Model-Based Systems Engineering Approach to Managing Mass Margin

5th International Workshop on Systems & Concurrent Engineering for Space Applications

- SECESA 2012 -

17–19 October 2012

Alameda Campus of IST / Technical University of Lisbon
Lisbon, Portugal

Seung H. Chung, Todd J. Bayer, Bjorn Cole, Brian Cooke, Frank Dekens, Christopher Delp, Doris Lam

Jet Propulsion Laboratory, California Institute of Technology
Pasadena, CA 91109, U.S.A.
Email: seung.h.chung@jpl.nasa.gov

INTRODUCTION

When designing a flight system from concept through implementation, one of the fundamental systems engineering tasks is managing the mass margin and a mass equipment list (MEL) of the flight system. While generating a MEL and computing a mass margin is conceptually a trivial task, maintaining consistent and correct MELs and mass margins can be challenging due to the current practices of maintaining duplicate information in various forms, such as diagrams and tables, and in various media, such as files and emails. We have overcome this challenge through a model-based systems engineering (MBSE) approach within which we allow only a *single-source-of-truth*. In this paper we describe the modeling patterns used to capture the single-source-of-truth and the *views* that have been developed for the Europa Habitability Mission (EHM) project, a mission concept study, at the Jet Propulsion Laboratory (JPL).

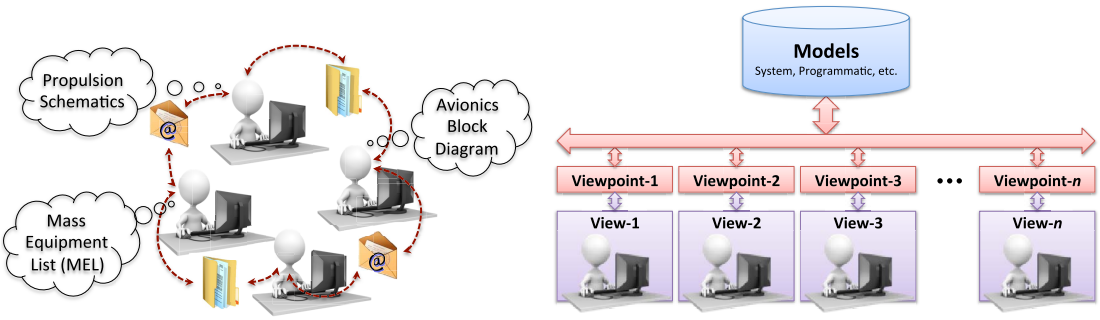
Significance of Mass Equipment List

A mass margin is generally associated with and generated from a MEL that lists all components of a flight system and their associated estimated mass properties. A MEL is typically visualized in a tabular view organized by subsystems. This view provides systems engineers a means to quickly gain insights to the completeness and correctness of the design and potential points of improvements and design change. This view also provides a means of communicating with the responsible subsystem engineers.

Concern for Information Consistency and Correctness

As stated before, generating a MEL and computing a mass margin is conceptually a trivial task. The challenge for systems engineers, however, is in managing them. While there are many considerations in managing mass, including programmatic and technical aspects, we will focus on the challenge of maintaining consistent and correct information. For example, a subsystem engineer may have his or her own list of the subsystem components and their mass properties, separate from and potentially inconsistent with the MEL maintained by a systems engineer. As another example, the set of components depicted in a flight system block diagram may be inconsistent with the MEL.

One of the reasons for such potential inconsistencies is due to engineers' varying perspectives, i.e. *viewpoints*, on a system design. Depending on the role and/or the discipline of an engineer, that engineer will represent information regarding a system design in a particular form, i.e. *view*, that is useful in addressing the *concerns* of that role and discipline. Some examples of views from varying viewpoints are MEL table, avionics block diagram and propulsion schematics. Note that while our use of the words *viewpoint*, *view*, and *concern* may be interpreted according to their general use, we use them specifically as defined in the ISO/IEC/IEEE 42010 formal systems and software architecture framework [5].



(a) Multiple-Sources-of-Truth (b) Single-Source-of-Truth

Fig. 1: (a) Current systems engineering practice of maintaing and exchanging information using files and emails. (b) A generic model-based systems engineering approach, under which the models are used as the single-source-of-truth.

