Intelligent Scheduling at NASA: Application to Ground Operations at Kennedy Space Center

Steven Xu
Stottler Henke Associates, Inc.
1650 S. Amphlett Blvd., Suite 300
San Mateo, CA 94402
sxu@StottlerHenke.com

Richard Stottler
Stottler Henke Associates, Inc.
1650 S. Amphlett Blvd., Suite 300
San Mateo, CA 94402
stottler@StottlerHenke.com

Robert Richards
Stottler Henke Associates, Inc.
1650 S. Amphlett Blvd., Suite 300
San Mateo, CA 94402
Richards@StottlerHenke.com

***Abstract*— NASA's Kennedy Space Center (KSC) is working with Stottler Henke to improve KSC's Ground Processing (GP) scheduling. The solution called, *Aurora-KSC*, has been designed, developed and deployed at KSC to automate a large amount of Kennedy Space Center's planning, scheduling, and execution decision-making. Aurora-KSC leverages *Aurora,* the world's most intelligent scheduling framework; Aurora itself originated in part from many earlier NASA-funded efforts and has been utilized by NASA for some of its most complex scheduling challenges, including the scheduling of the maintenance, repair & overhaul (MRO) of the Space Shuttle during its tenure. Aurora-KSC expands on this foundation, enhancing and extending Aurora for improved generalization and adaptability. This includes an interface with Primavera P6 and Microsoft Project so project information already available in these formats can benefit from Aurora-KSC. This allows KSC to realize significant efficiency improvements in several different areas. Aurora-KSC also includes adaptive execution, that is, executing the scheduled activities/procedures intelligently as actual execution actually deviates from the original schedule. Aurora-KSC is available to the wider NASA community to provide its many benefits to the rest of NASA.**

Aurora-KSC is being applied to several problems at KSC relating to Ground Processing (GP). The delivered application improves the scheduling of Space Launch System (SLS) Processing and verification & validation (V&V) activity, including reduced turnaround time in response to changes and what-ifs, and more optimal schedules.

This paper reviews the goals that lead to the development of Aurora-KSC, including background on the pre-existing proven technologies that were leveraged. The application to NASA challenges will be discussed with benefits to date. Finally, the benefits of leveraging Aurora-KSC for other NASA applications will be discussed.

TABLE OF CONTENTS

1. INTRODUCTION

Historically, Kennedy Space Center (KSC) has had some of the most complex, enormous, difficult, diverse, distributed, and unique set of integrated scheduling problems in the world. And it is only getting more difficult as ground operations are requiring the sharing of resources between/among separate organizations (i.e. commercial launch and vehicle providers). KSC has some of the most important, expensive, and unique resources in the world used for launching and launch preparations of vehicles as well as for payload processing. They include launch pads, mobile launchers, crawlers, VAB high-bays, general and specialized processing facilities, Launch Equipment Test Facilities, etc., not to mention many smaller facilities, resources, and manpower. It is therefore important to utilize these resources as efficiently as possible.

Meanwhile, these resources will have to be shared by different organizations and different types of vehicles. NASA will have the Orion and SLS (which itself will have several variants), SpaceX has the Dragon and Falcon 9 (and other Falcon variants), and ULA Orbital Science Corp, Blue Origin, Boeing, and Sierra Nevada Corp all have the potential to deliver different vehicles to KSC for launch. This evolving situation has created new challenges where major resources have to be efficiently reconfigured for different vehicles, and competing commercial interests will have to cooperate in their use of shared resources, which

978-1-5090-1613-6/17/$31.00 ©2017 IEEE

inherently requires a geographically distributed and mobile scheduling concept. The problem is further complicated by the wide variety of time scales. Launch manifests are planned many years in advance to allow sufficient time to produce the launch vehicles and payloads, while daily ground operations are often planned down to the minute and countdowns down to the second.

NASA addresses this massive scheduling undertaking hierarchically and in a distributed/integrated manner. At the highest level, the manifest schedule may extend 5 to 10 years into the future and address just the most major resources (pads, VAB high-bays, crawlers, mobile launchers, and large processing facility floor space needs) and the most important dates (pad rollout, launch, etc.). Each mission is further detailed as the launch date approaches, typically with the vehicle processing being planned in a primarily *forward* manner by one group, while payload processing is being planned *backward* from the launch dates by a different group. The high-level payload and vehicle processing schedules are further broken down and detailed by managers of the various facilities and manpower typically down to the hour or even minute by the time the processing actually is executing. At each level there is some coordination both vertically (between different levels of detail) and horizontally (among different schedules at the same level of detail that share resources or otherwise depend on each other).

So the scheduling problem is decomposed into a dizzying myriad of individual but coordinated scheduling problems—each with its own unique set of resources, tasks, constraints, ground rules, and scheduling techniques. The natural result is that the scheduling process is different for each of the individual applications. This has been traditionally addressed at KSC with semi-automation—human experts making the scheduling decisions (with a very small number of notable exceptions) while using graphical editing tools that may or may not do some rudimentary computations, such as pushing tasks later when an earlier linked task has been delayed. The uniqueness of each individual scheduling application, along with the general uniqueness of KSC's scheduling problems (e.g., KSC tends to have constraints unlike anyone else's) when compared to the world at large, has also meant that KSC scheduling has resisted any kind of single, global solution. This has historically resulted in a large number of scheduling systems, each with varying degrees of automation and sets of features. Most recently, Primavera P6 had been chosen as the scheduling system for purposes of rendering, coordination, and sending of schedules between different organizations. However, because Primavera cannot handle modeling all of KSC's constraints, the schedules it tries to generate automatically is almost certainly not correct and thus must be checked by an expert for violations of un-modeled constraints. Furthermore, even setting aside this issue, the schedules it generates are not very optimal. For example, in three separate studies, Primavera-generated resource-loaded schedules took an average of over 18% longer than those generated by Aurora, when given exactly the same set of tasks, constraints, and resources as the Aurora-generated schedules – and these models are significantly simpler than the scheduling challenges of KSC schedules. Given the importance of launch deadlines, this forces the highly expert schedulers to make the scheduling decisions themselves, consuming scarce specialized manpower and adding to turnaround time. For example, to respond to an unplanned work issue involving adding just one or a few unplanned tasks to the schedule, a ground processing scheduler estimated that it would take 30 minutes for him to draw the schedule on paper and hand it to a Primavera P6 operator to render, where an automated system should require just a few minutes for entry at most (and that being if new types of resources or new types of constraints must be defined) and only seconds to reschedule.

