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

USING DISCRETE EVENT SIMULATION TO MODEL FLUID COMMODITY USE BY 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 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 model utilized the flow process modules in Rockwell Arena to simulate the commodity flows and calculate total use. The study produced a validated DES model that showed that Kennedy Space Center's ground systems were capable of supporting a 48-hour scrub turnaround for the SLS.

1 INTRODUCTION

1.1 Purpose

The National Aeronautics and Space Administration (NASA) is currently transitioning its human exploration program 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, Shuttle 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. Missions to Mars will require multiple launches to assemble an integrated Mars Transfer Vehicle in earth orbit. (Drake 2009) It will be absolutely critical that GSDO be able to maximize the probability of launching each mission as close to schedule as possible. One key factor for that will be the ability to turnaround from one attempt to another in a quick fashion.

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. The four core launch commodities modeled for the study were liquid hydrogen (LH2), liquid oxygen (LO2), gaseous nitrogen (GN2), and gaseous helium (GHe). 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. A *scrub* is when a launch countdown is aborted before launching, and a *scrub turnaround* is the set of processes that are required to make a launch vehicle and pad safe for human work after a scrub and then prepare and execute another attempt. NASA wanted a tool to analyze the launch support capability of the commodities, and specifically their ability to support a 48-hour scrub turnaround. 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 third to provide a useful tool for helping NASA increase the Shuttle flight rate. (Cates et al. 2002)

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 than DES 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.

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. (Trocine, et. al 2010)

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 repeats the launch campaign process.

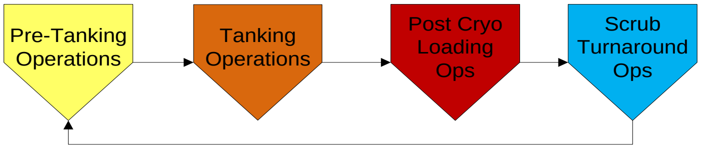


Figure 1: High-Level Process Map.

2.2 Commodity Data Collection

The model measures commodity use by total mass consumed during processing, so the data that needed to be 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. The first iteration of the model captured the tasks listed in the GOPD, which established the model timeline from start to scrub, and included tanking processes but nothing for GN2 and GHe. Non-commodity processing tasks were modeled using the Basic Process modules, but commodity use tasks had to be modeled differently because they were continuous flow operations. Arena's Flow Process modules simulate continuous flows by adding, removing, or transferring commodity units to and from simulated tanks. 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. Boiloff is when the liquid hydrogen in the LH2 tank warms up and boils to become gaseous hydrogen, which escapes through a pipe and burns in a flame stack. This happens continuously regardless of other operations.

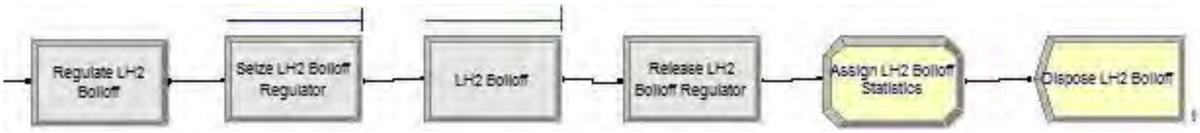
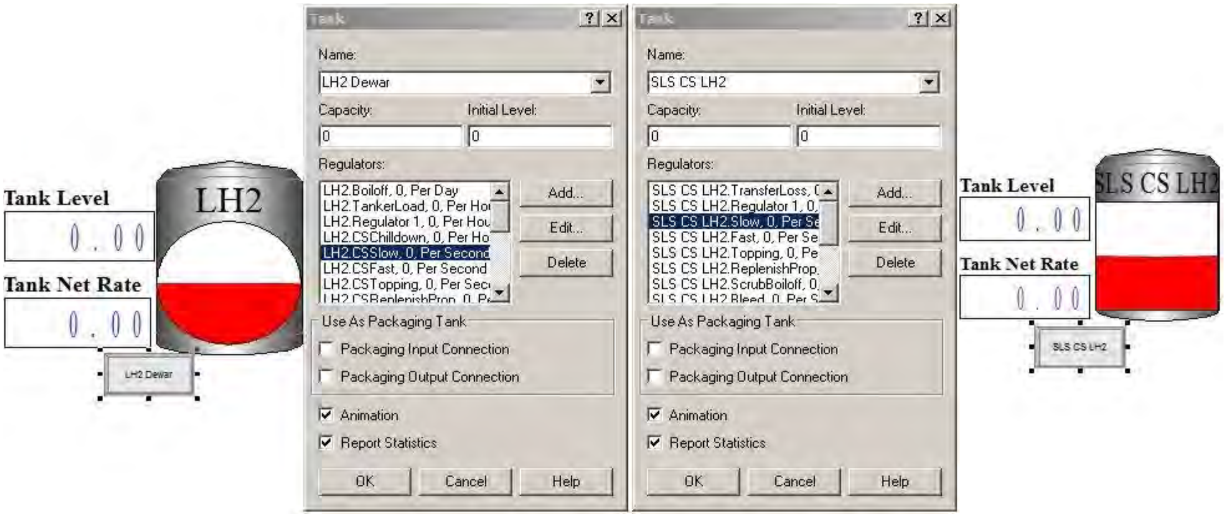


