https://ntrs.nasa.gov/search.jsp?R=20180004457 2019-04-30T03:57:15+00:00Z

28th annual INCOSE international symposium
Washington, DC, USA
July 7 - 12, 2018

A Pilot Study to Determine MBSE Utility for Process Modeling of Complex Interfaces

"Skip" Clark V. Owens III
NASA-LSP Kennedy Space Center
Mail Code: VA-G2
Kennedy Space Center, FL 32899
1+321-867-2935
skip.owens-1@nasa.gov

Alexandra M. Dukes
Purdue University
701 W. Stadium Ave.
West Lafayette, IN 47907
1+530-906-1879
dukesa@purdue.edu

Shaun Daly
NASA-LSP Kennedy Space Center
Mail Code: VA-G2
Kennedy Space Center, FL 32899
1+321-867-8400
shaun.m.daly@nasa.gov

Notice for Copyrighted Information

This manuscript is a joint work of employees of the National Aeronautics and Space Administration and independent contractors under Contract NNX13AJ45A with the National Aeronautics and Space Administration. The United States Government may prepare derivative works, publish or reproduce this manuscript, and allow others to do so. Any publisher accepting this manuscript for publication acknowledges that the United States Government retains a nonexclusive, irrevocable, worldwide license to prepare derivative works, publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes. Published and used by INCOSE with permission.

Abstract. Modeling a full system or a complete interface between systems in a MBSE environment is a very large task and not all organizations will benefit enough from using MBSE to offset the effort that is required to do this. Completely modeling a system or interface is not necessary to evaluate the utility of MBSE for a specific application or organization. A small pilot can be executed over a short period of time that only models small portions of a system or interface and, if structured properly, this pilot can successfully demonstrate the utility of MBSE for an organization before having to invest a larger amount of resources to fully implement and deploy MBSE. This paper documents one such pilot that was conducted for NASA's Launch Services Program.

MBSE Pilot Objective

The NASA Launch Services Program (LSP) MBSE pilot had a very specific objective, but unlike most pilot programs, this objective was not to start using Model-Based Systems Engineering (MBSE), but rather to answer this simple question..."Do the benefits of MBSE outweigh the modeling efforts (cost) required to sustain the use of MBSE for the LSP?"

Right from the start, it was not known whether this pilot would result in the LSP adoption of MBSE, result in deciding to not adopt the use of MBSE, or if we would simply decide to wait to adopt its use until a later date when our launch vehicle contractors and spacecraft customers had adopted its use more widely within their programs and projects. Since the objective of our pilot was to determine both the utility of MBSE for the LSP and the effort required to achieve that utility, we

needed to identify potential uses of MBSE that were specific to the LSP. Choosing what to model during the pilot involved taking into consideration both the overall scope of what our specific group, the Integration Engineering (IE) group, does within the LSP but also the limited duration we had to execute the pilot program. Due to the very challenging workload of our Integration Engineering group we needed to acquire additional resources in order to have a successful pilot program. The additional resource we decided to use was that of a student intern. NASA has a very robust internship program that allows current students to come work at NASA for a Fall, Summer or Spring semester. The LSP MBSE pilot was run during the Summer of 2017, which was only 10-weeks in duration. The student chosen for the internship was Alexandra Dukes, who is currently attending Purdue University pursuing a master's degree in Aeronautical and Astronautical Engineering with a focus in Aerospace Systems, is also a co-author on this paper.

The limited duration of the pilot led to the decision to only model small portions of the system/interface, rather than attempting to model the entire scope of the system interface for which we are responsible. There are a couple of benefits to this approach. Modeling, when done correctly, takes a considerable amount of time and effort. Not only is the modeling effort itself very time consuming, but the initial task of fully understanding the system before modeling it also takes a significant amount of work. If a large and complex item is chosen, then there is little time left over in the schedule to allow the team to change the modeling approach or decide that modeling that aspect of the system interface is not value added and then move on to the next modeling task. The ability to iterate and change direction based on lessons learned during the pilot was the main strategy for our pilot. Having a student intern performing the modeling also had to be taken into consideration when scoping the pilot. A student intern was a good analog for the systems engineers in our Integration Engineering group that were not already using MBSE, but it also meant that the summer intern needed to both learn the specifics of our MBSE tool and learn about the role of the Integration Engineering group within the LSP.

An Overview of LSP Integration Engineering

To assist our intern in expediting the familiarization process within our organization, we started simply by moving between physical groups and stopping to interview and explore examples of common interactions between team members within LSP. Further exposure to our team’s work regarding our external customers was accomplished through attendance at mission level meetings. This initial work sought to explain the role of Integration Engineering (IE) as the end-to-end systems engineering lead between launch vehicle and spacecraft who are seeking access to space. During the familiarization phase, we examined previous mission requirements, requirement verifications, and familiarized through the use of shadowing. Shadowing was necessary beyond book level examples because IEs provide technical leadership and direction to the Mission Integration Team (MIT) across varied engineering disciplines internally within NASA and interact externally with both launch vehicle providers and spacecraft developers.

This early work was followed up through mentee to mentor questions on a daily basis, weekly progress meetings, and further supported by the use of interviews with discipline experts within the LSP, so our intern began to understand why the NASA Launch Services Program is known as "Earth's Bridge to Space." It became evident that MBSE, if utilized within our organization, would not be for hardware development as many others currently use MBSE for their projects, but we instead look to model the processes our team uses regularly. These regular operations and product development occur between when spacecraft development is authorized to proceed (Phase C see figure below), when LSP has procured the launch services from a commercial offeror, and through integrated operations up to operations on orbit (Phase E see figure below).