NASA's "Ground and Launch Systems Processing Roadmap, Technology Area 13," dated April, 2012, states, "While some functions of planning and scheduling systems are automated today, much of this activity is labor-intensive" and then goes on to explain the need at KSC for automatic scheduling systems to "optimize the use of resources during ground operations" and "reduce the number of planners."

2. Current / Recent Situation

Because of all the complexity and change, a large number of highly skilled schedulers make scheduling decisions and maintain the payload processing schedules. NASA has investigated existing advanced planning tools to try to automate the scheduling process, but all these except Aurora have been found to be inadequate to address the complexity of the problem. The schedulers do make use of graphical scheduling editor tools and do utilize a shared common scheduling database, but nothing is in use that mimics the *decision-making process of the expert schedulers*.

As mentioned previously, KSC has designated Primavera P6 as the standard tool for graphically rendering schedules and transmitting them between different organizations. Primavera does include a rudimentary resource scheduling capability, but it performs poorly and because it cannot model many of the constraints found at KSC, its schedules are essentially invalid anyway. Even in (non-KSC) cases where Primavera can correctly model the problem, the schedules it generates are far from optimal.

3. FOUNDATION FOR IMPROVEMENTS

Stottler Henke has been working with NASA, and especially KSC, to improve the efficiency of its projects and other scheduling challenges since the 1990s. One of the projects completed in 1994, developed techniques for long-term Space Shuttle processing planning for NASA's Kennedy Space Center. Experienced mission planners were studied to identify relevant planning techniques, heuristics, and data. Their knowledge was captured using a combination of rules and object-oriented representations. Techniques were developed to automate the multi-mission planning 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 maintain manifests and perform advanced "what-if" studies. This project was the genesis of Stottler Henke's intelligent entities approach to planning and scheduling.

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 [1] [2] was created in such a way that every *decision point* that could be changed in a scheduling system is very easy to modify. Figure 1 shows a high-level representation of Aurora's Architecture.

To achieve this flexibility, we designed it to have a number of components that could be plugged in and matched 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 2 for a more detailed breakdown of configurable operations.

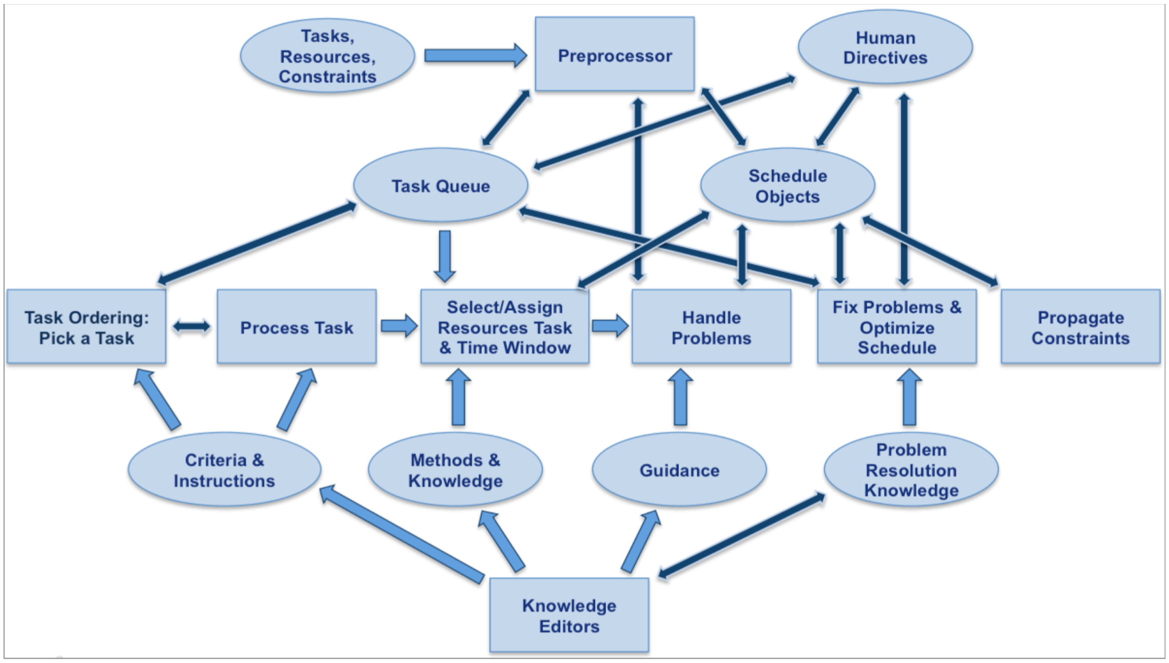


Figure 1. High-Level Aurora Architecture

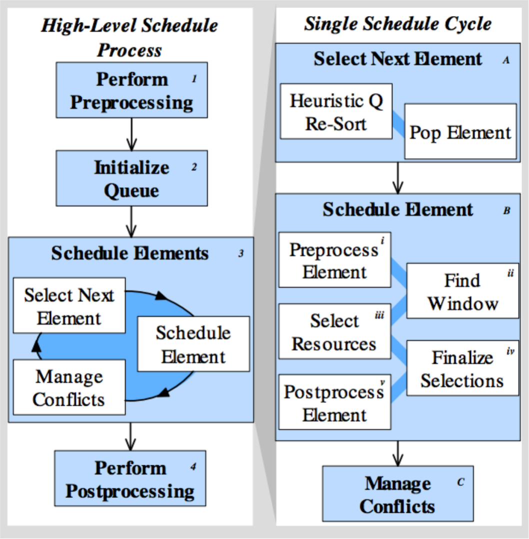
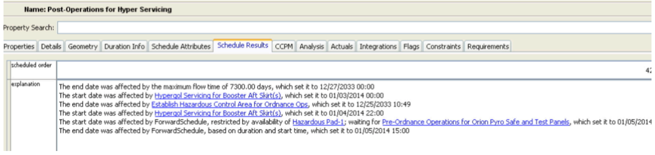


Figure 2. Aurora's reconfigurable scheduling system process breakdown.

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 [3] & scheduling related to space debris tracking [4], 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 [5] as well as similar operations for Bombardier and Learjet, to combining intelligent scheduling with Critical Chain Project Management (CCPM) [6], to scheduling the manufacturing facilities of pharmaceutical production.