Figure 2: Flow Modules Example.

Commodity sources (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. 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. The example in Figure 3 shows the LH2 Dewar and SLS Core Stage (CS) tanks with the slow fill process highlighted. Slow fill is the first fill stage of tanking when LH2 is slowly pumped from the Dewar to the CS. For all LH2 and LO2 transfers there are matching regulators for each tank because every "pipe" needs a beginning and end. Values in Figure 3 are set to zero for data security.

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. The boiloff process only removes from the LH2 Dewar so only one source is required to be listed, however the slow fill process is a transfer so both the source and destination regulators must be specified in the process module. The user also specifies one of three conditions that terminate flow: running for a specific duration, flowing a specific quantity, or receiving an external signal. In this example, boiloff ends when it receives a "3" signal, and slow fill ends after the time in the expression 4_6_3_7_1(11), which is defined in the Excel input file.

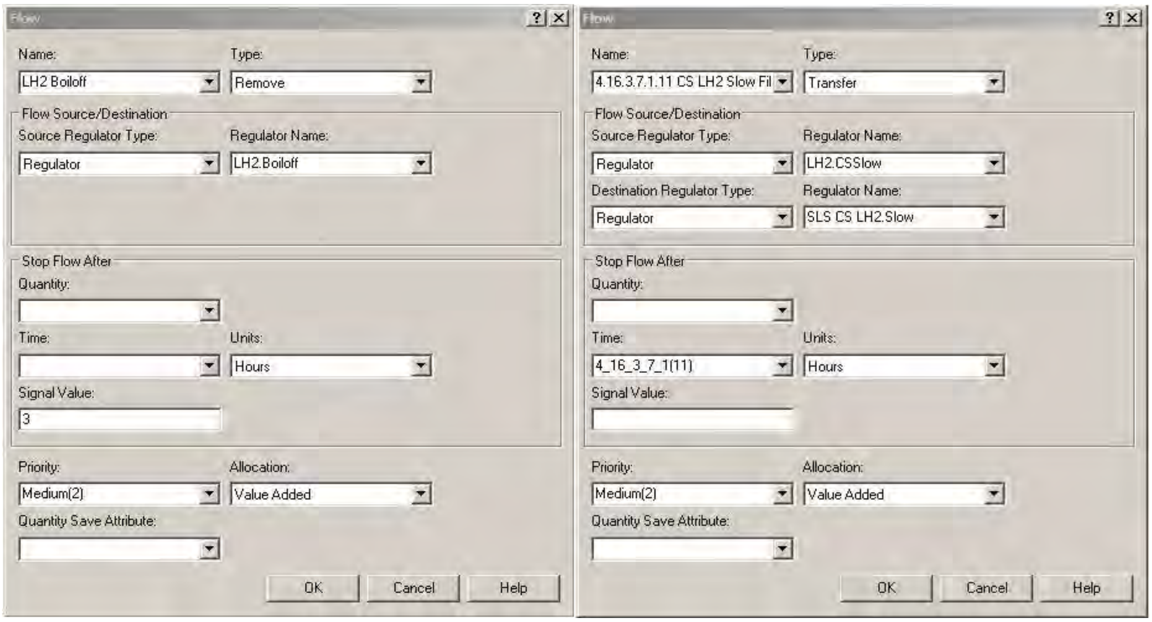


Name: LH2 Dewar
Capacity: 0
Initial Level: 0
Regulators:
LH2.Boiloff, 0, Per Day
LH2.TankerLoad, 0, Per Ho
LH2.Regulator 1, 0, Per Hou
LH2.CSChilldown, 0, Per Ho
LH2.CSSlow, 0, Per Second
LH2.CSFast, 0, Per Second