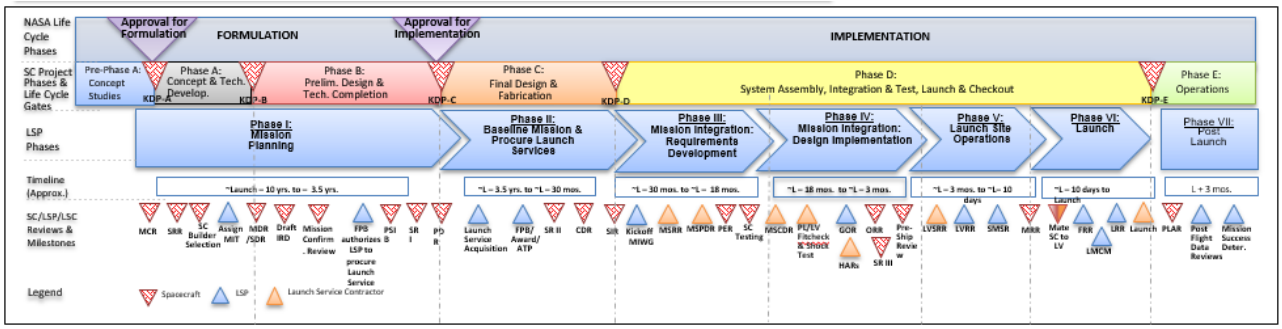

Figure 1. LSP Mission Support Phases

The primary focus of a LSP IE is to manage the interfaces between the launch vehicle and the spacecraft, ensuring requirements are developed and verified all while considering the safety of all teams and ensuring timely integration of the spacecraft with the launch vehicle for delivery on orbit. LSP IEs start this process early in the mission planning and development stage of the spacecraft project. This early involvement helps establish the spacecraft environmental test levels. The LSP IE's involvement continues through the spacecraft build, verification of integrated requirements, major spacecraft and launch vehicle design reviews, integrated operations, launch, and early orbit operations.

Integration Engineering is responsible for working with our spacecraft customer on the development of the spacecraft Interface Requirements Document (IRD) and later with our launch vehicle contractor to develop the Interface Control Document (ICD). The ICD is where all of the spacecraft to launch vehicle interface requirements are defined and verified. NASA LSP IE's have, over time, developed a set of best practices (lessons learned) we share with our spacecraft customers which has been proven to be helpful to both new and returning spacecraft development teams. Within our own internal team though, there are often areas which have a high level of input from different disciplines in a short timespan approaching the launch date. These points of debate are what we sought to first consider when modeling in MBSE. As we noted, there were interconnections we could see at the system level between disciplines that our disciplines were not themselves realizing existed. Identifying these and other common processes became a manageable subset of activities we could examine in the short pilot timespan. By exploring these areas, we believed we could share the pilot study's results with our internal teams to improve our interactions and develop better communication for both in times when teams are working linear and parallel verification closures and in the service to our customers.

MBSE Modeling During the Pilot

The Mars 2020 mission was chosen as the basis for the LSP MBSE Pilot for several reasons. The primary reason for selecting Mars 2020 for our pilot was that there is a significant amount of similarity between the Mars 2020 mission and that of the Mars Curiosity Rover (also known as Mars Science Laboratory (MSL)), which has been operating on Mars since August of 2012. Mars 2020 is using the same basic rover, cruise stage, and Entry Descent and Landing (EDL) designs as MSL. Even more importantly, the Mars 2020 mission has nearly identical interfaces and operational requirements with respect to the launch vehicle. Therefore, for the purposes of our MBSE pilot, we were able to use the large amount of historical engineering data products and verification closure data from the MSL mission as a starting point for our Mars 2020 MBSE pilot. This is a significant advantage compared to other MBSE pilot efforts that are attempted on active projects because we do not run into the issue of not having engineering closure data to complete the modeling process since the full life cycle of engineering products is available right at the start of the pilot.

The other advantage to using Mars 2020 as the basis for the LSP MBSE pilot is that the Jet Propulsion Laboratory (JPL) is the NASA organization in charge of the spacecraft. JPL has played a major role within the NASA community with respect to piloting the use the MBSE within the agency. If LSP were to eventually adopt the use of MBSE, JPL would be one of our first spacecraft customers who would be interested in engaging within that environment.

The MBSE Modeling Approach

The approach taken for the pilot study was performed in three steps: determine the needs of the LSP, develop an understanding of the system that would best test MBSE's ability to meet that need, and then create a model to determine that ability. This process was repeated often throughout the 10 weeks to refine both the needs and model as the capabilities of MBSE and the software were identified.

The main objective in answering the question, "Do the benefits of MBSE outweigh the modeling efforts (cost) required to sustain the use of MBSE for the LSP?" led to several driving questions that were asked at the start of each modeling effort within the pilot study, including:

- What is the need that is being addressed by the pilot study?
- How can MBSE make a productive addition to the LSP?
- What should be modeled and what should not be modeled?

Due to the time constraint, these questions were crucial to the success of the pilot study as they created a direction to properly evaluate the utility of MBSE for the LSP. The pilot study began by addressing three key needs within the LSP. The group wanted a more efficient way of performing their verification peer reviews, a visual representation of their integrated operations, and a better understanding of the relationships between internal discipline teams and contractor parties throughout the requirement verification processes.