Due to the past successes with NASA and the enhancements per the myriad of other applications of Aurora over the years, the then current Aurora framework was selected as the foundation for NASA's latest challenges. That is, Aurora was already proven as the intelligent planning and scheduling system that enabled NASA to solve some its most complex scheduling problems. Aurora's intelligent planning and scheduling has consistently generated more optimal schedules in every application it has been applied to both at KSC and every other domain it has been applied.

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 3 shows an example of an explanation.


Name: Post-Operations for Hyper Servicing

Property Search:

scheduled order 42

explanation
The end date was affected by the maximum flow time of 7300.00 days, which set it to 12/27/2033 00:00
The start date was affected by Hypergol Servicing for Booster Aft Skirt(s), which set it to 01/03/2014 00:00
The end date was affected by Establish Hazardous Control Area for Ordnance Ops, which set it to 12/25/2033 10:49
The start date was affected by Hypergol Servicing for Booster Aft Skirt(s), which set it to 01/04/2014 22:00
The start date was affected by ForwardSchedule, restricted by availability of Hazardous Pad-1; waiting for Pre-Ordnance Operations for Orion Pyro Safe and Test Panels, which set it to 01/05/2014
The end date was affected by ForwardSchedule, based on duration and start time, which set it to 01/05/2014 15:00

Figure 3. 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 4 shows this functionality where the user clicked at the point in time represented by the vertical dashed line.

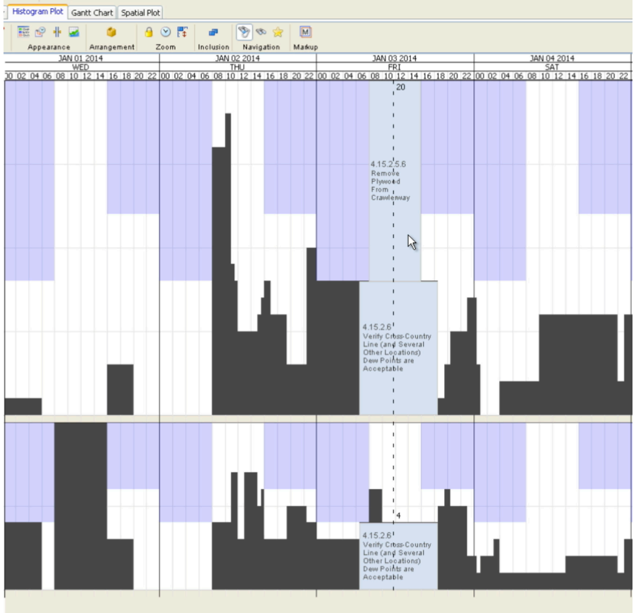

Histogram Plot
Gantt Chart
Spatial Plot
Appearance
Arrangement
Zoom
Inclusion
Navigation
Markup
JAN 01 2014
WED
JAN 02 2014
THU
JAN 03 2014
FRI
JAN 04 2014
SAT
00 02 04 06 08 10 12 14 16 18 20 22
20
4.15.2.5.6 Remove Plywood From Crawlerway
4.15.2.6 Verify Cross-Country Line (and Several Other Locations) Dew Points are Acceptable
4
4.15.2.6 Verify Cross-Country Line (and Several Other Locations) Dew Points are Acceptable

Figure 4. 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 5 shows a spilt view showing a Gantt chart.

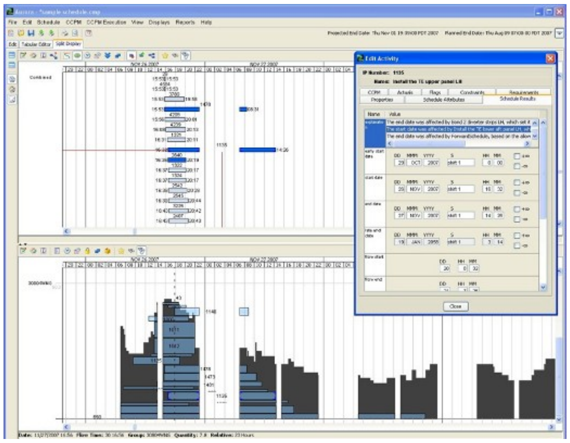


Figure 5. Split view showing Gantt chart same time slice as histogram, showing activities constituting resource need for one time instance

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 6 shows a network diagram and a single-element view shown in a separate window.

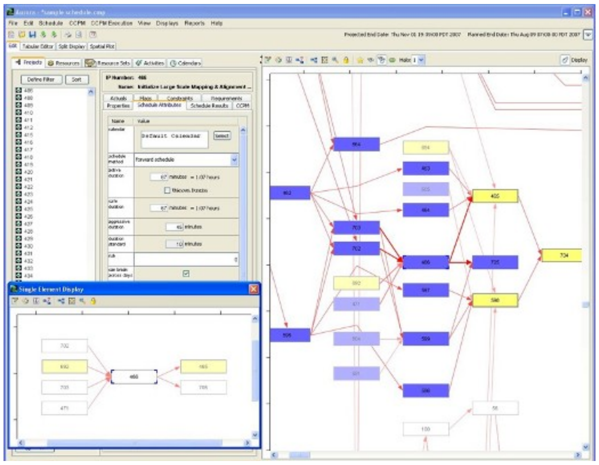


Figure 6. Network diagram showing single-element view option

From this foundation, that goal was to develop *Aurora-KSC* to support Ground Operation scheduling. Ground Operations scheduling consists of overlapping missions at KSC that compete for the same resources, as well as, ground rules, safety requirements, and the unique needs of processing vehicles and payloads destined for space which impose numerous complex constraints that must be satisfied by the schedules. Since the equipment and facilities required to carry out these operations are extremely expensive and limited in number, optimal assignment and efficient use are critically important.

4. What if / Execution Issues

There are several requirements of KSC scheduling problems that make these problems unique and extremely complex and, thus, not further automatable by existing commercial scheduling tools. Unlike most applications, much of the equipment and many elements to be processed must be explicitly scheduled to be stored, while awaiting further processing or departure. This is usually done with dedicated Storage, Dwell, or Float activities. Typically, the activities to be scheduled are complex in that they have several resources, several temporal links to other activities, and many rules of thumb and/or additional constraints that influence the scheduling decisions. Overall, there are a very large number of scheduling constraints caused by ground rules, safety requirements, and the unique needs of processing vehicles and payloads destined for space. For example, some operations are determined to be hazardous, meaning that when they are being performed, other activities cannot occur in a defined volume of space. There are also a large number of activities. Since several missions are in preparation simultaneously, they all compete for scarce resources. Simultaneous SLS processing would be a factor as multiple SLS missions will be required for a manned trip to Mars. But even a single SLS mission will have to compete in terms of resources such as manpower, equipment, and hazardous areas with other non-SLS missions that are active at the same time. And change is a constant phenomenon.