As depicted in Fig. 1(a), such views are generally maintained and exchanged among engineers in files and emails. The views from these varying viewpoints, however, may contain overlapping information and therefore duplicate information. As an example, avionics block diagrams, propulsion schematics and MELs all specify the components or a subset of the components that make up a system. The duplicate information, i.e. *multiple-sources-of-truth*, can potentially lead to inconsistencies and eventually incorrect information regarding the system design. Furthermore, systems engineers cannot iterate rapidly on the system design because much effort is required in generating consistent and correct diagrams, MELs, and mass margins, i.e. the views, that are required.

Model-Based Systems Engineering Approach

The view consistency and correctness problem is typically managed, but not necessarily mitigated, using file-based configuration and version management, along with manual and laborious consistency checking among the files. Under the EHM project [3], we are instead taking a MBSE approach to managing MELs and the mass margins, among other technical margins [2]. We use models as the *single-source-of-truth* under which we have no duplication of information. In the following sections, we describe how we build the single-source-of-truth and how the views are generated from the single source of truth.

MODEL AS THE SINGLE-SOURCE-OF-TRUTH

Our general approach to managing consistency and correctness of the views is by allowing only a single-source-of-truth that is captured as integrated models in a single, but potentially distributed, repository. This is depicted in Fig. 1(b). The project information is visualized through views that are generated from the models according to the viewpoints. Conversely, the models are generated from the views, where the information in the views is transformed into models according to the viewpoints. This approach is a foundation for JPL's Integrated Model-Centric Engineering (IMCE) [1].

While what is depicted in Fig. 1(b) is a generic MBSE approach that is based on the concepts of viewpoints and views of the systems and software architecture [5], Fig. 2 is the EHM project's implemented MBSE framework. As shown in Fig. 2, the models are constructed using the System Modeling Language (SysML) [4]. In fact, SysML's approach to creating models is effectively through views and provides a predefined set of generic viewpoints for generating and visualizing models. As an example, one of the SysML viewpoints is the *Internal Block Diagram* (IBD) that is equivalent to block diagrams generally seen in systems engineering. The standard set of views and viewpoints provided by SysML is, however, insufficient in addressing some of the basic systems engineering concerns such as mass, power and data margins. Thus, the EHM project has created additional views and viewpoints to help address those concerns. In this paper we will only focus on the Block Diagram, Mass Equipment List, and Mass Margin views and viewpoints among others that the EHM project has developed. In this section we describe the EHM project's modeling patterns.

Modeling Products and Mass Characterizations

One type of model in the single-source-of-truth is a set of products, i.e. components, and their associated mass characteristics, referred to as *characterizations* within IMCE [1]. An example of a product called Product-1 and its mass

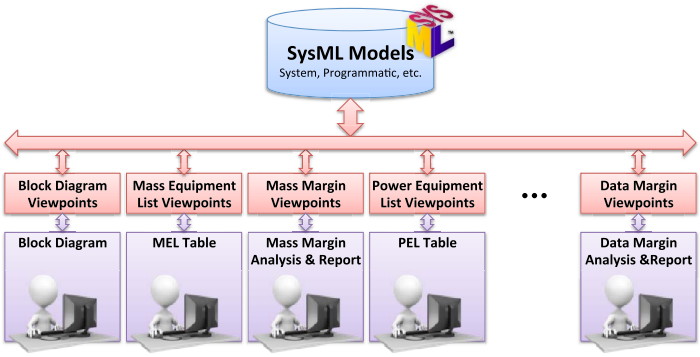
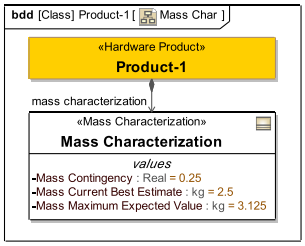
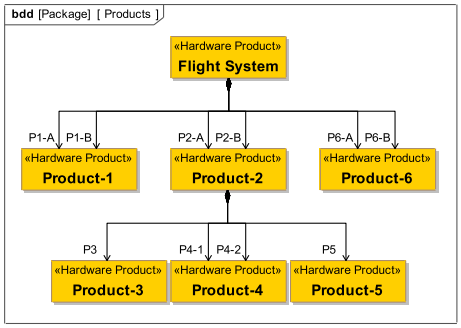


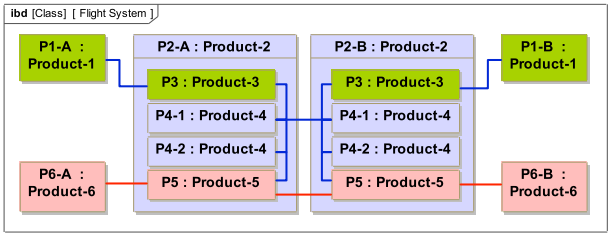
Fig. 2: The EHM project's model-based systems engineering framework.



