Proceedings of the 2014 Winter Simulation Conference
A. Tolk, S. D. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, eds.

USING DISCRETE EVENT SIMULATION TO MODEL INTEGRATED COMMODITIES CONSUMPTION FOR A LAUNCH CAMPAIGN OF THE SPACE LAUNCH SYSTEM

Daniel Leonard

DES Modeling and Analysis Center
Productivity Apex, Inc.
3505 Lake Lynda Drive, Suite 206
Orlando, FL 32817 USA

Jeremy Parsons

Operations Integration Division
Ground Systems Development and Operations Program
NASA Kennedy Space Center
Kennedy Space Center, FL 32899 USA

Grant Cates

DES Modeling and Analysis Center
The Aerospace Corporation
P.O. Box 21205
Kennedy Space Center, FL 32815-0205 USA

ABSTRACT

In May 2013, NASA's GSDO Program requested a study to develop a discrete event simulation (DES) model that analyzes the launch campaign process of the Space Launch System (SLS) from an integrated commodities perspective. The scope of the study includes launch countdown and scrub turnaround and focuses on four core launch commodities: hydrogen, oxygen, nitrogen, and helium. Previously, the commodities were only analyzed individually and deterministically for their launch support capability, but this study was the first to integrate them to examine the impact of their interactions on a launch campaign as well as the effects of process variability on commodity availability. The study produced a validated DES model with Rockwell Arena that showed that Kennedy Space Center's ground systems were capable of supporting a 48-hour scrub turnaround for the SLS. The model will be maintained and updated to provide commodity consumption analysis of future ground system and SLS configurations.

1 INTRODUCTION

1.1 Purpose

The National Aeronautics and Space Administration (NASA) is currently experiencing a time of transition in its human exploration program as it moves from a Space Shuttle centered model to one based on the Space Launch System and Orion Programs. To support these exploration endeavors new ground systems must be developed that will enable low cost and efficient processing of these vehicles while being flexible enough to allow multiple-users that can offset capacity costs. The Ground Systems Development and Operations Program (GSDO) at Kennedy Space Center is working to actively design, develop, and implement transitional ground systems that will reduce long term operational costs. To meet these aggressive development and operations goals GSDO is employing Discrete Event Simulation (DES) to quantitatively forecast future operations and influence design early in the lifecycle. DES is being used in a number of areas including: to forecast processing durations, delay risks, resource demands of personnel and ground support equipment, launch availability modeling, and most recently to understand the commodity demands that will be placed on the infrastructure for a launch campaign. This is allowing the management team to make architectural decisions based on quantifiable and empirical data grounded in advanced simulations.

The Space Launch System, or SLS, will be a human-rated heavy-lift launch vehicle. Initial test flights are planned for 2017 with the first crewed flight planned for 2021. The initial version of the SLS, the Block 1 vehicle, will be composed of an integrated Core Stage, heritage RS-25 engines and Boosters, and an Interim Cryogenic Propulsion Stage (ICPS). The SLS will carry the Orion spacecraft which will be capable of long duration missions to deep space destinations. To achieve the missions that are expected to be carried out by the SLS/Orion, GSDO needs to have the capability to perform multiple launch attempts in a limited window. In the long term destinations when multiple launches will be rendezvousing in orbit it will be absolutely critical that GSDO be able to maximize the probability of launching and one key factor for that is being able to turnaround from one attempt to another in a quick fashion.

The Space Shuttle could be fueled for launch as often as three days in a row. However, it became clear that the SLS would require significantly more liquid hydrogen such that a 24-hour scrub turnaround would not be feasible with the existing infrastructure at KSC. Analysis was required to determine how long the scrub turnaround timeline would require given the increased hydrogen demands. There was also concern that other commodities might also limit the ability to perform multiple launch attempts over a multi-day period.

The purpose of the Launch Campaign Integrated Commodities Analysis study (LCIC) was to use DES to analyze the launch campaign process for the SLS from an integrated commodities perspective. A *launch campaign*, as used in this study, means the collective series of activities that occur during launch preparation and countdown, a scrub turnaround, a second launch attempt, and finally another scrub. The four core launch commodities modeled for the study were liquid hydrogen (LH2), liquid oxygen (LO2), gaseous nitrogen (GN2), and gaseous helium (GHe). NASA wanted a tool to analyze the launch support capability of the commodities, and specifically their ability to support a 48-hour scrub turnaround. A *scrub turnaround* is the set of processes that are required to safe a launch vehicle after a scrub and then prepare and execute another attempt. The objective of the 48-hour scrub turnaround is to reach T-0 of the second attempt within 48 hours of the first scrub.

The launch campaign timeline is based on the processes for Exploration Mission 1 (EM-1) and Exploration Mission 2 (EM-2), the first planned launches of the SLS in 2017 and 2021 respectively. GSDO has a requirement to be capable of completing a 48-hour scrub turnaround in order to achieve two launch attempts during the launch window. GSDO wanted the study to produce a DES model that allows users to analyze the interactions of current and future ground systems configurations, launch campaign process variability, and SLS commodity requirements to make one integrated story of commodity use. The study used Rockwell Arena 14 for developing the model and performing analysis.