LH2.CSTopping, 0, Per Sec
LH2.CSReplenishProp, 0, P
Use As Packaging Tank
☐ Packaging Input Connection
☐ Packaging Output Connection
☑ Animation
☑ Report Statistics

Name: SLS CS LH2
Capacity: 0
Initial Level: 0
Regulators:
SLS CS LH2.TransferLoss, (
SLS CS LH2.Regulator 1, 0,
SLS CS LH2.Slow, 0, Per Se
SLS CS LH2.Fast, 0, Per Se
SLS CS LH2.Topping, 0, Pe
SLS CS LH2.ReplenishProp,
SLS CS LH2.ScrubBoiloff, 0,
SLS CS LH2.Bleed, 0, Per S
Use As Packaging Tank
☐ Packaging Input Connection
☐ Packaging Output Connection
☑ Animation
☑ Report Statistics

Figure 3: Tank Module Properties Example.

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, like the slow fill expression just mentioned, which is the range named 4_6_3_7_1 line number 11. 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.

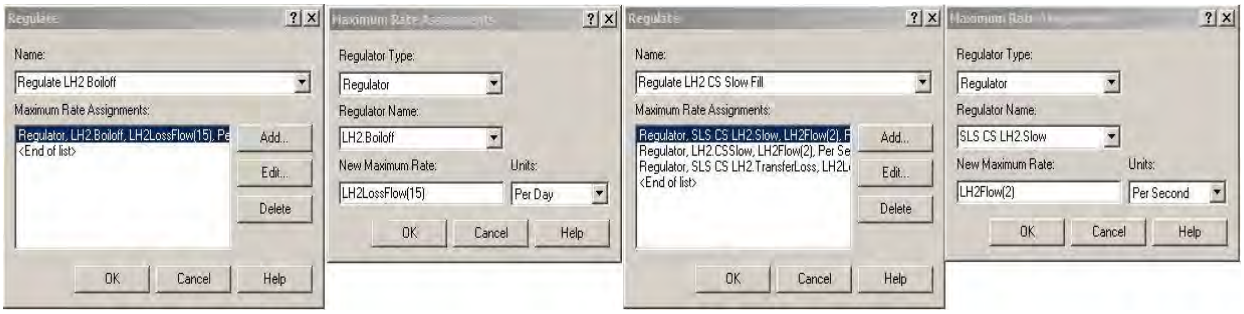


Name: LH2 Boiloff
Type: Remove
Source Regulator Type: Regulator
Regulator Name: LH2.Boiloff
Quantity:
Time:
Units: Hours
Signal Value: 3
Priority: Medium(2)
Allocation: Value Added
Quantity Save Attribute:

Name: 4.16.3.7.1.11 CS LH2 Slow Fil
Type: Transfer
Source Regulator Type: Regulator
Regulator Name: LH2.CSSlow
Destination Regulator Type: Regulator
Regulator Name: SLS CS LH2.Slow
Quantity:
Time: 4_16_3_7_1(11)
Units: Hours
Signal Value:
Priority: Medium(2)
Allocation: Value Added
Quantity Save Attribute:

Figure 4: Flow Module Properties Example.

