Intelligent Scheduling at NASA: Case Study per Ground Operations at Kennedy Space Center

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

Abstract— **This paper provides a case study of improving NASA's Kennedy Space Center's (KSC) Ground Processing (GP) scheduling. These improvements have been applied to projects of various size ranging up to mega projects. The improvements developed and deployed at KSC automate a large amount of the planning, scheduling, and execution decision-making. The implemented can leverage project data from both Primavera P6 and Microsoft Project, allowing KSC to realize significant efficiency improvements in several different areas. Improvements include hazardous constraints; preferred versus required temporal constraints; generation of near-optimal schedules; as well as near-optimal rescheduling in real-time in response to changes and requests; and support of modelling details unique to KSC, so models true to life can be built and scheduled.**

These intelligent scheduling enhancements have been applied to several problems at KSC relating to Ground Processing (GP). The delivered application improves the scheduling of Space Launch System (SLS) Processing, including reduced turnaround time in response to changes and what-ifs, and more optimal schedules.

TABLE OF CONTENTS

1. INTRODUCTION

Historically, Kennedy Space Center (KSC) has had some of the most complex, difficult, diverse, 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, 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 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, 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 challenges of KSC's scheduling problems, 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 the Aurora intelligent scheduling software, 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.

2. 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 artificial intelligence representations. This project was the genesis of Stottler Henke's *intelligent* 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.

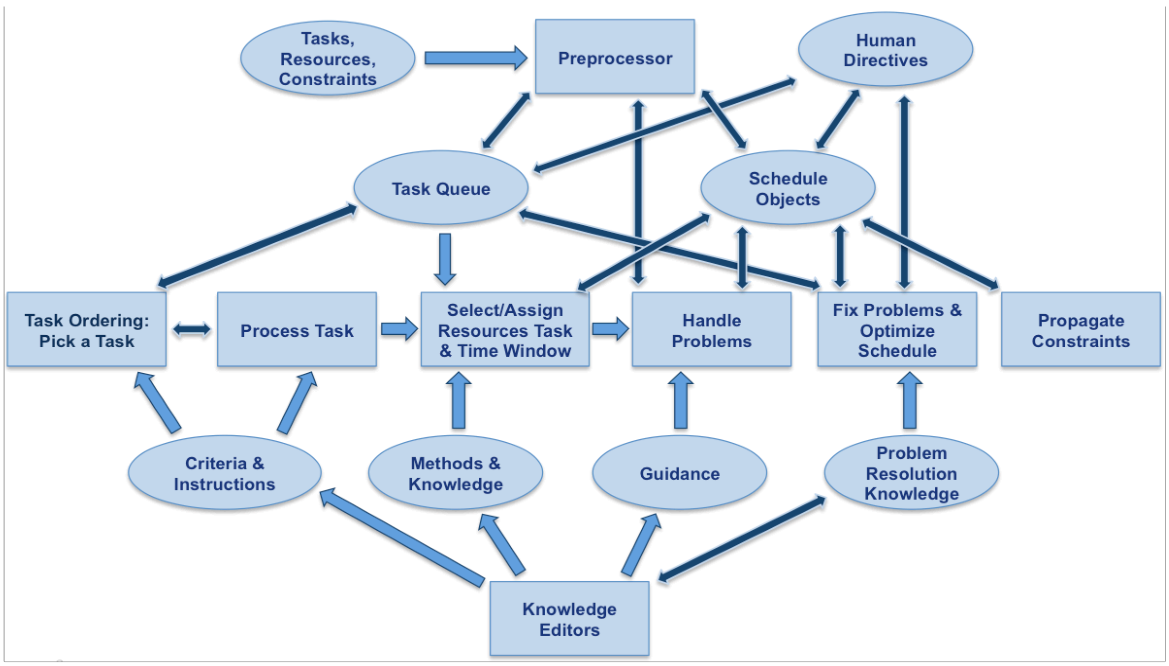

Figure 1. High-Level Aurora Architecture

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.

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. 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 2 shows an example of an explanation.