1.2 Literature Review and Method Selection

The LCIC analysis builds upon past DES related analysis efforts at KSC. In 1999, KSC entered into a Space Act Agreement with the University of Central Florida to develop a DES model of the entire Space Shuttle operational flow. The goal of this effort was threefold: first to demonstrate the utility of DES based analysis; second to develop a cadre of DES expertise at KSC; and finally to provide a useful tool for helping NASA increase the Shuttle flight rate. (Cates et al. 2002)

The Space Shuttle model that was developed was subsequently leveraged in 2002 to develop the Manifest Assessment Simulation Tool (MAST). MAST was first used in January 2003 to provide NASA with an assessment of the likelihood of achieving U.S. Core Complete of the International Space Station (ISS) by February 19, 2004. (Cates and Mollaghasemi 2005a) MAST was used by the NASA Chief Engineer in 2004 and then again in 2005 by the Shuttle/Station Configuration Options Team (S/SCOT) to explore the questions of: (1) when will assembly of the ISS be completed; and (2) how many Space Shuttle missions can be flown by the end of 2010? The NASA administrator subsequently reduced the number of planned flights remaining in the Space Shuttle manifest based upon the analysis results. (Cates and Mollaghasemi 2005b, Griffin 2005)

DES analysis was used to support the Constellation program. One notable model being the Constellation-Requirements Assessment by Simulation Technique (C-RAST). C-RAST was intended to provide a demonstration of how DES could be used to help the Constellation program analyze program level requirements. C-RAST was used to analyze the probability of launching both the Ares V and the Ares I in a timely fashion. (Cates, Cirillo, and Stromgren 2006, Stromgren 2009) Although the Constellation program was cancelled in 2010, elements of that program, including the Orion and the heavy lift SLS launch vehicle that is essentially a renamed Ares V, are continuing to be developed. The C-RAST model was modified to provide launch probability assessments for the SLS. This "Integrated Launch Probability Model" is currently being managed as a joint effort between GSDO and the SLS and Orion program offices. (Watson 2014) The scrub turnaround capability to support SLS launches directly influences the cumulative probability of launch. The results of the LCIC model will provide this critical information for use in the Integrated Launch Probability Model.

The launch campaign process involves a significant amount of continuous commodity flows. It is difficult for a DES software to accurately model continuous operations, so to solve that problem the first Discrete Rate Simulation (DRS) software tool was introduced by Simulation Dynamics, Inc. in 1997. (Siprelle and Phelps 1997) Discrete Rate Simulation "is a method for simulating continuous, rate-based flow systems and hybrid (combined continuous and discrete event) systems." (Damiron and Nastasi 2008) DRS is much more precise when modeling continuous rate systems because the software calculates the exact time of the event that changes the flow rates instead of reacting to a different state during the next interval update. DRS is also better than assuming that a DES entity represents a small amount of mass that flows through the system, because the large number of entities would greatly reduce run time efficiency and would not account for fractional units. (Siprelle and Phelps 1997)

Arena uses DRS with its flow process modules that are designed to run a continuous flow and stop at discrete points in time to achieve extreme precision. The start and stop times are determined by the main SLS processing timeline, which means the durations of commodity usage are variable like the SLS task times. This variability can cause significant swings in processing time, which could mean increased commodity consumption, so studying the impact of variability on commodity availability is one of the main reasons for this project. The model helps the user support NASA personnel with usage analysis and can show changes to consumption if they plan system or processing changes.

2 METHODOLOGY

2.1 Process Mapping

The first task in the study was to understand and map the launch campaign process. Process mapping started with studying a tool that GSDO developed called the Ground Operations Planning Database (GOPD), which is used for SLS timeline planning and contains information necessary for this study like three-point estimates for task durations and predecessor/successor relationships. Three-point estimates are the minimum, expected, and 95th percentile estimates for task durations and are used as inputs for the time distributions of stochastic tasks. This model uses lognormal distributions for time inputs in Arena because previous studies done by this team have shown lognormal to be the best distribution for estimating stochastic process times.

The critical path was defined using the precedence relationships and task durations, then an initial process map was developed based on the critical path as well as any other tasks that consumed launch commodities. The first draft was corrected and improved during meetings with NASA subject matter experts (SME) who had experience with Space Shuttle processing or commodity use. The SLS launch campaign process can be broken down into four main operations groups: pre-tanking, tanking, post-cryo loading, and scrub turnaround. (Figure 1) Pre-tanking operations start with a task called Call to Stations, and includes other processes like powering up the SLS, configuring systems, performing pad walkdowns, and turning on gas purges. Tanking Operations is the sequence of steps to fill the SLS with LH2 and LO2 (sometimes referred to as "cryo loading" or simply "tanking") as well as flowing various GN2 and GHe purges. Post-Cryo Loading involves crew ingress and securing, hold time, and terminal count operations that end with the scrub, which is assumed to be 9 seconds before T-0 in the model. Scrub Turnaround operations include crew egress, tank drain ("detanking"), refilling the LH2 storage tank on the pad, and running SLS tank inerting purges. When the first scrub operations are completed the model goes back to Call to Stations and does everything one more time.