The Regulate module is used like a valve that can change the flow rate of any tank regulator. The model primarily uses Regulate modules for flow rate inputs 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 value in the LH2LossFlow(15) expression. The LH2 CS slow fill regulator regulates both LH2 Dewar and CS tank rates because both regulators have to have the same flow rates or else Arena will default to the smallest one, which might not be correct. 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 when the process repeats. The entity that was seized by the Flow module then moves on to the next Arena module. The last module utilized for this study was the Signal module, which could send a signal to the model that a process started or finished, and it 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.



Name: Regulate LH2 Boiloff
Maximum Rate Assignments:
Regulator, LH2.Boiloff, LH2LossFlow(15), Pe
<End of list>

Regulator Type: Regulator
Regulator Name: LH2.Boiloff
New Maximum Rate: LH2LossFlow(15)
Units: Per Day

Name: Regulate LH2 CS Slow Fill
Maximum Rate Assignments:
Regulator, SLS CS LH2.Slow, LH2Flow(2), F
Regulator, LH2.CSSlow, LH2Flow(2), Per Se
Regulator, SLS CS LH2.TransferLoss, LH2L
<End of list>

Regulator Type: Regulator
Regulator Name: SLS CS LH2.Slow
New Maximum Rate: LH2Flow(2)
Units: Per Second

Figure 5: Regulate Module Properties Example.

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. Each process that was supposed to trigger a flow task had a Signal module that sent a global signal indicating that the process started or finished. The model duplicated the launch entities at time 0.00 in the model so that there was one for every GN2 and GHe task because many of these tasks happened simultaneously and each flow module needed its own entity to trigger it. Hold modules were placed before every flow module like start gates, and they held the entities until they received the required signal, which released the held entity to start a flow task. The flow did not stop until it received a second signal that was triggered by a later event. In Figure 6 the Hold module is waiting for the "12" signal at the start of LH2 tanking before releasing the entity to initiate the GHe purge at the LH2 Dewar. Helium purges are used extensively by NASA to inert the air during hazardous flow tasks, like when liquid hydrogen is flowing. Using this signal system linked all of the flow processes to the established GOPD timeline, and thus its variability as well, because if a processing task took longer to finish then the flow task had to wait for that process to end and send the signal before it could shut off the commodity flow.

Figure 6: Flow Modules Example with Hold.

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.

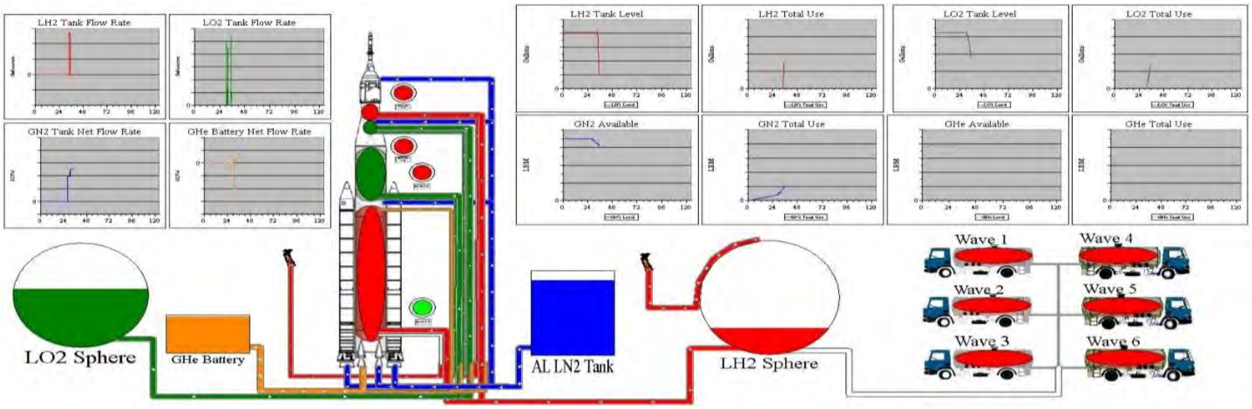


Figure 7: LCIC Model Animation, with four commodity tanks, SLS, and extra LH2 tankers (not to scale).