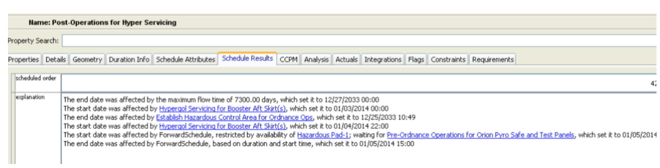

Figure 2. Automatically generated explanation

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

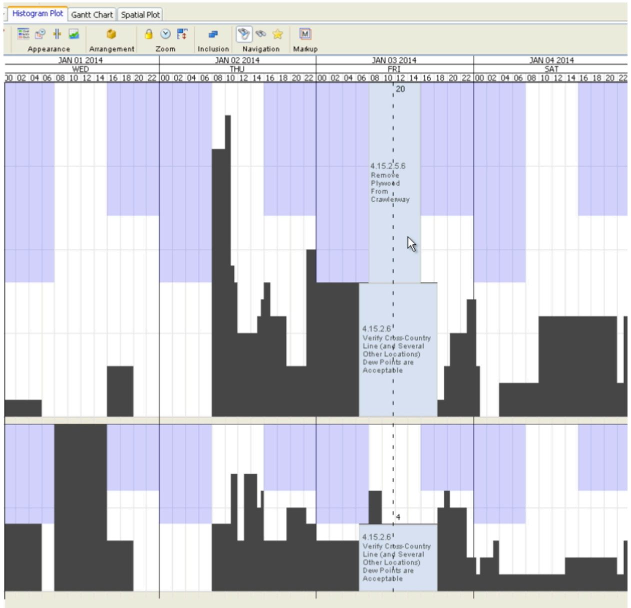

Figure 3. Manpower histogram, showing activities constituting manpower need for one time instance

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 4 shows a spilt view showing a Gantt chart.

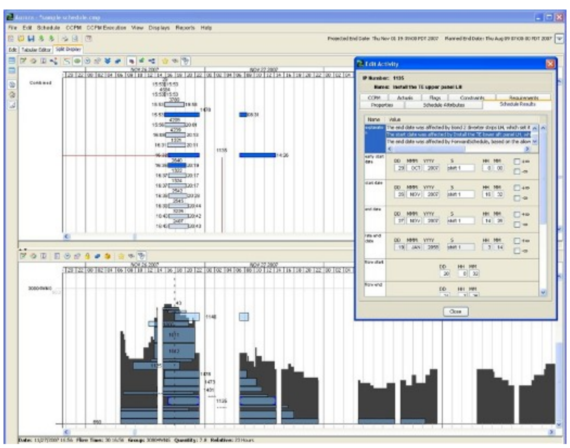

Figure 4. Split view showing Gantt chart same time slice as histogram, showing activities constituting resource need for one time instance

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

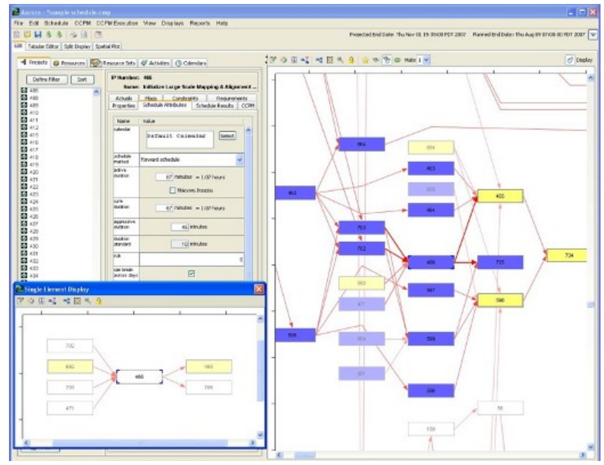

Figure 5. Network diagram showing single-element view option

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.

3. Automatic Scheduler Capabilities Needed for Ground Processing

Stottler Henke worked with KSC schedulers and other subject matter experts (SMEs) to understand 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
- 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 6 shows non-concurrent constraint for tasks A, B and Figure 7 shows concurrent constraints for task B, A, & C.