Additionally, many what-if scheduling studies are requested. Many of the resources are operating at or beyond their capacity. This, combined with the fact that many of the resources are extremely expensive, often with very long lead times, offers a strong incentive to create optimal schedules.

Because scheduling must be done so far in advance, many of the processing requirements are immature and therefore will undergo significant evolution during the course of several scheduling episodes. KSC's need to schedule its processing floor space, along with a large number of spatial constraints, is unique. These unique requirements mean that no existing commercial scheduling tools are adequate to further automate the scheduling process. Beyond the hazardous areas described above, other unique scheduling constraints relate to states (the need for a facility or piece of equipment to be in a certain state to allow certain tasks to occur (and where it is understood that state will disallow others at the same time)), utilities (a resource that can support an unlimited number of activities as long as it is operating, such as power or helium), work space (the space taken up by workers and equipment as they perform their tasks), and weight (i.e., particular platforms can only support a specified maximum weight (and where weight encompasses that of workers and equipment)).

5. Automatic Scheduler Capabilities Needed for Ground Processing

Stottler Henke worked with KSC schedulers and other subject matter experts (SMEs) to understand the all the challenges and thus the capabilities required for intelligent optimized Ground Processing scheduling. Although the majority of NASA's needs were already met by the then current version of Aurora, some additional capabilities were identified:

- Hazardous constraints
- Interim Problem Report (IPR) insertion/deletion
- Enhanced import/export with Primavera P6
- Import from Ground Operations Planning Database (GOPDb)
- Scheduling and Display in Seconds in Addition to Minutes, Hours, and Days
- Reference Tasks from other files
- Preferred versus required temporal constraints
- KSC Specific Displays, Printing and PDF Export
- Scheduling Algorithm Adjustments
- Miscellaneous user interface efficiency enhancements.

The Aurora framework was modified and enhanced to create the Aurora-KSC version with these capabilities. The following provides some more details regarding these capabilities.

Hazardous constraints

Aurora-KSC added the capability to mark activities as being ‘hazardous’ to other activities. The result of such a hazardous marking means that Aurora will never schedule the hazardous activities to occur simultaneously with any of the activities it is hazardous to. Graphical enhancements now allow for hazard activities to be denoted in the PERT Chart, with special arrows emanating from the activity causing the hazard and pointing to the activities affected.

Aurora already had the concept of both concurrent constraints and non-concurrent constraints. Figure 7 shows non-concurrent constraint for tasks A, B and Figure 8 shows concurrent constraints for task B, A, & C.

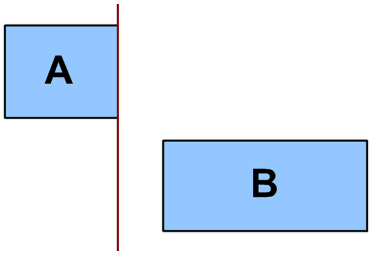


Figure 7. Non-concurrent tasks

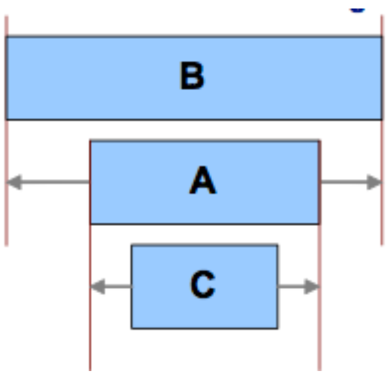


Figure 8. Concurrent tasks

So the hazardous constraint is a variation of the non-concurrent constraint.

Interim Problem Report (IPR) insertion/deletion

An Interim Problem Report (IPR) is modeled in Aurora as an activity that is inserted when something goes wrong during execution of an activity. Aurora-KSC supports the easy insertion and deletion of IPRs; the user specifies how long the activity had been running and the type of IPR, and Aurora will split the activity into two pieces at that time, inserting an IPR in the middle and preserving all constraints. Likewise, on deletion, Aurora will merge the two split halves back into a single activity, and remove all redundant constraints.

Enhanced Import/export with Primavera P6

Aurora already supported translation to and from Primavera P6 as well as Microsoft Project. However, translators always need to be maintained and not all information is translated between the programs as each program may store information the other program does not use or need. There was extensive testing and updating during the development of Aurora-KSC to support the import/export of Primavera projects in XML format. The Aurora import/export code translates between Primavera activities, resources, relationships, and properties and their Aurora counterparts. Some items that were not already supported during translation, such as Resource Rates, required specialized conversion to be used effectively in Aurora-KSC.

The translation functionality provides great flexibility to the user so that they can use the capabilities of Aurora and Primavera P6 in a manner that is best for them. For example, one use case is to take a model that is already built in Primavera P6 entrance late at once into Aurora and then continue all the planning in Aurora, and execute the project using only Aurora. Another use case is to build a model in Primavera P6, then import the model into Aurora, refining the model as needed in Aurora so that it includes the details that could not be modeled in Primavera. Then schedule in Aurora and start execution of the project. This Aurora model can then be transferred back to Primavera, so that users can update the status of tasks in Primavera and use any other functionality of Primavera they wish. In this use case users update the status of tasks in Primavera and then use a special translation capability provided by Aurora that extracts the execution mode updates to update the Aurora model appropriately, allowing for only Single entry of data. This use case is shown below in Figure 9.