The data collection and modelling team frequently met with stakeholders and commodity experts to validate the model outputs and look for any missing flow tasks. We looked over every input to make sure they were correct, made updates as specifications changed, and we used the animation to spot unusual events that indicated that either the inputs or model were wrong. After months of data collection and model revisions the model was accepted by the KSC Integrated Engineering Review Board to be used as a program planning tool.

2.4 Problem Solving

There were several quirks in the Arena 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 real 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. It is important to note that the tanks in the model represent existing equipment that NASA did not want to augment if at all possible. A related problem was that 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 if there was nothing left to flow. Our novel approach to solving these simulation practitioner problems was to add pseudo 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 pseudo tank capacity the expressions for level outputs were subsequently adjusted by subtracting the pseudo 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 so there is one for every flow task. A result of this duplication and the significant use of signals was that sometimes processes or holds that were waiting for a signal might not release the entity when a signal is triggered. If several signals are in series without something else in between then the signals could all be triggered before the modules waiting for a signal can react because all the events happen at the same time in the events calendar. 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 (the time between finishing tank fill and T-0); 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. The *launch window* is the period of time in which a vehicle can be launched and reach its destination, and can range from minutes to hours in length (assumed to be 2 hours for SLS). A vehicle can launch at any time during the window, and it *misses* its launch window, and is officially scrubbed, if it does not launch before the predetermined end time.

The logic for determining whether the attempt missed the launch window (MLW) is made up of three decision modules: first or second attempt; missed or did not miss the launch window; and deterministic or stochastic. (Figure 7) The first decision is used because the processing times are different for each attempt, and the assign modules after it determine whether the elapsed time in the model has surpassed the total time allowed for processing, which is the sum of the nominal durations for critical path activities plus additional hold time that acts as a buffer for variability. This extra time simulates starting the launch campaign processing earlier than the nominal time before a launch window like in real launch plans. If the processing time surpassed the launch window time then the attempt is counted as a MLW by the second decision module. The last decision and assign modules determined the actual hold time that would occur before the vehicle "launches" in the model.

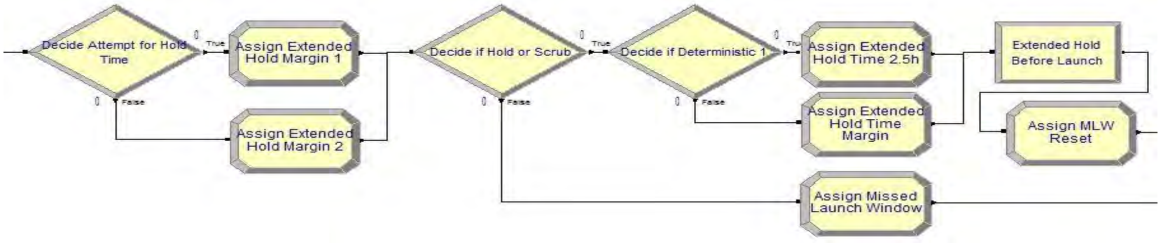


Figure 8: Missed Launch Window and Hold Time Logic.

The key to this logic was then determining the length of the buffer time. Running the simulation with 11 different hold time options produced the LH2 failure rates and MLW rates in Table 1. Initially the goal of the analysis was to find the time where the rates were equal (because they have an inverse relationship), but after examining the results for the number of attempts that no LH2 failures or MLW occurred, we found that the highest percentage of successful launch attempts occurred when the buffer time was 4.25 hours instead of when the failure rates were equal at 5.33 hours. Therefore, 4.25 hours was added to the nominal processing time for subsequent stochastic runs.

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.

