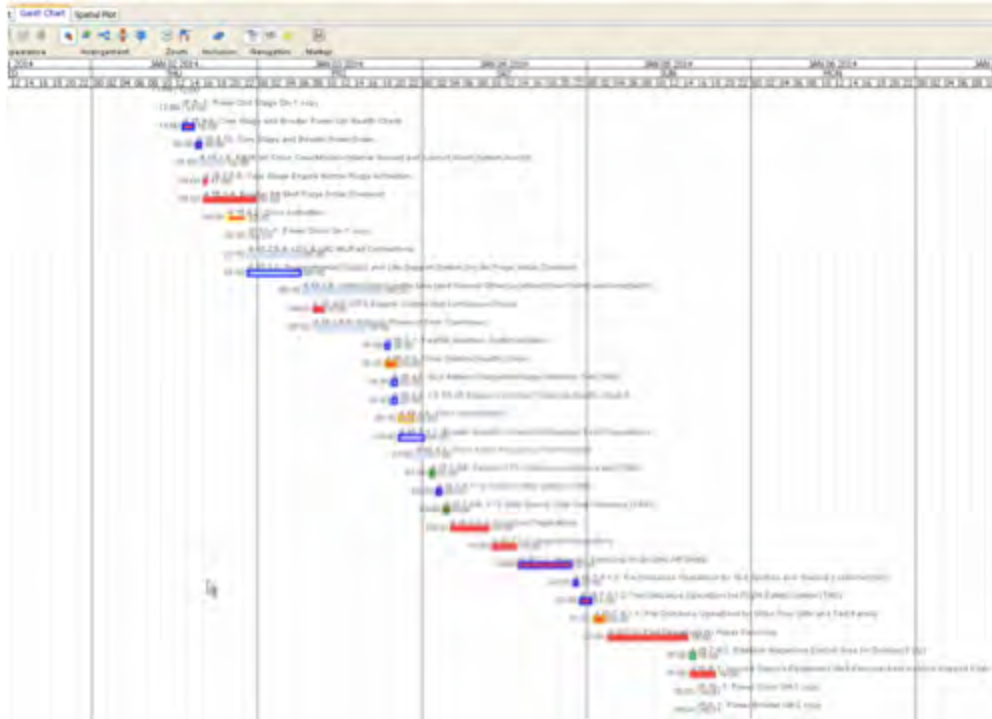


Figure 10. Gantt chart color-coded for non-concurrent constraints and other requirements

Conflict Reports

Aurora includes a feedback to help users understand the source of conflicts when they arise. Figure 11 shows a conflict report.

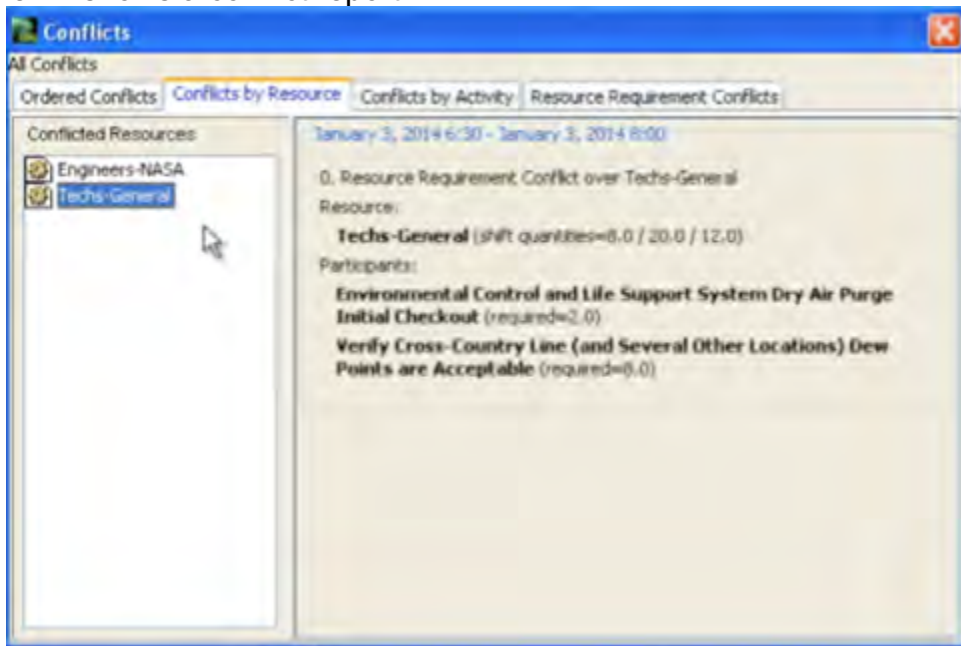


Figure 11. Conflict report

Aurora Real World Scheduling Comparisons

Boeing selected Aurora initially for the final assembly scheduling of the Dreamliner 787 aircraft due to its superior scheduling. Boeing was kind enough to provide a subset of real data that Stottler Henke is permitted to share. Even though this subset is much simpler than the actual project, it still reveals the significant difference between the scheduling results. Figure 12 shows the results of scheduling the exact same resource-loaded Boeing file with different software.