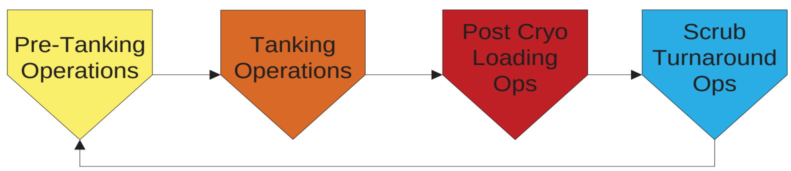

Figure 1: High-Level Process Map

2.2 Commodity Data Collection

The model measures commodity use by total mass consumed during processing, so the data that needed collected were flow rates (gal/sec for liquids and lbm/sec or scfm—standard cubic feet per minute—for gases), start times, and durations. Sometimes only a quantity estimate based on Shuttle data could be obtained so dividing quantity by the duration gave the flow rate for the model. These data were collected from SMEs in various NASA organizations including NASA Engineering, Center Operations, GSDO, SLS, and Orion. The flow rates were organized in timelines based on the deterministic start times and durations for Center Operations to verify that KSC's ground systems could support the totals. When flow rates, start times, and durations were confirmed they would be added to the model, which was developed concurrently with data collection due to information delays and discovering missing flow tasks while validating the model. The final model had 140 unique flow tasks between all four commodities.

2.3 Developing the Model

The LCIC model was developed using Rockwell's Arena simulation software. A typical simulation model uses entities that represent people or widgets, but the processes the LCIC modeled were continuous flow operations, which needed a different set of simulation modules than discrete Process modules. Therefore, the model relied heavily on Arena's Flow Process modules that simulate continuous flows with a single entity representing the launch campaign that activates flow tasks when it arrives at the modules. The Flow Process modules used in this model were Tank, Flow, Regulate, Seize Regulator, Release Regulator, and Signal. Figure 2 is an example of the flow modules used to simulate boiloff in the LH2 source tank.

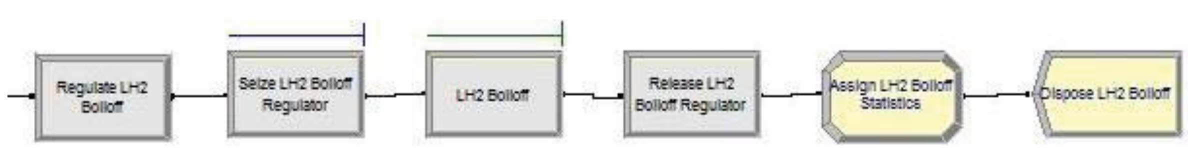

Figure 2: Flow Modules Example

Sources of commodity flows (LH2 and LO2 spheres on the launch pad, LH2 refill tankers, the LN2 tank, and GHe batteries) and destinations (LH2 and LO2 tanks in the SLS) were represented with the Tank modules. (Figure 3) Each tank is assigned a capacity and initial level (either full or empty) as well as many regulators, which are like pipes for input or output so every flow requirement has its own dedicated pipe.

For LH2 and LO2 transfers from storage to the SLS there are matching regulators for each simulation tank, which act as endpoints for the transfer pipe.

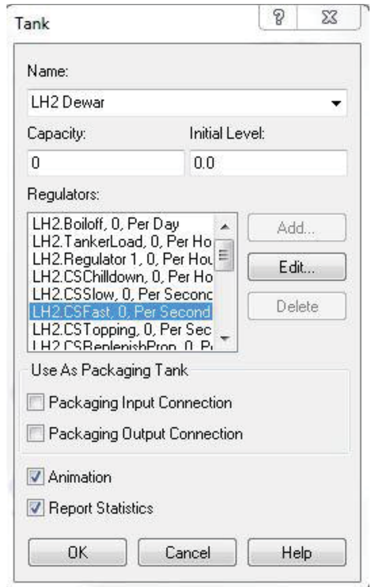

Figure 3: Tank Module Properties Example

Seize Regulator modules seize the tank regulators like a resource, so each regulator can only be used for one flow operation at a time. The Flow modules are similar to a typical Process module, but can either add to, remove from, or transfer between tanks. (Figure 4) The user selects the tank(s) and the regulator(s) to or from which to flow and then one of three conditions that terminate flow: running for a specific duration, flowing a specific quantity, or receiving an external signal.