One strategy for addressing these needs was to start modeling by creating small test cases and expand the model with time permitting. The goal of the pilot study was to determine whether MBSE would be a useful tool for the LSP, not to deliver a complete model. Starting small allowed us to evaluate the usefulness of the model artifacts as well as change the direction of the model based on lessons learned as the model was developed. Another strategy utilized during the study was to treat MBSE as a supplement to current document-based processes and not as a replacement of the current processes. This allowed us to focus on what MBSE could provide that current processes cannot. This strategy, in turn, created an interesting inquiry into how MBSE could be integrated as a supplemental model application to an existing process. In order to shorten the learning curve required to properly use MBSE, Lenny Delligatti's book 'SysML Distilled' (Delligatti, 2014) as well as the consulting services of the author through his company, Delligatti Associates, was utilized throughout the duration of the pilot study. His insights were invaluable in creating a greater understanding of the SysML language and MagicDraw, the software chosen for the study, as well as grounding our needs and efforts with known MBSE applications and examples. His consultation sessions allowed for the time it would have taken to understand MBSE as a modeling application to be focused on evaluating the benefits of MBSE for the LSP.

Choosing Test Cases for the Pilot

Specific modeling, or test cases, were needed for the purposes of our pilot. Before the pilot started, we created a list of potential modeling activities and goals to consider for the purposes of the pilot.

Figure 2. Potential Goals & Modeling Tasks

We knew we couldn't possibly tackle every potential modeling case or objective with such a short duration pilot, so we chose the modeling cases we thought represented the greatest potential benefit but were small enough in scope to be modeled in a short amount of time. The test cases specifically chosen to address the objective of our pilot were the verification of the In-Flight Disconnect (IFD) system and the Mass Properties of the system. The verification of the IFD provided a hardware focused perspective in the MBSE artifacts meaning the focus of the model was on the hardware of the system itself and the verification processes carried out on that hardware. The verification of the Mass Properties provided a process focused perspective in that the verification of the mass properties does not focus on a specific system but is a process involving the entire system and several groups both within and outside of the LSP throughout the system's development until launch. These two test case studies, and their varied perspectives, provided a platform to address the identified needs of the LSP and the objective question. These two test cases also addressed the three key needs identified in the previous section of the paper (more efficient verification peer reviews, visual representation of integrated operations, and a better understanding of the relationships between internal discipline teams).

After identifying the needs and test cases, the second step of the modeling process is developing an understanding of the systems chosen for the test cases. This step involves "pre-coding" the model, or defining the model elements and their relationships to other identified elements before modeling

the system. The information necessary to model the MBSE test cases were contained across multiple documents from multiple sources including NASA documentation and subsequent contractor documentation for the payload and launch vehicle. This step aimed to answer the questions: What system elements (i.e. actors, hardware, and requirements) should be modeled? What are the relationships between those elements? What are the verification activities involving those elements? Assembly of the elements began with a search of the system of interest within the requirements documentation. The search was then expanded to requirements that are directly related to the first set of requirements identified by the system of interest. All verification activities for the identified requirements were gathered and the hardware, actors, and steps noted within the verification documentation were recorded. A timeline, per the NASA lifecycle phases, of the verification activities was then constructed. The information gathered through this process was recorded in Excel. The pre-coding allowed for organization of the model before construction of the model and allowed for greater understanding of the elements and their effect on each other per their requirements, verification activities, and shared ownership within and outside of the LSP. This was the most time-consuming step in the modeling process and emphasized the benefit of MBSE in its ability to visualize the relationships between actors, hardware, requirements, and verification processes in an efficient manner.

The final step in our modeling approach was creating the model from the test cases that would allow us to evaluate the benefit MBSE would provide to the LSP and the cost of constructing the model. The SysML language and modeling tool MagicDraw were selected for the pilot study due to its use at other NASA centers and its availability to LSP. Due to the nature of the LSP's needs, the model focused on the requirements and verification processes and the actors and hardware involved in those processes. Using the excel sheet assembled in the previous step, a requirements diagram was the first diagram created since the requirements are the unifying elements between the actors, hardware, and verifications as well as the most straightforward to model due to their established hierarchy within the existing documents. Verification activities and the actors who verified them were then added to the requirements diagram. From the verification activities, activity diagrams were created. An example of an activity diagram for a spacecraft shock test is presented below.