(a) Mass Characterization



(b) Product Assemblies



(c) Block Diagram

Fig. 3: The modeling patterns in SysML for specifying (a) the mass of a product and (b) assemblies of products. (c) A SysML representation of a block digram, i.e. IBD, of Flight System shown in Fig. 3(b). Note that this IBD shows interfaces between the products, which are not shown in Fig. 3(b).

characterization is shown in Fig. 3(a). As shown in this example, Product-1 labeled ≪Hardware Product≫, a.k.a. a stereotype in SysML, is associated with a Mass Characterization, that is stereotyped ≪Mass Characterization≫. The stereotype ≪Hardware Product≫ specifies that it is a product with mass, in contrast to a ≪Software Product≫. The stereotype ≪Mass Characterization≫ specifies a characterization of mass properties, rather than power load or data production properties of a product. A Mass Characterization includes three properties: *Mass contingency* as a Real number, *current best estimate* (CBE) in kg and *maximum expected value* (MEV) in kg. The mass contingency, represented as a factor or a percentage of CBE, is the allowance for growth established to cover uncertainty. The MEV is the mass of a component that takes the contingency into account, i.e. $MEV \equiv CBE \times (1 + contingency)$. These properties are sufficient for keeping track of the mass margin. As we will discuss later, these properties are either calculated or manually specified.

Modeling Product Assemblies

Another type of model in the single-source-of-truth is the deployment of products into assemblies. Examples of assemblies are shown in Fig. 3(b). In this example, Product-2 is an assembly of one of Product-3, two of Product-4, and one of Product-5. Flight System is an assembly of two of Product-1, two of Product-2 and two of Product-6. While the mass characterization pattern is not a standard in SysML and was developed within the EHM project, the assembly composition

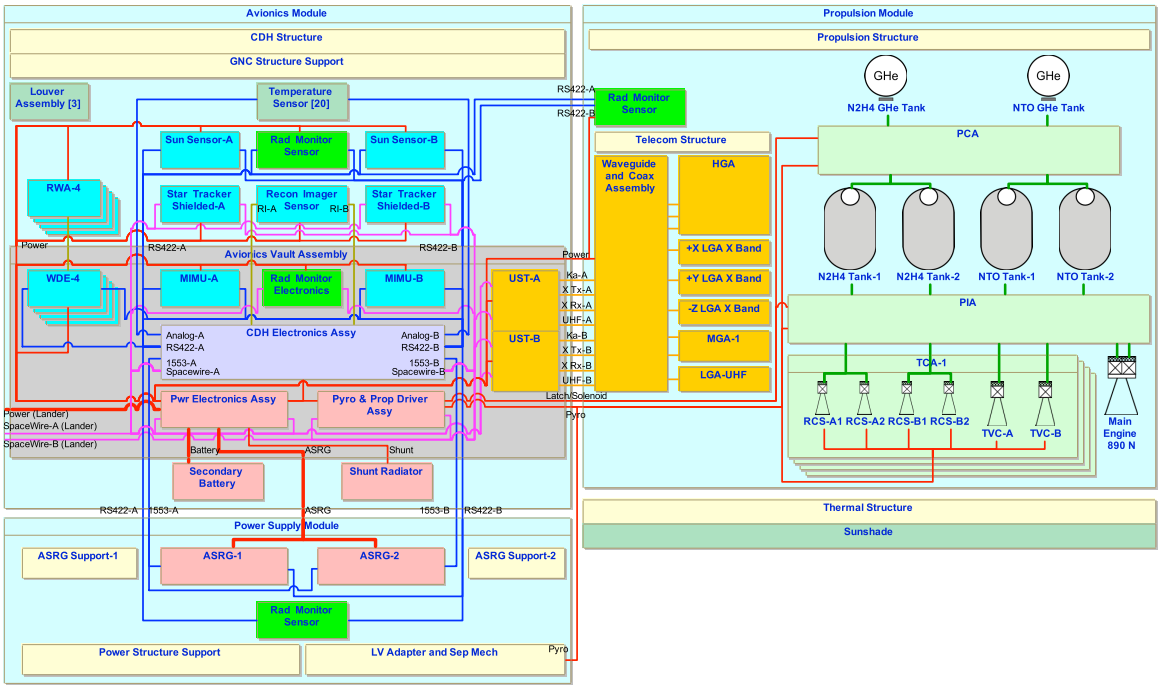


Fig. 4: The overall block diagram of the Carrier from the Europa Lander Mission study [3, pg. D-98]