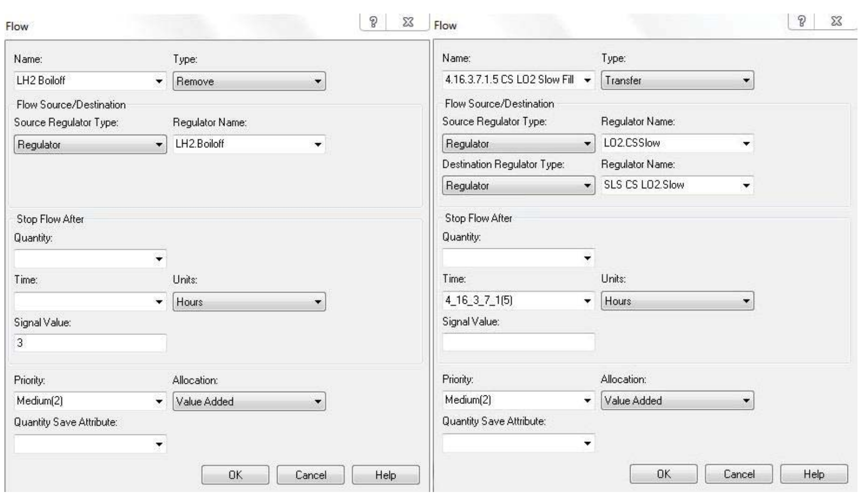

Figure 4: Flow Module Properties Example

The Regulate module is used like a valve to change the flow rate of any tank regulator. The model primarily uses Regulate modules for flow rate input before a flow starts, but they can also change the flow rates while a Flow module is running. Figure 5 is an example of a Regulate module properties menu, showing the options to change the LH2.Boiloff regulator's flow rate to the LH2LossFlow(15) expression, which is defined in the Excel input file. When a Flow module terminates the Release Regulator module releases the tank regulator(s) to be used by another flow module or wait until the next launch attempt. The last module utilized for this study was the Signal module, which could create an entity, assign a variable, or regulate a regulator if a certain condition was reached in a tank, such as being empty or full. Together, the Flow Process modules were able to capture every type of action required to simulate the flow tasks for a launch campaign.

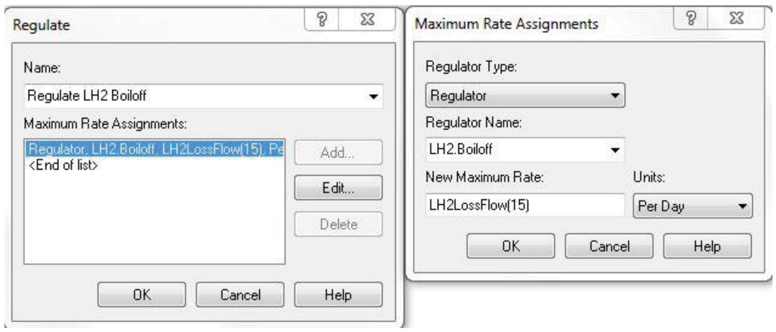

Figure 5: Regulate Module Properties Example

The first iteration of the model captured the tasks listed in the GOPD, which established the model timeline. The GOPD had minimal GN2 and GHe information, but complete flow data was available for fill and drain tasks for LH2 and LO2. Non-commodity tasks were modeled using the Basic Process modules and the Flow Process modules were added for fill and drain. As the data collection process added more GN2 and GHe tasks, the model was expanded to accommodate them with new regulators and flow modules for each one. Most GN2 and GHe tasks had start and end times that depended on a GOPD task, such as start of cryo fill, start of terminal count, or end of drain. The model utilized Hold modules that waited for a signal before releasing an entity to start a flow, and the flow did not stop until it received a second signal that was triggered by a later event. Using this signal system linked all of the flow processes to the established GOPD timeline as well as its variability.

GOPD process time three-point estimates and commodity flow rates are listed on an Excel input file and are read from the model via named ranges. The ranges then become variable expressions for use in the process modules. This input method is used because it is easier to change and manipulate inputs on the spreadsheet than digging through a model, especially if the input value is used in several different modules. A reverse method is used for recording outputs. Output data are recorded in variables, which are then written to named ranges in an Excel output file. Writing the output to Excel allows the user to easily view and analyze the data.

The last step in model development was adding animations for users and stakeholders to see the current status of the tanks and the SLS. This study utilized Arena's Level visualizations to depict both tanks and pipes. The Level tool has circles that can fill vertically to represent storage tanks, SLS tanks, and tanker trucks and it has flow level visualizations to represent pipes with arrows for flow direction. The animation and result charts are color-coded with red for LH2, green for LO2, blue for GN2, and orange for GHe. Line graphs were also included alongside the animation to see the specific levels, especially if the tank goes empty. Figure 6 is a screenshot of the animation during the replenishment phase of the first launch attempt.