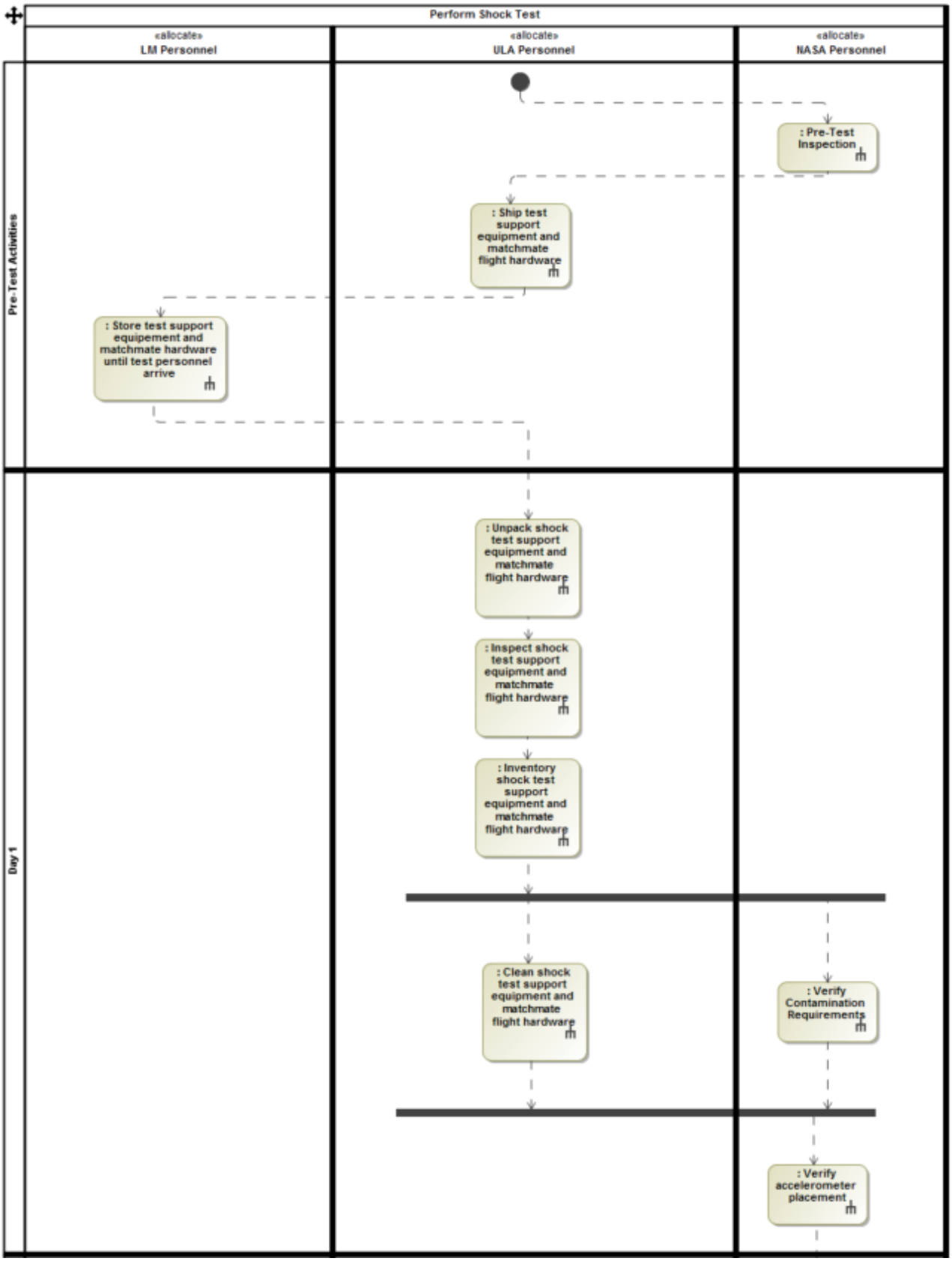

Figure 3. Example Spacecraft Shock Test Activity Diagram

In Figure 3 above, the columns indicate the organization that is responsible for the specified actions (i.e. the rounded corner rectangles) within the verification, and the rows indicate the time phase in which the activity occurs. The black horizontal bars indicate actions performed in parallel, or at the same time. The activity diagram is especially useful for communicating specific steps within a verification/activity or decisions that are required as part of the process. Before modeling this

operation in MBSE, this type of information was only available in a combination of lengthy spacecraft and launch vehicle procedures. Now, with MBSE, this single diagram not only shows the high-level operational flow, but also dynamically links to all the other modeling artifacts used within this operation in our model. The activity diagram in Figure 3 meets the specific need identified earlier in the paper, which is to visualize the relationships between actors, hardware, requirements, and verification processes in an efficient manner.

In addition to modeling operations with an activity diagram, we also generated a verification matrix as another MBSE proof of concept for our pilot.

	Actors/Disciplines								Mass Prop Requirements [MS							
	Controls	Electrical	Flight Design	Flight Software	Loads	Mechanical	MGSE	Stress	3.1.2.3 SC Hoist Design Loads	3.1.3.1 SC Mass Properties	3.3.4.1 Low-Mid Frequency Vibration	3.4.1 Target Specification	3.4.4.3 Reference Mission Targets	3.4.4.5 Orbit Injection Accuracy	3.4.7 Post-Separation	3.4.9 Trajectory Data
Mass Prop Verification Activities		2	1	12		3	2	1		1	3	3	1	4	1	4	2
MPDR [Phase III]		2	1	3		1	2				3	1		2			1
31310-0	2	↗	↗						1		↗						
31310-1	1						↗		1		↗						
31310-2	2	↗					↗		1		↗						
33410-2	1					↗			1			↗					
34431-1	1			↗					1				↗				
34432-1	1			↗					1				↗				
34900-1	1			↗					1								↗
Phase IV				1		2						2	1				
CLA						2						2					
33410-0	1					↗			1			↗					
33410-1	1					↗			1			↗					
Final Mission Analysis				1									1				
34100-0	1			↗					1				↗				
Phase V				4				1		1						4	
SC Ship								1		1							
31230-0	1						↗		1	↗							
Space Separation Analysis (CCAM)				4												4	
34701-0	1			↗					1							↗	
34701-1	1			↗					1							↗	
34702-0	1			↗					1							↗	
34702-1	1			↗					1							↗	
FRR [Phase VI]				4									2	1			1
34431-0	1			↗					1				↗				
34432-0	1			↗					1				↗				
34450-0	1			↗					1					↗			
34900-0	1			↗					1								↗

Figure 4. Example Mass Properties Verification Matrix

The matrix in Figure 4 is automatically generated by MagicDraw using the information provided from the requirement and activity diagrams. The matrix reads left to right with the rows displaying the verifications sorted by the lifecycle or milestone phase in which they occur, and the columns define the actors responsible for the verification and the requirement that is being verified. For example, the verification of mass properties is performed by eight different disciplines and verifies eight different requirements.