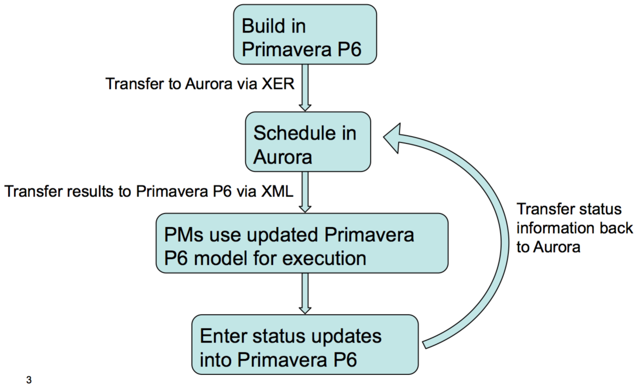

Build in Primavera P6
Transfer to Aurora via XER
Schedule in Aurora
Transfer results to Primavera P6 via XML
PMs use updated Primavera P6 model for execution
Enter status updates into Primavera P6
Transfer status information back to Aurora
3

Figure 9. Potential workflow with Primavera P6 showing use case where execution updates are done in P6

Import from Ground Operations Planning Database (GOPDb)

Extensive work was also done both to import the 20+ sections of the GOPDb into appropriate Aurora fields and structures and to export those to Primavera. The GOPDb contained the first version of the entire SLS and Orion Processing Flow, broken into sections primarily based on the facility where the work was occurring. This involved translating GOPDb codes to actual resources and manpower types, hazards, resources, and temporal constraints, as well as preserving other information stored in GOPDb records. Figure 10 shows a small example of Pad Ops Section imported from the GOPDb.

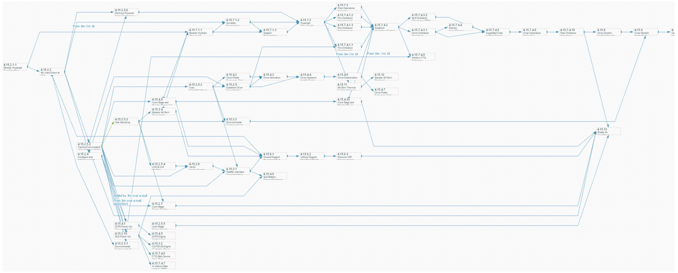


Figure 10. Network diagram

Scheduling and Display in Seconds in Addition to Minutes, Hours, and Days

Aurora was modified to handle scheduling and display of activities at the seconds-level, to support short-duration activities towards the end of launch countdown.

Reference Tasks from Other Files

A project file may reference a task scheduled in another file. I.e., this capability allows linking dependencies between separate projects that are kept in separate files.

Preferred versus required temporal constraints

Temporal Constraints, which are normally considered to be a requirement, can be marked as Preferred and given a 0.0 to 1.0 Importance. Preferred constraints are honored if there is enough time in the schedule, but they can be broken, if need be, to meet required constraints (such as deadlines). Algorithmic logic was created to break constraints (least important first) to fix a broken schedule.

KSC Specific Displays, Printing and PDF Export

KSC has very specific requirements for what information is displayed on PERT and Gantt charts (e.g., L – time) as well as how it is rendered. This necessitated leveraging Aurora's already extensive filtering and coloring capabilities, to create the correct rules for these displays and adding accompanying printing and PDF Export capabilities. Additionally, to aid the planners in finding mistakes in their scheduling models and to aid them in understanding those models as well as how the models affected scheduling decisions, additional display and printing capabilities were added. Figure 11 shows a Gantt chart color-coded for hazardous and powering requirements.

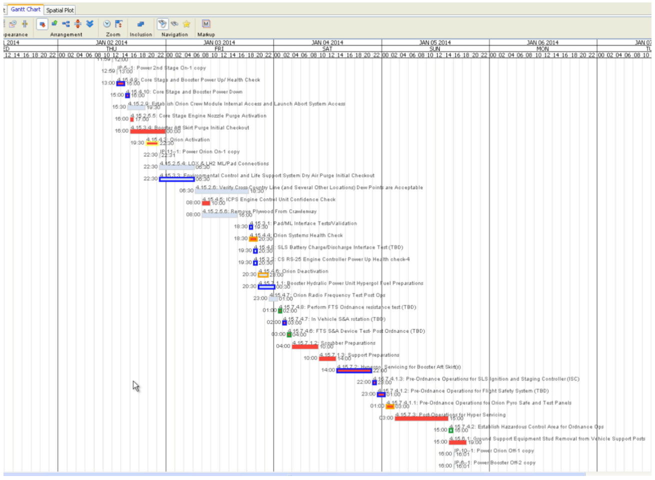

Gantt Chart
Spatial Plot
Appearance
Arrangement
Zoom
Inclusion
Navigation
Markup
JAN 02 2014 THU
JAN 03 2014 FRI
JAN 04 2014 SAT
JAN 05 2014 SUN
JAN 06 2014 MON
IP-5--1: Power 2nd Stage On-1 copy
4.15.4.9: Core Stage and Booster Power Up/ Health Check
4.15.4.10: Core Stage and Booster Power Down
4.15.2.9: Establish Orion Crew Module Internal Access and Launch Abort System Access
4.15.2.5.5: Core Stage Engine Nozzle Purge Activation
4.15.3.4: Booster Aft Skirt Purge Initial Checkout
4.15.4.3: Orion Activation
IP-11--1: Power Orion On-1 copy
4.15.2.5.4: LOX & LH2 ML/Pad Connections
4.15.3.3: Environmental Control and Life Support System Dry Air Purge Initial Checkout
4.15.2.6: Verify Cross-Country Line (and Several Other Locations) Dew Points are Acceptable
4.15.4.5: ICPS Engine Control Unit Confidence Check
4.15.2.5.6: Remove Plywood From Crawlerway
4.15.3.1: Pad/ML Interface Tests/Validation
4.15.4.4: Orion Systems Health Check
4.15.4.8: SLS Battery Charge/Discharge Interface Test (TBD)
4.15.3.2: CS RS-25 Engine Controller Power Up Health check-4
4.15.4.6: Orion Deactivation
4.15.7.1.1: Booster Hydralic Power Unit Hypergol Fuel Preparations
4.15.4.7: Orion Radio Frequency Test Post Ops
4.15.7.4.8: Perform FTS Ordnance resistance test (TBD)
4.15.7.4.7: In Vehicle S&A rotation (TBD)
4.15.7.4.6: FTS S&A Device Test- Post Ordnance (TBD)
4.15.7.1.2: Scrubber Preparations
4.15.7.1.3: Support Preparations
4.15.7.2: Hypergol Servicing for Booster Aft Skirt(s)
4.15.7.4.1.3: Pre-Ordnance Operations for SLS Ignition and Staging Controller (ISC)
4.15.7.4.1.2: Pre-Ordnance Operations for Flight Safety System (TBD)
4.15.7.4.1.1: Pre-Ordnance Operations for Orion Pyro Safe and Test Panels
4.15.7.3: Post-Operations for Hyper Servicing
4.15.7.4.2: Establish Hazardous Control Area for Ordnance Ops
4.15.6.1: Ground Support Equipment Stud Removal from Vehicle Support Posts
IP-10--1: Power Orion Off-1 copy
IP-6--1: Power Booster Off-2 copy

Figure 11. Gantt chart color-coded for hazardous and powering requirements

Scheduling Algorithm Adjustments

The addition of the above capabilities as well as special KSC ground processing circumstances and constraints necessitated additions and modifications to the automatic scheduling algorithms. For example, the concept of hazardous activities was added to the modeling capabilities, and as a result, the scheduling algorithms had to be adjusted to make sure this new constraint was honored.