pattern used for assemblies is a standard pattern of SysML, typically viewed using the Block Definition Diagram (BDD) as shown in Fig. 3(b).

In addition to specifying the makeup of an assembly, the model may also specify how the products of an assembly are connected in terms of data, power, fluid, etc. This is the information necessary to produce typical engineering products such as system block diagrams and propulsion schematics. In SysML such connectivity is captured and viewed using the Internal Block Diagram (IBD). An example of data (in blue) and power (in red) connectivity for Flight System product in Fig. 3(b) is shown in Fig. 3(c). While Fig. 3(c) is a very simple, contrived example, a more complex and real block diagram of the Carrier of the EHM Lander Mission [3] is shown in Fig. 4. In addition to block diagrams, we are also able to capture propulsion schematics using the same approach as shown in Fig. 5.

Modeling Work Breakdown Structure

Another type of model in the single-source-of-truth is the work breakdown structure (WBS). A WBS is a hierarchical decomposition that is generally used to organize the work and cost associated with designing, building and operating a system. Accordingly, a WBS can be used to organize a MEL as well. As defined by the IMCE work at JPL [1], each element of a WBS is called a *work package*. A WBS is defined by a hierarchy of work packages, where the hierarchy represents the hierarchy of *authorizes* relationships. An authorizes relationship specifies the authority of the higher-level work package and its approval of the lower-level work packages' work. A simple example of a contrived WBS is shown in Fig. 6. A work package is represented in SysML by creating a block that is stereotyped by ≪Work Package≫. An authorize relationship is specified by creating a dependency, stereotyped by ≪authorizes≫, between two work packages.

Another pattern associated with work packages is the *supplies* relationships. One of the things that a work package is responsible for is supplying products. Through this relationship, we can determine which work package is responsible for which product, and help organize a MEL accordingly. An example of supplies relationships based on the products shown in Fig. 3(b) and the work packages in Fig. 6 is shown in Fig. 7. A supplies relationship is specified in SysML by creating a dependency, stereotyped by ≪supplies≫, between a work package and a product. The MEL and Mass Margin Analysis Views section will show how the WBS patterns are used in a MEL.