The current verification peer review process within the LSP Integration Engineering group involves going line by line through an excel sheet containing all the verifications and information about their verifications including the description, the responsible LSP group, and the point in time during the mission integration phase in which it occurred. The key problem with this process is the amount of time it takes to confirm, line by line, that all the verifications were performed by the correct actor(s) at the right time within the mission integration. The MBSE model's verification matrix allows for the identification of this information in an easy to read format which is automatically generated

from the previously built requirement and activity diagrams. This table allows for the ability to visually identify which verification activity verifies which requirement, which actor is verifying each requirement, and when each verification occurs within one view that was not present in the current excel based process. This artifact directly contributes to the need discussed earlier, which is a more efficient way to perform verification peer reviews and to gain a better understanding of the relationships between internal LSP discipline teams. Figure 4 is a tool the LSP IE would use to coordinate the creation and review of the LSP verification plans with the LSP discipline experts as well as a tool used during the formal peer review of the entire set of verification plans. The verification matrix in Figure 4 is also a product that can be re-used with very little effort, which reduces the total time/effort required to create and implement this across multiple missions.

Another useful diagram that can then be created within the MBSE environment is the block definition diagram, which can be used to associate the actors with the hardware they are responsible for during specific integrated operations (which typically involve LSP, and our spacecraft and launch vehicle contractor personnel). Figure 5 below is an example of using a block definition diagram from SysML to show which organizations are responsible for each piece of hardware in an integrated operation, which was one of the needs identified earlier in the paper.

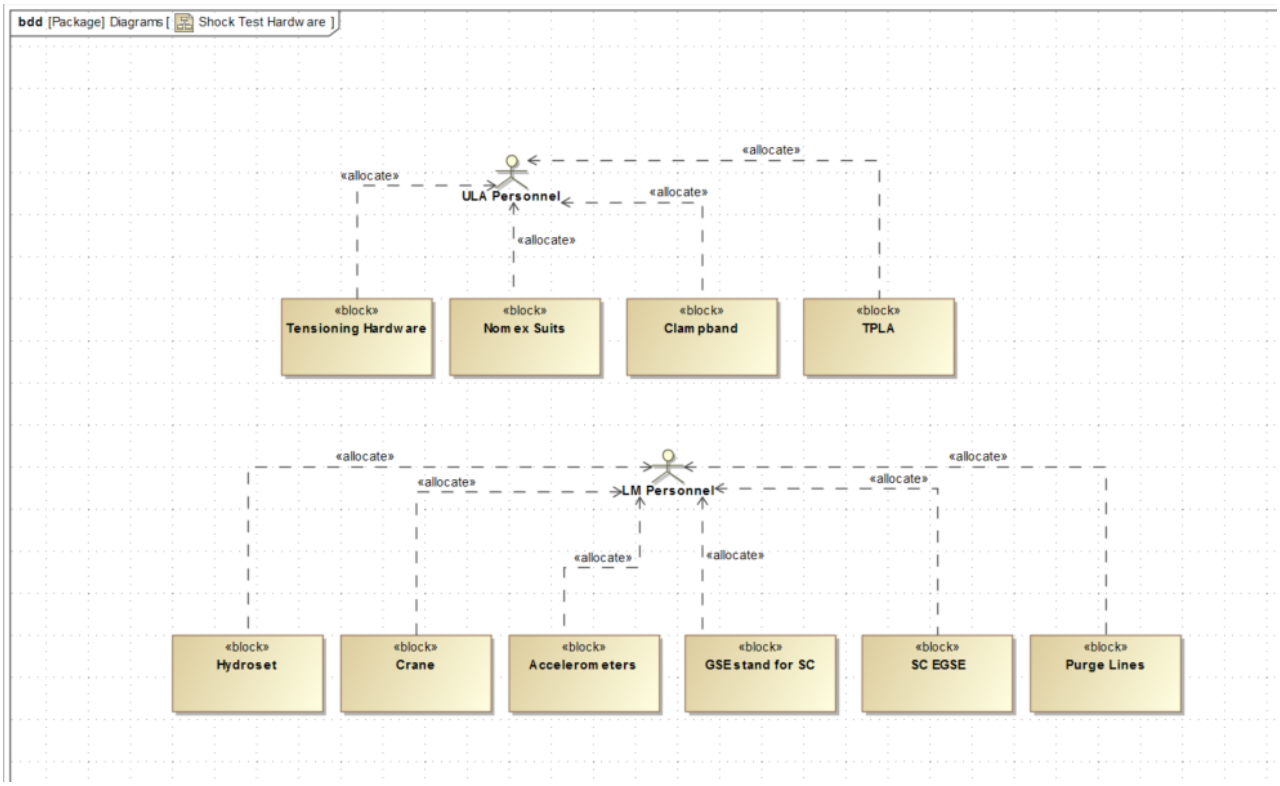

Figure 5. Example Hardware Diagram for a Spacecraft Shock Test

The block definition diagrams, requirements diagrams, and activity diagrams can all be assembled into a single interface through the use of a package diagram so that all the information can be accessed from a single location within the model as shown in Figure 6.