Miscellaneous user efficiency interface enhancements

Several capabilities were added to make the schedulers more efficient. Examples include batch editing and right click menu options. Figure 12 below shows the network diagram with the hazardous constraints as red arrows, emanating from the highlighted task that is hazardous to the tasks at the end of the red arrows.

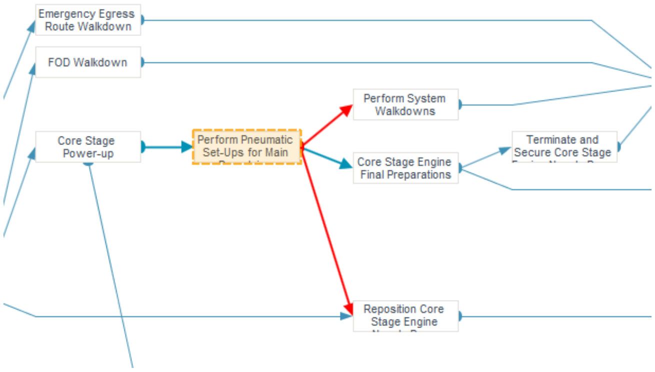
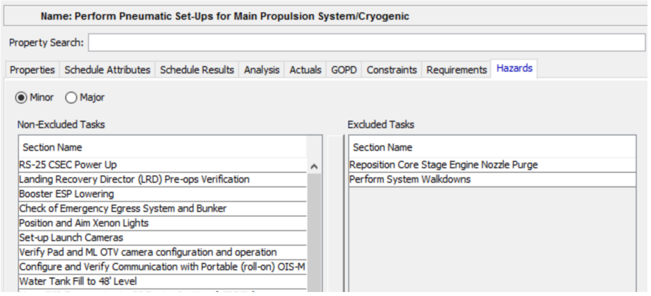

Emergency Egress Route Walkdown
FOD Walkdown
Core Stage Power-up
Perform Pneumatic Set-Ups for Main
Perform System Walkdowns
Core Stage Engine Final Preparations
Terminate and Secure Core Stage
Reposition Core Stage Engine

Figure 12. Hazardous constraints shown with red arrows

Figure 13 shows one of the options to set up hazardous constraints via a new Hazards tab.



Name: Perform Pneumatic Set-Ups for Main Propulsion System/Cryogenic

Property Search:

● Minor ○ Major

Non-Excluded Tasks

Section Name
RS-25 CSEC Power Up
Landing Recovery Director (LRD) Pre-ops Verification
Booster ESP Lowering
Check of Emergency Egress System and Bunker
Position and Aim Xenon Lights
Set-up Launch Cameras
Verify Pad and ML OTV camera configuration and operation
Configure and Verify Communication with Portable (roll-on) OIS-M
Water Tank Fill to 48' Level

Excluded Tasks

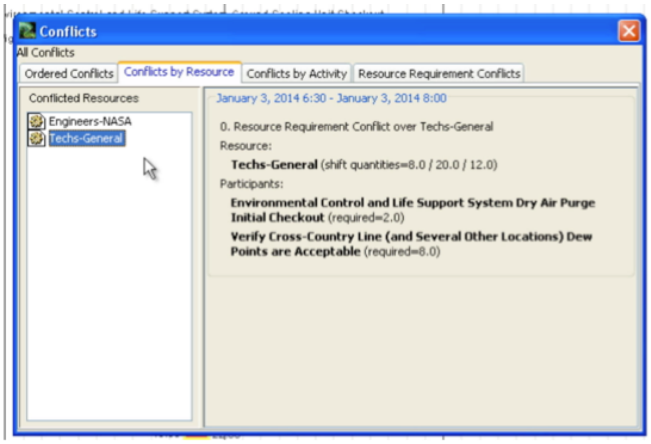
Section Name
Reposition Core Stage Engine Nozzle Purge
Perform System Walkdowns

Figure 13. Hazards tab for setting up hazardous constraints

6. Benefits To NASA

Although Aurora-KSC has been designed to solve some incredibly complex scheduling challenges unique to Kennedy Space Center, the solution was integrated with the rest of the Aurora solution so that all of NASA now has access to the intelligent project management and scheduling of Aurora. That is, the foundation for Aurora-KSC, was Aurora which was already the world's most advanced intelligent scheduling project management tool, and the result is now even more powerful with even more NASA specific capabilities.

So now NASA has the option to utilize Aurora for more of its complex scheduling challenges, both at KSC and any other NASA locations. Aurora can be used in conjunction with current solutions, for example if users prefer to model in their current tool they can continue to do so and then transfer the model to Aurora for enhancement and improved resource scheduling. Aurora, of course, can be used completely stand-alone, since it is a complete project management solution, and the model can be built directly in Aurora if desired or data can be pulled from various sources ranging from spreadsheets to enterprise databases. Figure 14 shows a conflict report.



Conflicted Resources

Engineers-NASA
Techs-General

January 3, 2014 6:30 - January 3, 2014 8:00

0. Resource Requirement Conflict over Techs-General

Resource:

Techs-General (shift quantities=8.0 / 20.0 / 12.0)

Participants:

Environmental Control and Life Support System Dry Air Purge Initial Checkout (required=2.0)

Verify Cross-Country Line (and Several Other Locations) Dew Points are Acceptable (required=8.0)

Figure 14. Conflict report

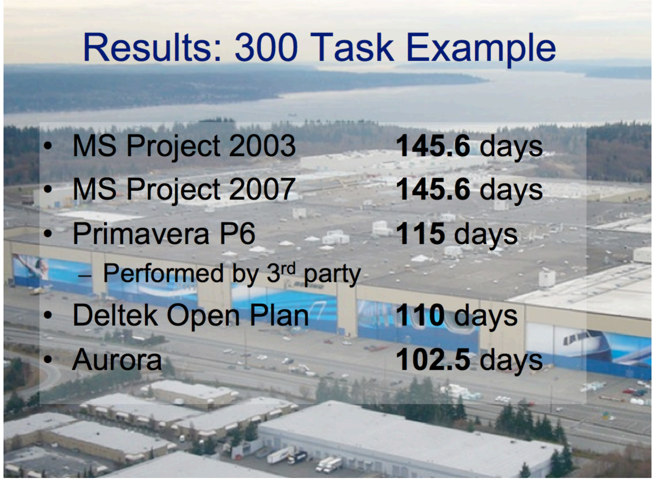
Aurora-KSC now provides a unique set of capabilities that NASA can leverage to increase efficiency across a wide a range of NASA operations.