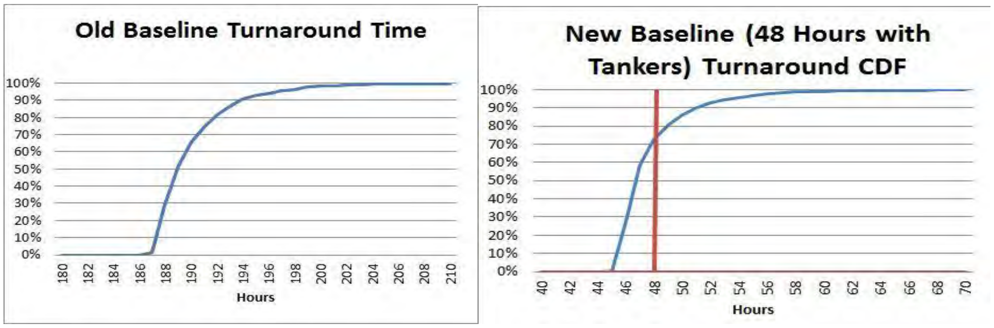


Figure 9: Scrub Turnaround Cumulative Distribution Function.

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 (while LH2 is flowing) because of safety purges.

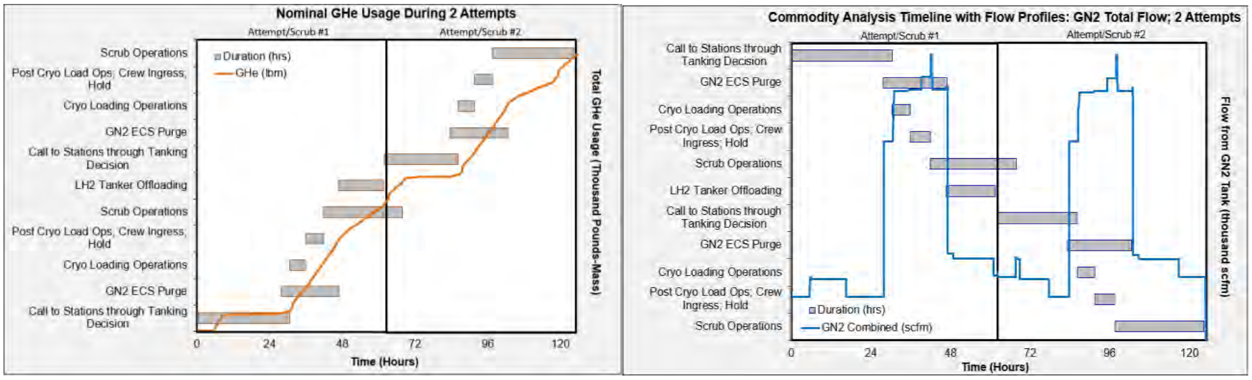


Figure 10: Deterministic GHe Usage and GN2 Cumulative Flow Rates During 2 Attempts.

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 follow a similar level pattern to LH2 except without refilling the source tank. The LO2 supply could break during replenishment on the second attempt in extreme cases, but it only went empty once during the 1000 campaign replications.

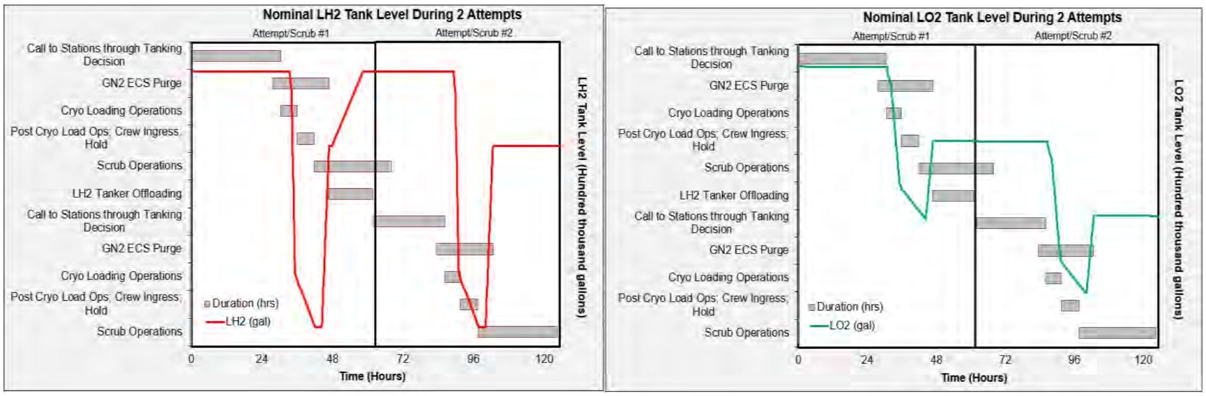


Figure 11: Deterministic LH2 and LO2 Tank Levels during 2 Attempts.