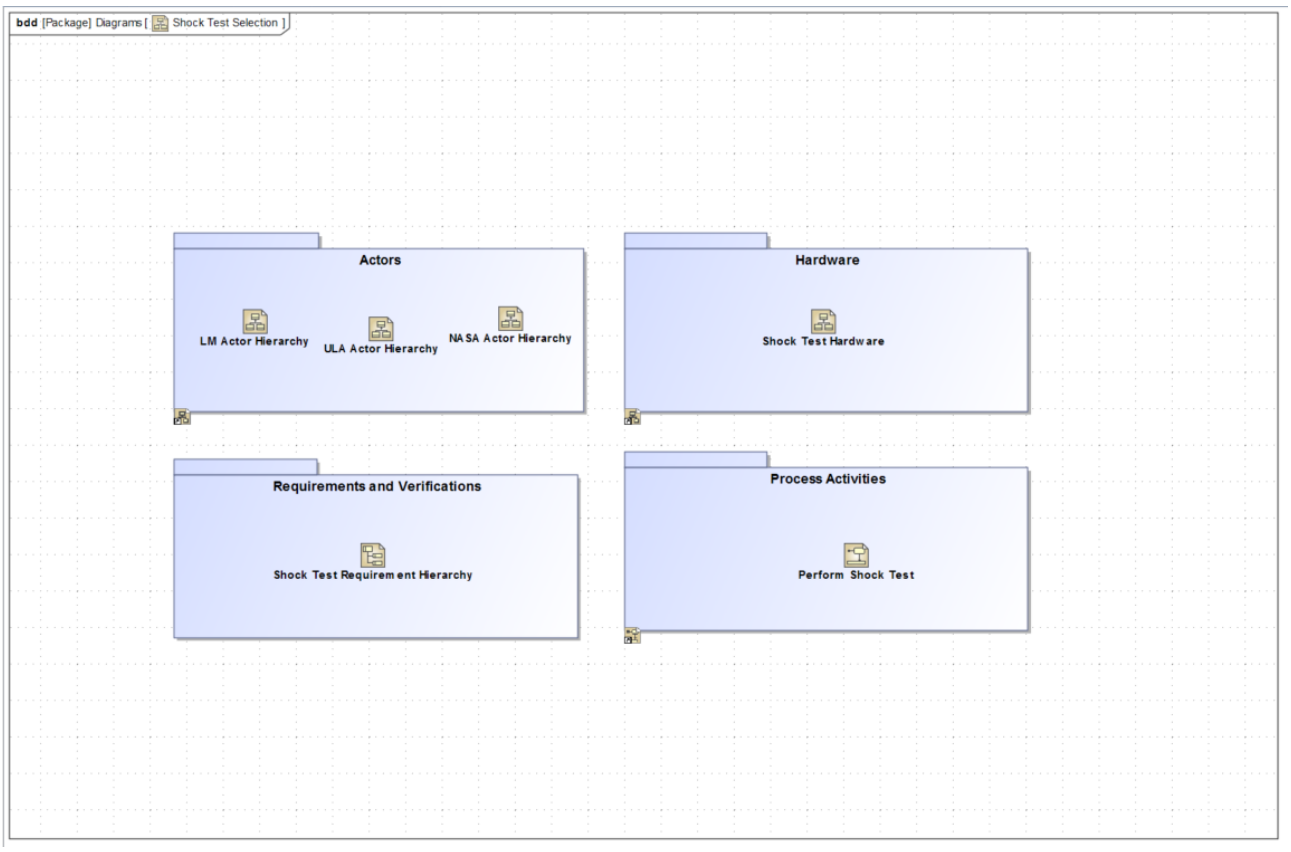

Figure 6. Example Package Diagram: Shock Test Operations Summary

While views similar to the package diagram in Figure 6 can be created in a set PowerPoint charts, the consistency of the elements and their relationships within the various diagrams, the ability to easily replicate previously made diagrams into different models, and the amount of time it takes to construct these diagrams within MagicDraw versus other flowchart software, make a MBSE approach a highly-desirable option for meeting the need to visually and dynamically depict integrated operations.

Conclusions, Next Steps, and Lessons Learned

The LSP MBSE pilot was executed with a single modeler, who had previous MBSE experience but was new to both Magic Draw and to the type of work done within the Integration Engineering organization within the LSP. Despite the learning curve with a new MBSE tool and a new organization, a single MBSE modeler was able to learn enough about the LSP processes and functions to create multiple sample MBSE products and demonstrate the utility of these products with respect to LSP needs.

MBSE is traditionally used as an application of modeling from conceptual design through development of a system; whereas, the system is 90% designed and developed by the time it reaches LSP. Additionally, this pilot study was unique in its application of MBSE as it focused on the verification of requirements and the processes within those verifications. We found that MBSE has enough potential to become a productive modeling application to LSP that it is worth pursuing future, larger scale, pilot studies which would work towards transitioning from asking the question “if” MBSE is useful to LSP, given its costs, to demonstrating “how” MBSE is useful to LSP.

In order to reach this conclusion, the pilot study needed to answer the questions “How is MBSE better than current processes?” and “What is MBSE’s value to LSP?”. The answers to these questions arise out of the pros we found of using MBSE for LSP. The key pros contributing to its potential productive addition to LSP are MBSE’s ability to:

- Utilize MBSE artifacts without being an MBSE expert
- Provide a single snap shot of the interrelations during LSP IE processes
- Provide process models of verification activities

The ability of the MBSE tools to create publishable model products without requiring everyone to know MBSE, or its tools, allow for those who wish to use the model, interact with it, and benefit from it without needing to understand the software. This was essential for LSP as it would not add to the burden LSP personnel already have in managing documents through multiple systems and software suites. This feature allows those who were interested in using MBSE to develop the models, and those who did not want to, or do not have the time to, use MBSE to still benefit from the models. Additionally, a developed MBSE model answers the questions, "Who is attending the verification activity?", "Which is responsible for bringing what hardware?", and "What is everyone's role during a verification activity?" with a click of the mouse rather than searching through LSP's current document-based process. This allows for a single, easy to read, view of the interrelationships between LSP's internal groups as well as those outside of LSP. It also provides information that is across multiple sources to be synthesized into one format, allowing for easier access to information pertinent to the successful execution of verification processes. Currently, verification processes are regarded as "tribal knowledge", and only exist as text which can be interpreted differently across the actors involved in the verification. MBSE allows for a pictorial view of the verification process that can be better communicated across attending actors and verifies everyone begins a verification on the same page. These three pros of MBSE provide a utility that LSP does not have in the current document-based process, and provides enough evidence to move forward with a larger scale pilot study on MBSE's potential role within LSP.