- 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, 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 or hosted by NASA.

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) when compared with other scheduling methods. Aurora is an uncommon tool in that it combines sophisticated scheduling mechanisms with domain knowledge and case-based expert conflict resolution techniques to solve scheduling problems.

For example, 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 15 shows the results of scheduling the exact same resource-loaded Boeing file with different software.



Results: 300 Task Example

- MS Project 2003 **145.6 days**
- MS Project 2007 **145.6 days**
- Primavera P6 **115 days**
  - Performed by 3rd party
- Deltek Open Plan **110 days**
- Aurora **102.5 days**

Figure 15. Scheduling results – aerospace model

Finally, let's considers 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 16.

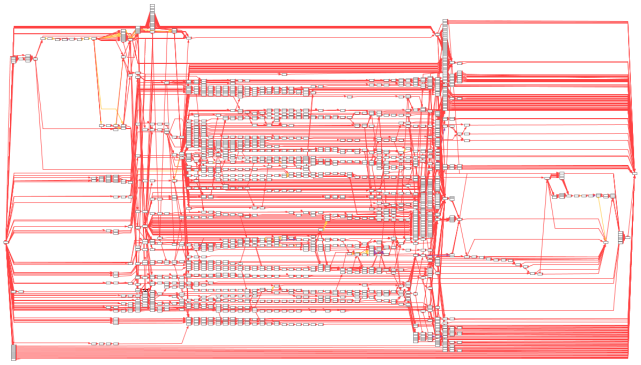
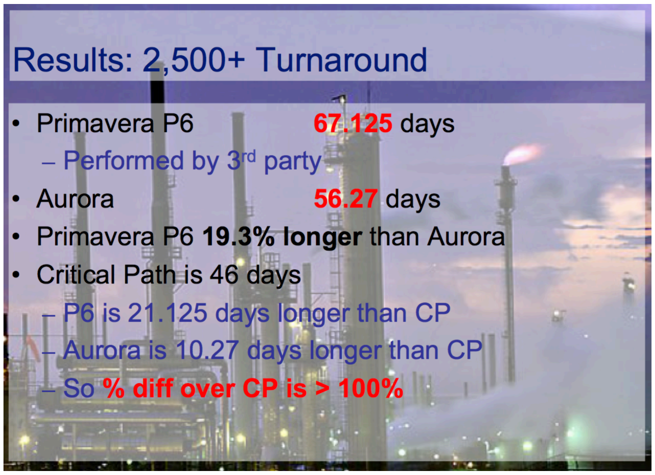


Figure. 16. Turnaround project network

The results of the analyses are shown in Figure 17.



Results: 2,500+ Turnaround

- Primavera P6 67.125 days
  - Performed by 3rd party
- Aurora 56.27 days
- Primavera P6 **19.3% longer** than Aurora
- Critical Path is 46 days
  - P6 is 21.125 days longer than CP
  - Aurora is 10.27 days longer than CP
  - So **% diff over CP is > 100%**

Figure 17. 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 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 intelligent scheduling. Recall that everything besides the method for scheduling is the same in both cases.

Leveraging intelligent scheduling 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.

7. CONCLUSIONS

Stottler Henke working in conjunction with NASA has been able to create an intelligent project management and scheduling solution that is not only significantly benefiting the ground processing at Kennedy Space Center, but also provides a general intelligent project management and scheduling solution that can be leveraged by other projects and scheduling challenges throughout NASA.

Some of the specific goals for KSC ground operations met by Aurora-KSC include:

- Saving the manpower of highly trained and highly skilled planners and schedulers and greatly improving their turnaround time to changes and requests.
- Automatically generating near-optimal SLS processing and assembly plans.
- Automatically near-optimally rescheduling in real-time in response to changes and requests.
- Supporting constraints unique to KSC, so models true to life can be built and scheduled.
- Supporting import from and export to Primavera,
- Supporting import from the GOPD, and
- Allowing editing, display, and printing of PERT and Gantt charts with to-the-second accuracy (as needed for later tasks in the launch countdown).

In addition, the entire NASA community can leverage Aurora-KSC for its myriad benefits including;

- Large multi-project support, able to handle 100,000+ tasks per project
- Multiple-pass intelligent resource-constrained scheduling, resulting in shorter projects and greater transparency.
- Mixed-mode scheduling, supporting both forward and backward scheduling, available on a task-by-task basis.
- Schedule explanations for each task providing greater understanding and transparency.
- Support for various constraint types, which allow for the correct modeling of NASA realities.

NASA and Stottler Henke have been working together for decades. NASA has benefited from the intelligent scheduling advances Stottler Henke has developed, and industry has also benefited. Beneficiaries include Boeing, Pfizer, Bombardier, Harvard Medical School, Alaska Airlines, Mitsubishi and various others. In addition, the US Air Force is leveraging Aurora's intelligent scheduling for the downlink scheduling of their satellites and various other aspects of satellite operations. Now NASA is coming full circle and benefiting from the many advancements that have been made to Aurora to meet the needs of other clients.

REFERENCES

[1] Kalton, A., R. Richards. (2008) Advanced Scheduling Technology for Shorter Resource Constrained Project Durations. AACE International's 52nd Annual Meeting & ICEC's 6th World Congress on Cost Engineering, Project Management and Quantity Surveying. Toronto, Ontario, Canada. June 29 – July 2, 2008.

[2] Kalton, A. (2006) Applying an Intelligent Reconfigurable Scheduling System to Large-Scale Production Scheduling. International Conference on Automated Planning & Scheduling (ICAPS) 2006. Ambleside, the English Lake District, U.K. June 6-10, 2006.

[3] Mohammed, J., Stottler, R., "Rapid Scheduling of Multi-tracking Sensors for a Responsive Satellite Surveillance Network," Proceedings of the Infotech@Aerospace 2010 Conference, Vol. 1, AIAA, Reston, VA, 2010.