3.3 Failure Rate

Figure 11 is especially notable for how low the LH2 level goes by the end of tanking, which leads into Figure 12, 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.

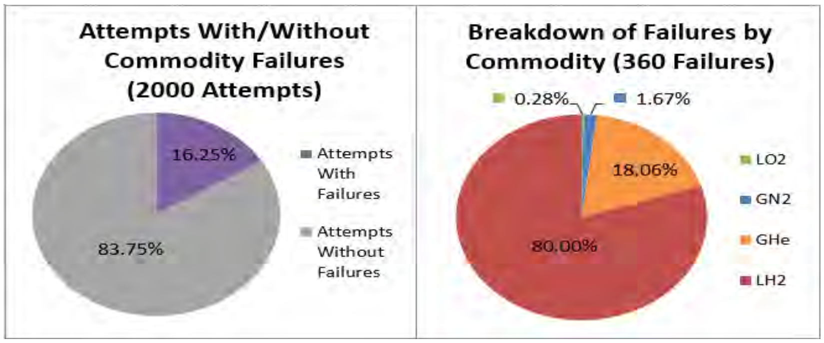


Figure 12: Stochastic Failure Rates of Commodities.

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. Also, considering the frequency of LH2 and MLW failures, NASA managers will need to decide if extra capacity is necessary, if the timeline needs to be changed, or if the risks are acceptable as is.

If this project were to be done again we would try to get the complete picture of the flow rates and interfaces before starting the Arena model. Many data updates caused a lot of modelling rework. Going forward, flow rates must be maintained on the input file as they are updated, so a data submission process and schedule between analysts and SMEs will make this process much easier.

5 CONCLUSION

The LCIC model successfully demonstrated the use of a combined DES and DRS technique to address challenges faced by GSDO in supporting the launch of the SLS. The novel approach of adding pseudo capacity to existing storage tanks improved the utility of the model and is a technique that can be used by other simulation practitioners in the future. In addition, these same methods can be used to analyze dynamic flow systems to estimate usage and when combined with interface models can very closely predict totals. This has proven the use of discrete event simulation in combining operational flows with complex engineering interface analysis.

GSDO is trying to gain the most flexibility from heritage Shuttle systems that would be extremely expensive to replace. The Program can make better decisions through understanding the probabilities of meeting 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. Assessments used to take place individually, 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 data that are essential to future SLS and GSDO plans and designs.

ACKNOWLEDGEMENTS

Thank you to Rogelio Franco for your guidance and expertise in completing this project, to Etosha Ottey for your support during data collection and continuing the work of this project in the future, to Danielle

Walsh for your support in understanding the process and data collection, and to Nate Rychlik for your advice during the project and guidance for this paper. 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., 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.

Drake, B.G., "Human Exploration of Mars, Design Reference Architecture 5.0" NASA-SP-2009-566, July 2009.

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.

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

Trocine, L., N. Cummings, A. Bazzana, N. Rychlik, K.L. LeCroy, G.R. Cates, M. Mollaghasemi. 2010. "Statistical and Probabilistic Extentions to Ground Operations' Discrete Event Simulation Modeling." AIAA SpaceOps 2010 Conference, Huntsville, AL, April 25-30, 2010.

Watson, M., E. Staton, G. Cates, R. Finn, K.M. Altino, K.L. Burns. 2014. "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 was a simulation analyst for the GSDO Program at Kennedy Space Center. 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 B.S. from the University of Central Florida and his M.S. from the University of Miami, both in Industrial Engineering. 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.