MBSE, while providing several benefits to LSP, has one major drawback: Time. Currently there is a large discrepancy between the time it takes to properly build a functional model that provides the benefits listed above versus the time LSP IEs have available to create the model. With the short 10-week duration of this pilot study, we have only scratched the surface of providing the right motivations for LSP to invest in building MBSE based models. With additional time and a larger scale study, LSP would have the ability to further determine if the benefits to integrated operations and verifications processes is feasible for current personnel, if a MBSE focused position within LSP is appropriate, or if the effort would not be worth its benefit. The initial results from this small pilot study show that the value produced from the models, especially when considering their potential for re-use, outweigh the costs/time it took to produce them.

The example activity diagram shown in Figure 3 was able to convey a "big picture" perspective and a timeline for a spacecraft shock test, all in a single diagram. Before creating this diagram in MBSE the only way we have been able to achieve getting multiple teams on the same page before a major operation was to have everyone read all of the applicable procedures for the operation and conduct a review of the operation in a conference room setting. While an activity diagram would not replace the existing practice of reviewing procedures and walking through the operation in a conference room setting, it would make a very valuable addition to our current process.

The example verification matrix showing mass properties verifications in Figure 4 immediately provided value. Within a few seconds of an IE reviewing this diagram, the IE was able to identify a few verifications that were missing. This was not so much an error in how the MSL mission verified mass properties but was an artifact of a process change that occurred between the time we processed the MSL mission and the present. The ability to quickly spot a missing or incorrect verification like we were able to do using this mass properties verification matrix was exactly the type of enhancement we were hoping for to our verification plan peer review process.

And finally, the block definition diagram showing hardware responsibilities in Figure 5 demonstrated the ability to quickly display in a single graphic the hardware responsibilities for a single operation.

That ability alone would not be all that valuable, but when coupled with the interactive and dynamic abilities of a working model, along with the capabilities of a package diagram summary example shown in Figure 6, we now have demonstrated the ability to satisfy another need…a visual representation of integrated operations.

Lessons Learned

The following is a list of lessons learned that were identified during the course of the 10-week LSP MBSE pilot activity:

- Pre-coding your engineering material before modeling within a MBSE environment, while time-consuming, was a very necessary and valuable step to ensure an accurate model
- The true power of MBSE does not lie with its ability to create "pretty diagrams" but rather with its ability to automatically generate engineering analysis (which can sometimes take the form of a diagram)
- A community of practice for interface management utilizing MBSE does not exist and in general a robust MBSE community can be hard to find due to the highly specialized nature of applying MBSE to a wide variety of systems and environments
- One organization's lessons learned concerning MBSE may not be applicable to another organization using MBSE
- An MBSE expert, whether it be a direct team member or a consultant, is an absolute "must have" in order to ensure a productive pilot

A Summary of the Pros and Cons of MBSE

The following is a graphical summary of the "pros" and "cons" that resulted from the 10-week LSP MBSE pilot activity:

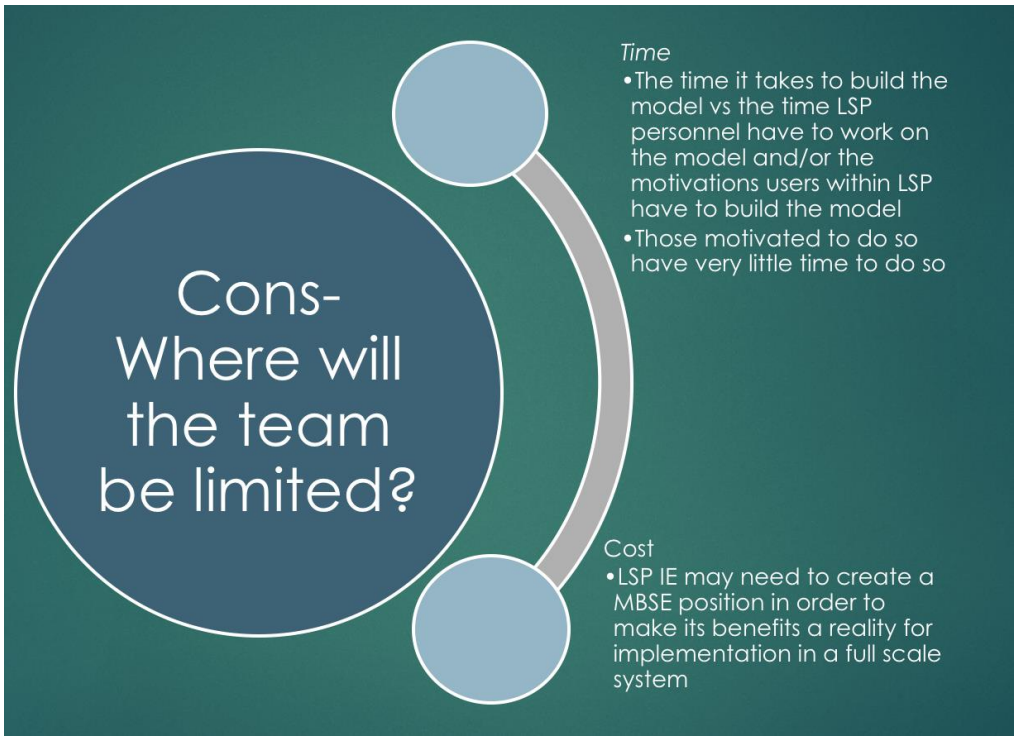

Figure 6. "Cons" of MBSE

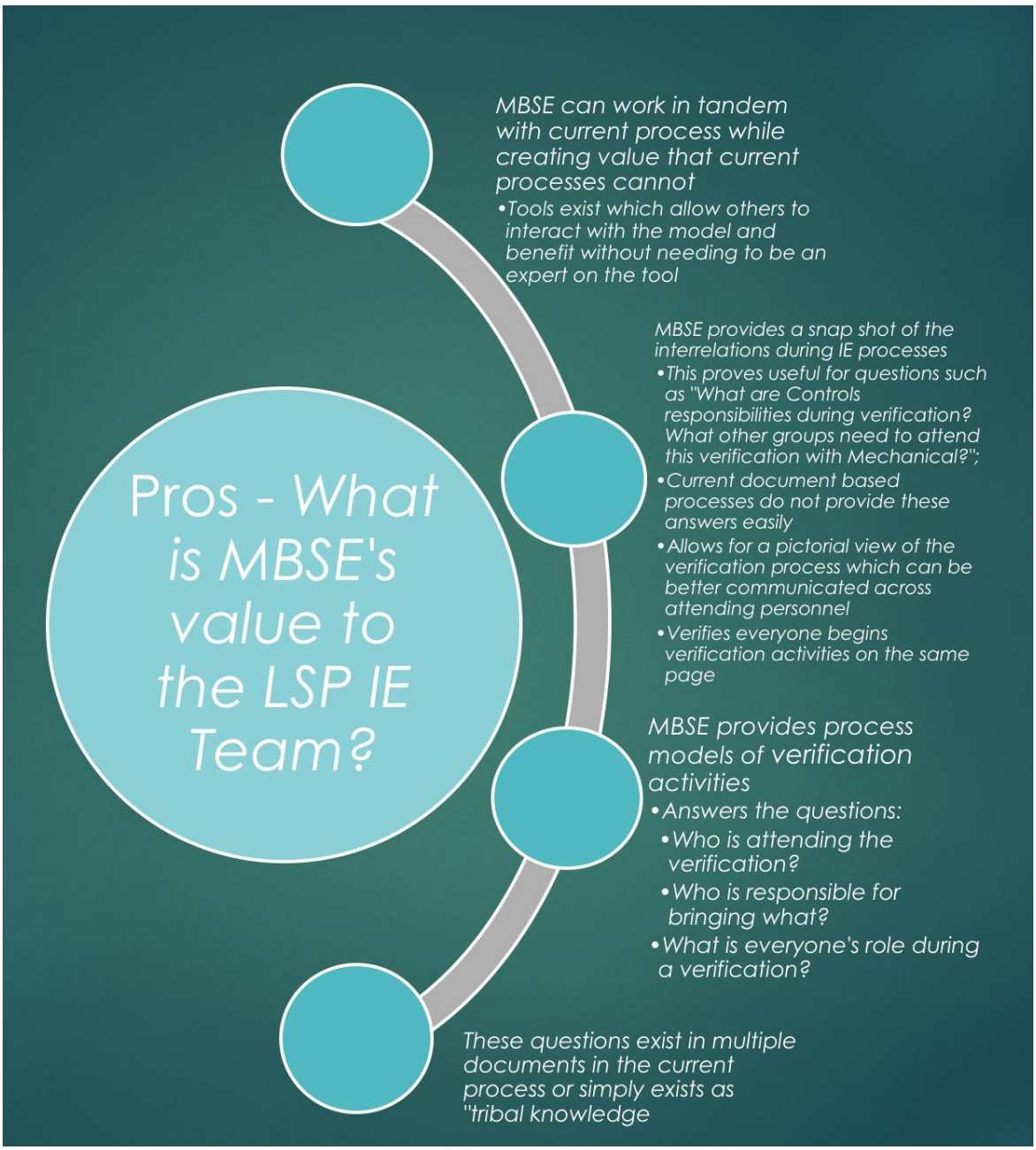

Figure 7. “Pros” of MBSE

References

Delligatti, Lenny. SysML distilled: a brief guide to the systems modeling language. Addison-Wesley, 2014.

Biography

"Skip" Clark V. Owens III is a systems engineer in the Integration Engineering Branch of the Launch Services Program at NASA Kennedy Space Center. Mr. Owens graduated from Wichita State University with a B.S. in Aerospace Engineering and has a M.S in Space Systems Engineering from the Stevens Institute of Technology. Mr. Owens has experience working space mission design and operations from both a spacecraft and a launch vehicle perspective.

Alexandra Dukes is a graduate student at Purdue University pursuing a Master's in Aeronautical and Astronautical Engineering with a focus in Aerospace Systems. Ms. Dukes graduated with her B.S. in Aeronautical and Astronautical Engineering from Purdue University, and currently works as a research assistant in the Systems of Systems laboratory on projects involving system architectures and MBSE applications for both federal and commercial space initiatives.

Shaun Daly currently serves as a systems engineer in the Integration Engineering Branch of the Launch Services Program at NASA Kennedy Space Center. Mr. Daly is a graduate of the University of Central Florida with a B.S. in Electrical Engineering and has a M.S. in Systems Engineering from the Florida Institute of Technology. Since transitioning to NASA from the U.S. Air Force, Mr. Daly has worked as both an Avionics engineer and in Integration Engineering focusing on technology development, STEM mentoring for nanosatellite development, and improving communication between distributed development teams.