[4] Stottler, R., Thompson, R., "Globally Optimized Scheduling for Space Object Tracking," Proceedings of the Infotech@Aerospace 2011 Conference, Vol. 1, AIAA, Reston, VA, 2011.

[5] Richards, R. (2010). Enhancing Resource-Leveling via Intelligent Scheduling: Turnaround & Aerospace Applications Demonstrating 25%+ Flow-Time Reduction. 2010 PMI College of Scheduling Conference PMICOS. Calgary, Canada. May 2-5, 2010.

[6] Richards, R. (2010). Critical Chain: Short-Duration Tasks & Intelligent Scheduling in e.g., Medical, Manufacturing & Maintenance. Proceedings of the 2010 Continuous Process Improvement (CPI) Symposium. Cal State University, Channel Islands. August 19-20, 2010.

[7] Richards, R. (2015) Packaging Line Scheduling Optimization. Pharmaceutical Manufacturing Vol 14 no 8 pp. 13-15, Oct 2015.

[8] Stottler, R., "General Scheduling Service using Intelligent Resource Management Techniques," Infotech@Aerospace 2010 Conference, American Institute of Aeronautics and Astronautics, Atlanta, GA, 2010.

[9] Stottler, R., Mahan, K., and Jensen, R., "Bottleneck Avoidance Techniques for Automated Satellite Communication Scheduling," Infotech@Aerospace 2011 Conference, American Institute of Aeronautics and Astronautics, Reston, VA (submitted for publication).

[10] Vallado, D., Crawford, P., "SPG4 Orbit Determination," Proceedings of the 2008 AIAA/AAS Astrodynamics Specialists Conference, AIAA, San Diego, CA, 2008.

[11] Barbulescu, L. , Howe , A., and Whitley, D., "AFSCN Scheduling: How the Problem and Solution Have Evolved," Mathematical and Computer Modeling, 2006.

[12] Miller, James G., "A new sensor allocation algorithm for the Space Surveillance Network," 74th MORS Symposium, 2006.

[13] Abbot, R.I., and Wallace, T.P. "Decision Support in Space Situational Awareness," Lincoln Laboratory Journal, Vol. 16, No. 2, 2007, pp. 297-335.

[14] Stottler, R., "Satellite Communication Scheduling, Optimization, and Deconflcition Using Artificial Intelligence Techniques," AIAA Infotech@Aerospace 2010, Atlanta, GA, 2010.

[15] Policella, N. (2005). Scheduling with Uncertainty: A Proactive Approach Using Partial Order Schedules. AI Communications 18. IOS Press.

[16] Robinson, H (2007, April). An Introduction to the Theory behind Goldratt's Critical Chain Project Management. Project Management Institutes College of Scheduling Conference (PMICOS) 2007, Vancouver, Canada

[17] Lu, M, Wu, D.P., and Zhang, J.P.(2006) "A Particle Swarm Optimization-Based Approach to Tackling Simulation Optimization of Stochastic, Large-Scale and Complex Systems", Lecture Notes in Computer Science, Vol 3930, pp. 528-537, Springer Berlin / Heidelberg

[18] Lam, H.C. Lu, M., (2006). Critical Path Scheduling Under Resource-Availability And Activity-Interruption Constraints Proceeding Of 2006 Annual CSCE Conference Of The Canadian Society For Civil Engineering, Page No: Ct-035 1-9, Calgary, Canada, May, 2006.

[20] Rizzo, A. (2006, May) Lean Enterprise Project Management. Product Development Institute Train-The Trainer Program, Everett, WA, USA.

BIOGRAPHY

***Steven Xu** is a Software Engineer at Stottler Henke. He has been involved in developing intelligent scheduling systems in several domains, ranging from ground and launch systems processing for NASA's Space Launch System to satellite communications scheduling for the Air Force and student class scheduling for the NYU College of Dentistry. More recently, he completed a project effort that augments intelligent scheduling with consumable resource management and the ability to interface with diagnosis, planning and adaptive execution systems in an autonomous framework. Steven received his B.S. in Electrical Engineering and Computer Science from the University of California, Berkeley.*

Richard Stottler *co-founded Stottler Henke in 1988 as a software company dedicated to providing practical solutions to difficult problems by skillfully drawing upon a large repertoire of artificial intelligence technologies. Under Dick's leadership, Stottler Henke has grown steadily and profitably into a 40-person research and software development company with distinctive expertise in intelligent tutoring systems, intelligent simulation, automated planning and scheduling, and intelligent knowledge management. Dick provides technical leadership in the design and development of intelligent tutoring systems, intelligent planning and scheduling systems, and automated design systems. He combines a strong applied research record in artificial intelligence with practical experience in rapid and efficient knowledge engineering. He has led the development of intelligent tutoring systems that encode the expertise of instructors to provide practice-based learning and automated evaluation of student performance in subject areas spanning navy tactics; army tactics, command, and control; sonar data analysis; astronaut training; helicopter cockpit operations; and battlefield emergency medicine. He also led the development of intelligent planning systems for NASA space shuttle missions and aircraft assembly and automated scheduling for the International Space Station. He also led the development of intelligent systems that encode and apply human expertise and experience to automate the design of manufacturing processes and aircraft systems to lower manufacturing costs, increase product quality, or achieve demanding performance criteria. Dick has written or presented more than two dozen papers and articles for publications such as the proceedings of the International Joint Conference on Artificial Intelligence (IJCAI). He received his BS in engineering from Cornell University and his MS in computer science (artificial intelligence) from Stanford University.*

Robert Richards *received a Ph.D. in Mechanical Engineering from Stanford University. Dr. Richards is managing and has managed multiple projects for both commercial and government clients, including various intelligent scheduling. Dr. Richards is the Principal Scientist and Manager of Stottler Henke's Pfizer project for scheduling pharmaceutical packaging plants, the end product is Aurora-ProPlan. Aurora-ProPlan is being rolled out to all of Pfizer's packaging plants throughout the world. Dr. Richards also lead the project for adapting Aurora to optimize the vehicle testing process by selecting the best vehicle configurations to minimize the vehicle count and overall schedule duration, the result is called Aurora-VT. Dr. Richards has also worked on and continues to work on various projects spanning a wide range of research and application area interests, including: training system development; applying automation and artificial intelligence techniques; and decision support tool development for life-critical situations. Dr. Richards has publications in all of these domains.*