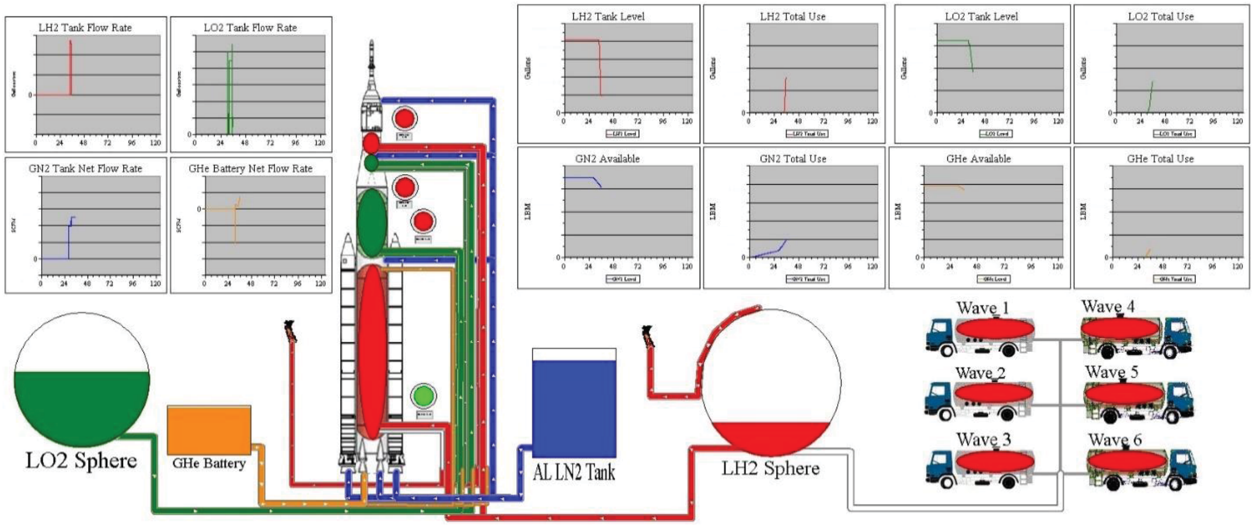

Figure 6: LCIC Model Animation

2.4 Problem Solving

There were several quirks in the simulation environment that were addressed during model development. The first problem was inputting the flow rates. Expressions could not be assigned to regulators in the Tank modules and using the numbers would make changing them harder, but the Regulate modules did allow expression input so a Regulate module was included before every Flow module to set the correct flow rate before every flow task begins. Unfortunately the inputs for capacity and initial values of tanks could only be manually added in the tank modules because expressions were not valid inputs. The small number of tanks in the model and infrequent capacity changes meant this was not a major inconvenience.

One objective of this study was analyzing commodity usage and finding out how much the SLS needs, but if a tank in the model ran out of commodity units during a flow task then the output could not show how much was really needed. If a flow module was coded to end after a certain quantity of commodity flowed through, then the entity would be stuck and never leave because there was nothing left to flow. This was fixed by simply adding capacity in the model so that flows were never interrupted, however this added a new problem with the output variable for the current tank levels. To offset the additional tank capacity the expressions for level outputs were subsequently adjusted by subtracting the extra capacity so that the output graph would show when the tank goes empty and then how much more was needed by going negative. In addition to knowing when a commodity was “broken,” which means the system cannot handle the demand, counting the number of times a tank runs negative over 1000 replications provided the probability of breaking.

Because many tasks happen simultaneously during a launch campaign, the model duplicates the mission entity many times in a replication. A result of this duplication and the significant use of signals was that if several signals are triggered by one entity at a single instance in time then different modules that are waiting for one of the signals might miss it. The Arena events calendar sometimes sequenced things that happen simultaneously in a way that would stall a replication. To get around this problem very short delays of 0.01 seconds were added between back-to-back signals so entities that respond to the first signal could move on to their next modules before the next signal was sent. Even though this might have added some time to the overall processing time a few hundredths of a second are negligible when the nominal total time is 125 hours.

2.5 LH2 Failure Rate vs. Missed Launch Window Problem

During the development of this model, the team came across a difficult problem that NASA managers will have to address as they approach the first SLS launch date. The problem is that if the processing time takes too long because of the variability, then they could run out of LH2 during the replenishment phase; however, if they start the processing later in order to conserve LH2, then they might not have enough time to finish all of the required processing before the launch window closes. This might just be a problem that is unique to this model, because NASA plans processing timelines by counting backwards from T-0, but Arena only counts forward from the start of the model at time 0.00. In addition, NASA adds countdown holds throughout a final timeline to act as a buffer for variability, but those holds are not defined yet and therefore are not included in the main processing timeline. Nevertheless, the only way to give any reasonable stochastic output analysis for this model at this time was to find the optimal buffer duration that minimized running out of LH2 and missing the launch window.

The logic for determining whether the attempt missed the launch window is made up of three decision modules: first or second attempt, because they are not the same nominal length of time; missed or did not miss the launch window; and deterministic or stochastic. (Figure 7) The assign modules after the first decision calculate the "extended hold margin", which is the difference between the duration of the launch window and the elapsed time in the model. The only official hold in the GOPD timeline is a hold of 2.5 hours after crew ingress and before terminal count (T-5 minutes), so if the elapsed time surpasses the extra 2.5 hours then that is counted as a "missed launch window" (MLW) in the next decision module and is immediately scrubbed without holding. If the elapsed time has not exceeded the launch window then the last decision logic is to assign the hold time, which is 2.5 plus the hold margin. If the margin is between -2.5 and 0 then the hold will be shorter because the processing has eaten up some of that hold time; if the margin is positive then the actual hold time will be longer than 2.5 hours because the processing finished before the launch window even opened, but we are trying to model a worst-case scenario as well which means using the whole 2.5-hour hold time.