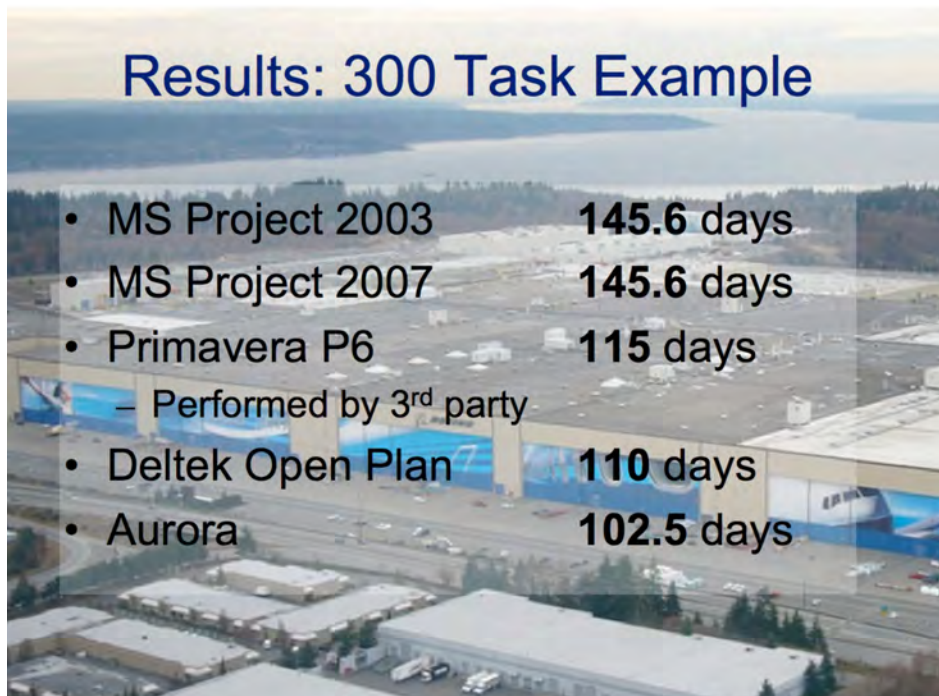


Figure 12. Scheduling results – aerospace model

Finally, let's consider the analysis of a real refinery turnaround project. Note that no Microsoft Project results are provided because the MS Project software could not successfully resource-level this project.

The project network consists of over 2,500 activities. A view of the network is shown in Figure 16. Note the red lines link tasks with Finish to Start constraints, this network also has some start-to-start constraints that are shown with yellow lines, some may be seen in the upper-left portion of the network shown in Figure 13

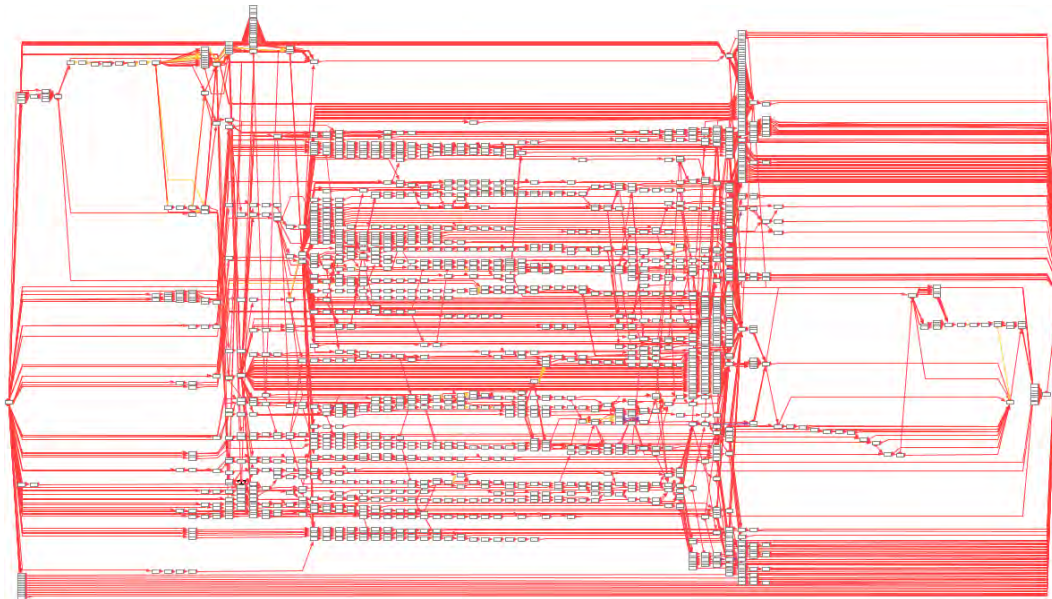


Figure. 13 Turnaround project network

The results of the analyses are shown in Figure 14.