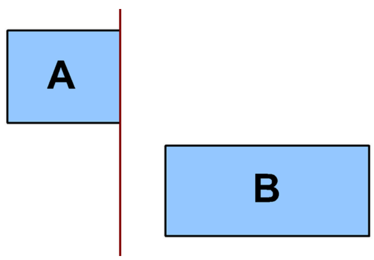

Figure 6. Non-concurrent tasks

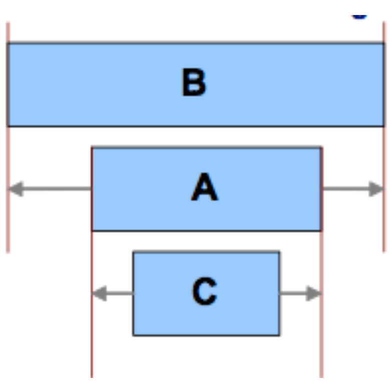

Figure 7. Concurrent tasks

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.

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 8 shows a Gantt chart color-coded for hazardous and powering requirements.

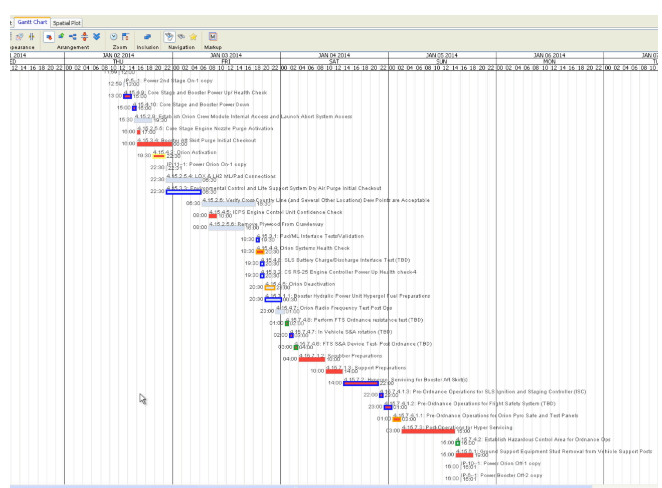

Figure 8. Gantt chart color-coded for hazardous and powering requirements

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 9 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.

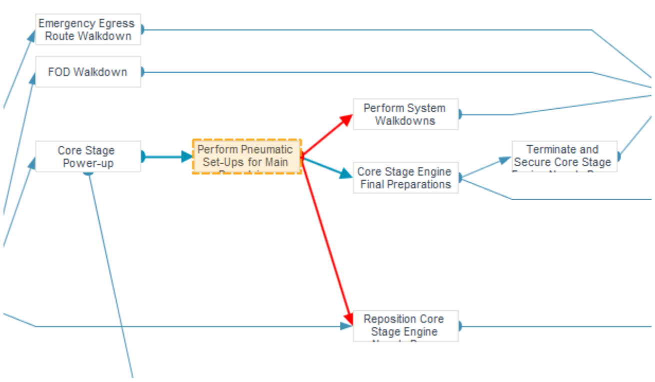

Figure 9. Hazardous constraints shown with red arrows

Figure 10 shows one of the options to set up hazardous constraints via a new Hazards tab.

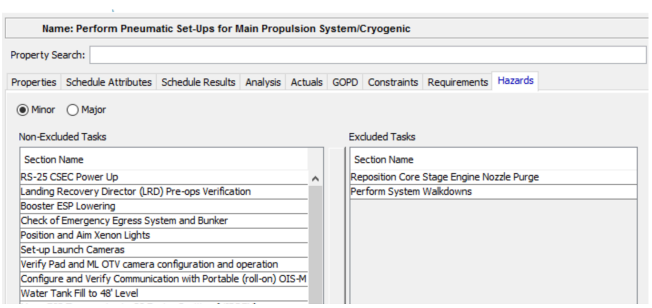

Figure 10. Hazards tab for setting up hazardous constraints

4. 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.

BIOGRAPHY

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.*