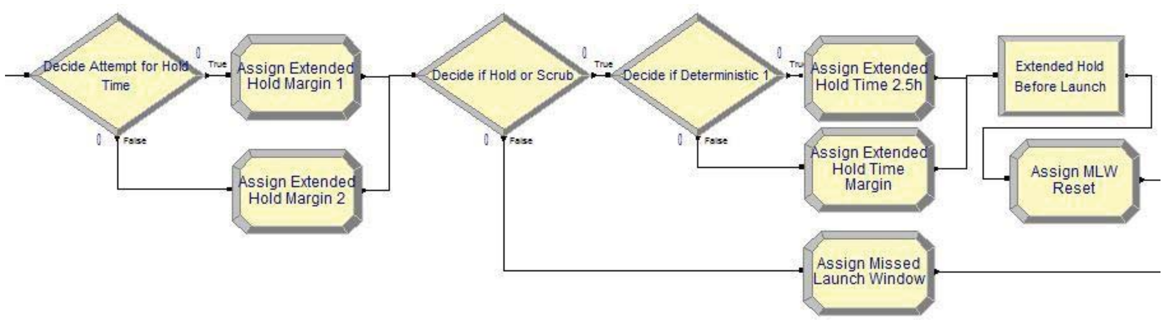

Figure 7: Missed Launch Window and Hold Time Logic

The "sweet spot" between LH2 failures and MLW is the duration of the assumed additional hold time, which is added to the nominal (deterministic) duration to simulate starting the launch campaign processing earlier than without any additional time. Running the simulation 11 times with varying additional time produced two results. Table 1 shows the time margins tested and the resulting LH2 failure and MLW rates. Initially the goal of the analysis was to find the time where the rates were equal, but after examining the results for the number of attempts that no LH2 failures or MLW occurred, we found that the highest percentage of clean attempts was a different time than the equilibrium time. Therefore, 4.25 hours of additional launch window time was added to the nominal time for the stochastic runs that produced the results later in this paper.

Table 1: LH2 Failure Rates vs. Missed Launch Window Rates

Add'l Time	LH2	LH2 % Fail	MLW	LW % Fail	% Fail	% Success
2	36	1.80%	1393	69.65%	71.05%	28.95%
2.5	59	2.95%	1251	62.55%	65.10%	34.90%
3	94	4.70%	1129	56.45%	60.75%	39.25%
3.5	155	7.75%	1017	50.85%	58.20%	41.80%
4	243	12.15%	890	44.50%	56.25%	43.75%
4.25	288	14.40%	839	41.95%	**55.95%**	**44.05%**
4.5	359	17.95%	780	39.00%	56.55%	43.45%
5	506	25.30%	664	33.20%	58.10%	41.90%
5.25	569	28.45%	608	30.40%	58.45%	41.55%
5.33	**588**	**29.40%**	**591**	**29.55%**	58.55%	41.45%
5.5	644	32.20%	553	27.65%	59.45%	40.55%

3 RESULTS

3.1 Turnaround Time

Output analysis was performed on the model after it was validated by KSC's Engineering Review Board (ERB). This study ran 1000 replications of the launch campaign, for a total of 2000 attempts. When this study began the operations plan for refilling the LH2 tank involved small groups of tankers making three trips between KSC and New Orleans, where the LH2 is produced. The result was a scrub turnaround time between 186 and 206 hours. GSDO eventually baselined a new plan that has all required tankers parked on site in case of a scrub. This change reduced the turnaround time from over seven days down to 48 hours. GSDO wanted to confirm that the commodities could support the 48-hour scrub turnaround requirement. The model output showed that 73% of the time the second attempt occurred within 48 hours (Figure 8), but that can be increased by reconfiguring the planned processing timeline to move tasks earlier or have more happen in parallel.

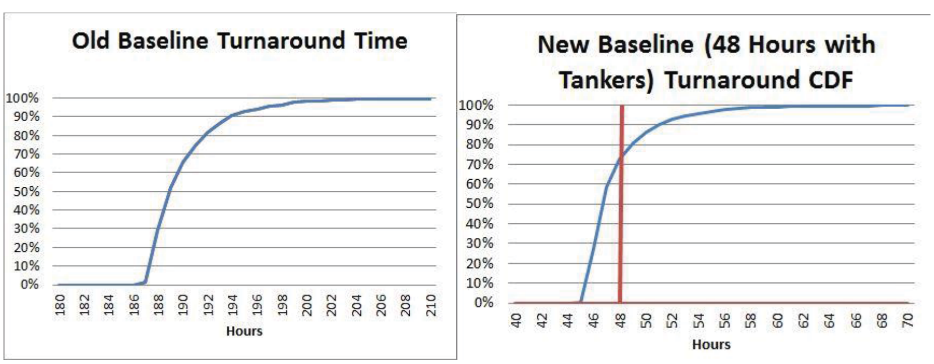

Figure 8: Scrub Turnaround Cumulative Distribution Function

3.2 Commodity Usage

The team developed several other output charts depicting three types of commodity use during two attempts: total use, instantaneous flow rate, and tank level (availability). The charts in this paper are deterministic data that include a timeline of events with lines showing the quantity used, flow rate, or tank level, but the scales on the vertical axes were removed for data security. Figure 9 (left) shows the total use of GHe, which steadily increases during the launch campaign due to many gas purges being on continuously. Figure 9 (right) also shows the instantaneous GN2 flow rates, which is never zero due to GN2 being required by other locations at KSC, and is highest during tanking and detanking.