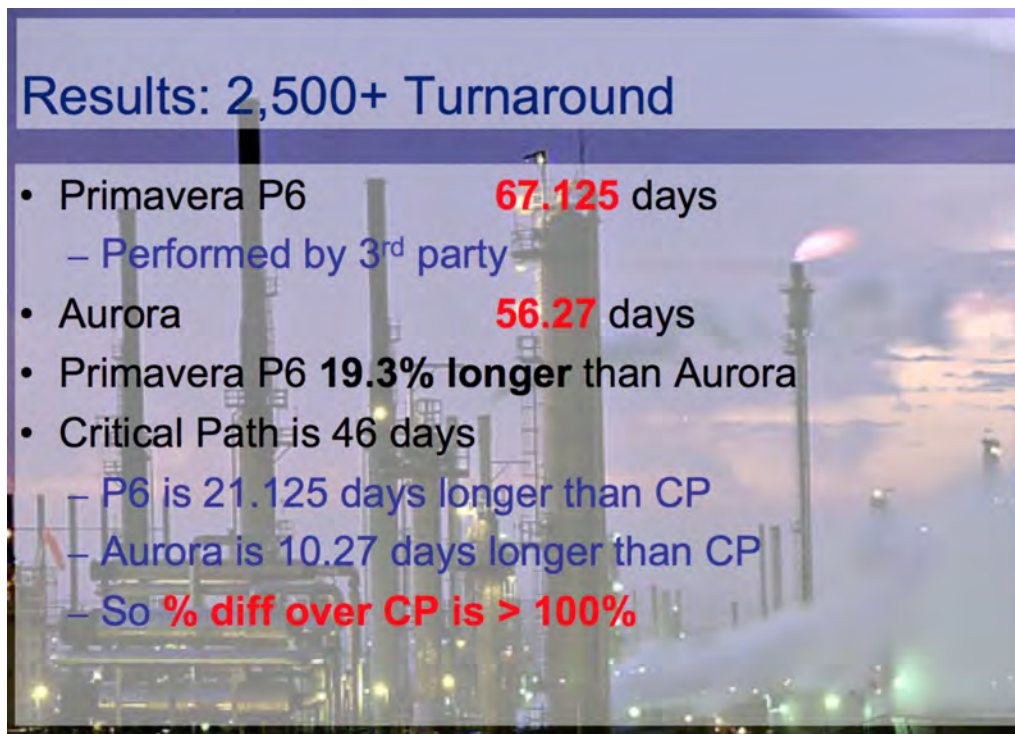


Figure 14. Scheduling results – refinery project

The difference in absolute terms is over 10.5 days. There are a few ways to compare

these results; the simplest is to simply compare overall durations, using the Aurora's intelligent scheduling results as the basis:

Primavera P6 resource leveling is **19.3% longer** than intelligent scheduling
(67.125 - 56.27) / 56.27.

Using the Primavera P6 resource-leveling as the bases:

Intelligent scheduling is **16.2% shorter** than Primavera P6 resource-leveling
(67.125 - 56.27) / 67.125

Another valuable perspective lies in comparing the resource-constrained result with the Critical Path, that is, the situation assuming unlimited resources. Why is this perspective valuable? Because the Critical Path is the best case scenario, and the valid schedule when considering resources must always be longer than the Critical Path, so the length longer than the Critical Path is the only portion of the total project duration that the resource-leveling or intelligent scheduling can affect.

The Critical Path for the refinery turnaround project is **46 days**.

Primavera P6 resource-leveling results longer than Critical Path:	21.125 days
Percent longer than Critical Path	45.9 %

Intelligent scheduling results longer than Critical Path:	10.27 days
Percent longer than Critical Path	22 %

The percent difference between days more than Critical Path for Primavera P6 versus intelligent scheduling is

105.70%.

These results demonstrate the significant benefit of leveraging Aurora's intelligent scheduling. Recall that everything besides the method for scheduling is the same in both cases. Leveraging Aurora saved over 10.5 days, and all of the associated costs with all the resources that are needed, as well as the lost revenue from the refinery being unavailable.

Of course the cost savings and other benefits of leveraging intelligent scheduling are huge for the initial plan, but even more potential benefit comes in the *execution phase* of the project, where unexpected circumstances need to be dealt with. By leveraging intelligent scheduling, rescheduling can be done quickly and the updated schedule will be shorter than if one used resource-leveling only. Therefore, every time a reschedule is performed, the overall benefit of leveraging intelligent scheduling increases.

Conclusions

It is likely that most *users of commercial project management software are NOT aware that the results from the resource leveling process are not optimal, and could be*

improved upon significantly. It is unfortunate that project teams that have put in the significant effort and cost to create a resource-constrained model may not know that they could potentially reap huge time and cost savings simply by running their already built model through a different scheduling engine.

Therefore the benefits of intelligent scheduling grows as the complexity of the scheduling challenge increases. All

Air Force Maintenance, Repair, and Overhaul (MRO) can benefit from intelligent scheduling.

Beyond this, CBM adds another layer of complexity to MRO, so

combining intelligent scheduling with CBM for Air Force MRO can maximize the benefits to the US Air Force.

Aurora has already provided disruptive benefits to a diverse range of domains and is already proven for MRO. Aurora is a proven solution that can be implemented quickly to significantly improve the US Air Force's MRO throughput.