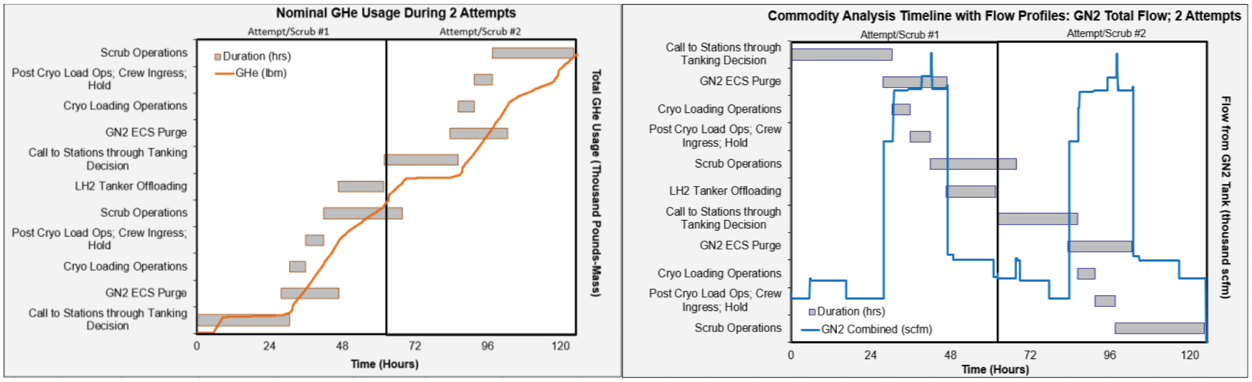

Figure 9: Deterministic GHe Usage and GN2 Cumulative Flow Rates During 2 Attempts

Figure 10 (left) shows the LH2 tank levels. The precipitous drops in the level is during tanking; the slow decline is replenishment; the fast recovery is drain back from detanking; the last slower increase in tank level is due to tanker refill, which only happens after the first scrub. It is clear that the tank level gets very close to zero during replenishment, which is why LH2 breaks much more often than the other commodities. Figure 10 (right) also shows the LO2 tank levels, which could break during replenishment on the second attempt in extreme cases, but only went empty once during the 1000 campaigns.

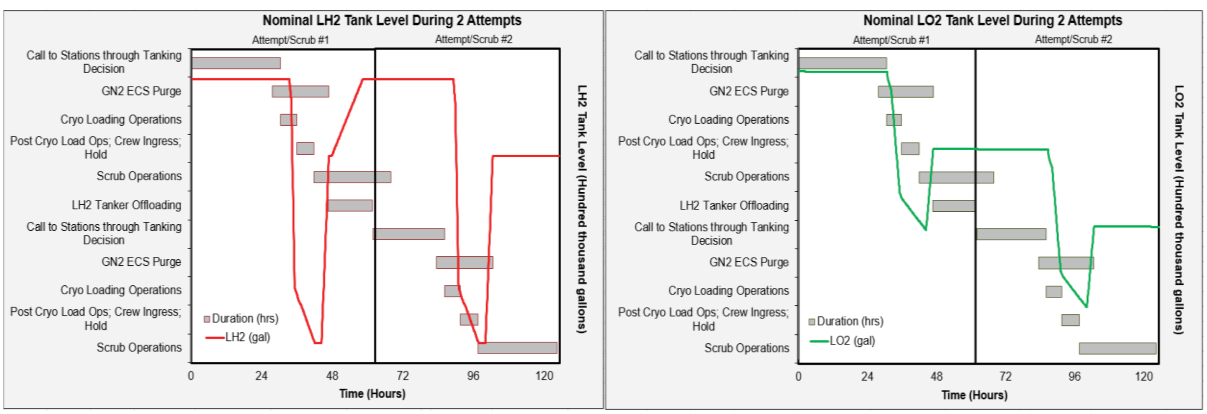

Figure 10: Deterministic LH2 and LO2 Tank Levels during 2 Attempts

3.3 Failure Rate

Figure 10 is especially notable for how low the LH2 level goes by the end of tanking, which leads into Figure 11, showing the frequency that commodities break during attempts. 83.75% of 2000 attempts did not have any commodity failure. Of the 16.25% of attempts that had failures, almost all were due to LH2 and GHe, which made up 80% and 18.06%, respectively. The other two only failed 7 times in total, where GN2 broke six times, and LO2 broke once. The total of 360 failures is slightly larger than 16.25% of 2000 because some attempts had more than one commodity failure. These and other similar charts were used to present to the GSDO Program Review Board (PRB) to conclude the study.

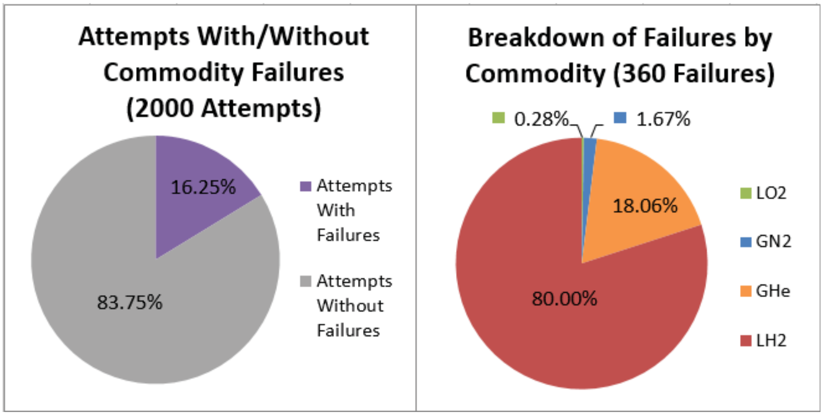

Figure 11: Stochastic Failure Rates of Commodities

4 RECOMMENDATIONS

The study produced a valid model of launch campaign commodities consumption, but it was limited in scope to a narrow time window. Further work on this model for GSDO analysis support should include expanding the time frame to five days, instead of two, before T-0 in order to capture all commodity usage tasks that occur between SLS arrival at the launch pad and T-0. Flow rates must be maintained on the input file as they are updated, but a data submission process or schedule will need to be established between analysts and SMEs.

5 CONCLUSION

It should be acknowledged that GSDO is attempting to gain the most flexibility out of a heritage shuttle system that would be extremely expensive to replace. Therefore the Program can make better decisions through understanding its operating margins and the probabilities of being able to make mission goals without increasing up front investments. The LCIC model can be used to support decision-makers as they plan upgrades to KSC facilities. Knowing how often LH2 or GHe break and how much extra is needed to avoid failures is key to the continuing improvements GSDO is making at KSC.

In addition to decision support, the LCIC model provides significant documentation support as well. The model has given GSDO a new way of quickly assessing design changes to Interface Control Documents (ICD) between the Programs, which contain current design plans for the SLS and are important reference documents for the Programs. Previously, assessments would take place individually in a non-integrated fashion, but the creation of this model allows all aspects to be investigated at once. The LCIC study itself has integrated previously disparate sets of commodity use data that are essential to future SLS and GSDO plans and designs.

ACKNOWLEDGEMENTS

Thank you to Rogelio Franco for your guidance in completing this project, to Etosha Ottey for your support during data collection and continuing the work of this project in the future, and to Danielle Walsh for your support in understanding the process and data collection. Thank you to the following NASA subject matter experts who provided commodity data or process information for this project: Angela Krenn, Christian O’Connor, Rogelio Franco, Joe Gaddone, Janet Jingle, Don Jennings, Craig Rosczypala, Ed Ryan, Mark Stewart, Miles Ashley, John Sterritt, and Chuck Davis.

REFERENCES

Cates, G. R., M. J. Steele, M. Mollaghasemi, and G. Rabadi. 2002. "Modeling the Space Shuttle." In *Proceedings of the 2002 Winter Simulation Conference*, edited by E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, 754-762. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Cates, G. R. and M. Mollaghasemi. 2005a. "A Discrete Event Simulation Model For Assembling The International Space Station." In *Proceedings of the 2005 Winter Simulation Conference*, edited by M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, 1260-1264. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Cates, G. R. and M. Mollaghasemi. 2005b. "Supporting The Vision For Space With Discrete Event Simulation." In *Proceedings of the 2005 Winter Simulation Conference*, edited by M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, 1306-1310. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Cates, G. R., W. M. Cirillo, and C. Stromgren. 2006. "Low Earth Orbit Rendezvous Strategy for Lunar Missions." In *Proceedings of the 2006 Winter Simulation Conference*, edited by L. F. Perrone, F. P Wieland, J. Liu, and B. G. Lawson, 1248-1252. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Damiron, C. and A. Nastasi. 2008. "Discrete Rate Simulation Using Linear Programming." In *Proceedings of the 2008 Winter Simulation Conference*, edited by S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, J. W. Fowler, 740-749. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Griffin, Michael D., Administrator, National Aeronautics and Space Administration before the Committee on Science, House of Representatives, Nov. 3, 2005

Siprelle, A. J. and R. A. Phelps. 1997. "Simulation of Bulk Flow and High Speed Operations." In *Proceedings of the 1997 Winter Simulation Conference*, edited by S. Andradóttir, K. J. Healy, D. H. Withers, and B. L. Nelson, 706-710. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Stromgren, C., G. Cates, and W. Cirillo. "Launch Order, Launch Separation, and Loiter in the Constellation 1½-Launch Solution." 2009 IEEE Aerospace Conference, Big Sky, MT, March 7-14, 2009.

Watson, M., Staton, E., Cates, G., Finn, R., Altino, K.M., Burns, K.L. "Use of DES Modeling for Determining Launch Availability for SLS." *AIAA SpaceOps 2014* / 13th International Conference on Space Operations, Pasadena, CA. 5-9 May, 2014.

AUTHOR BIOGRAPHIES

DANIEL LEONARD is an industrial engineer at Productivity Apex, Inc. in Orlando, Florida, and worked as a contractor at Kennedy Space Center as a simulation analyst for the GSDO Program. He received his B.S. in Industrial Engineering from the University of Central Florida in 2013. He is a member of IIE. His email address is dan@productivityapex.com.

JEREMY PARSONS is the branch chief of the Operations Integration division of the GSDO Program at NASA Kennedy Space Center. He received his M.S. in Industrial Engineering from the University of Miami and his B.S. from the University of Central Florida. His email address is jeremy.w.parsons@nasa.gov.

GRANT CATES is a modeling and simulation analyst at The Aerospace Corporation and the GSDO Program at Kennedy Space Center. He received his Ph.D. from the University of Central Florida in 2004. He is a member of IIE, INFORMS, and AIAA. His email address is grant.r.cates@nasa.gov.