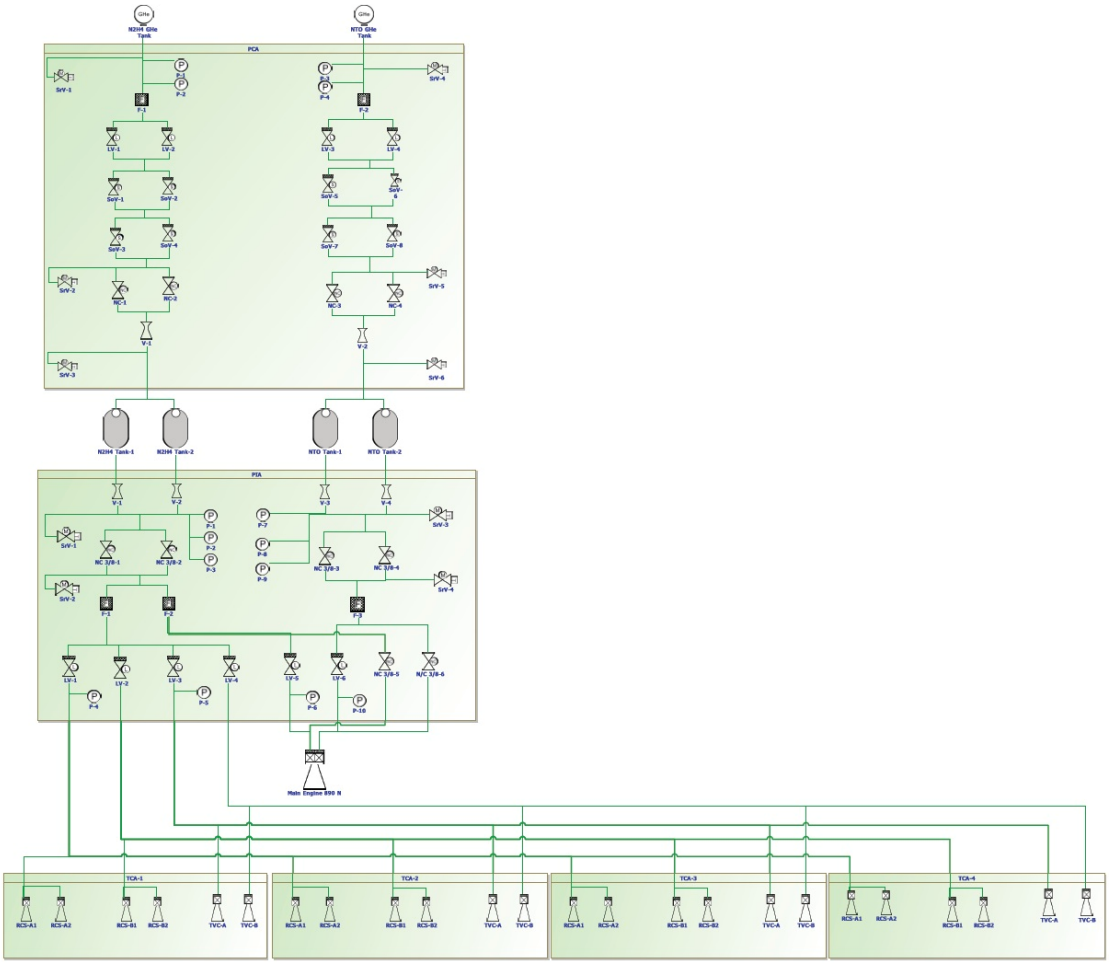


Fig. 5: The propulsion schematic of the Carrier from the Europa Lander Mission Study [3, pg. D-106].

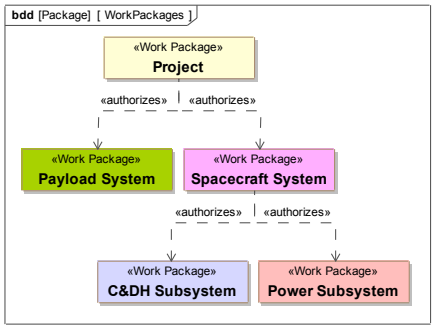


Fig. 6: The modeling pattern in SysML for specifying the Work Breakdown Structure (WBS).

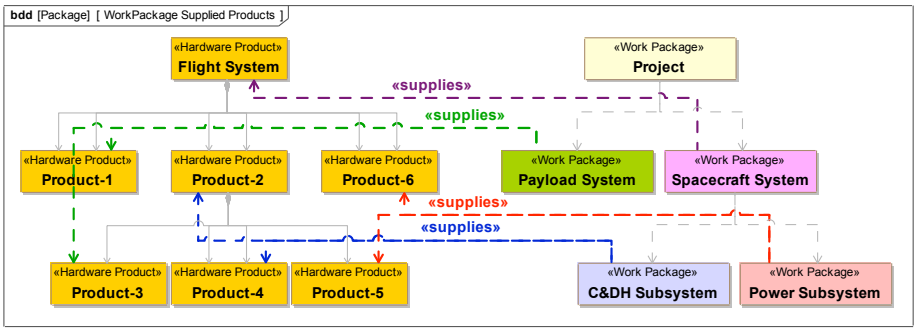


Fig. 7: A SysML representation for the ≪supplies≫ relationships between the work packages and the products.

Table 1: The bill of materials table of the Flight System shown in Fig. 3(b) grouped by the Work Breakdown Structure shown in Fig. 6.

WorkPackages	Products	Mass per Unit			Num of Units	Total Mass		
		CBE (kg)	Cont.	MEV (kg)		CBE (kg)	Cont.	MEV (kg)
Project						106.4	0.28	136.0
Payload System						7.4	0.27	9.4
	Product-1	**2.5**	**0.25**	3.1	2	5.0	0.25	6.3
	Product-3	**1.2**	**0.30**	1.6	2	2.4	0.30	3.1
Spacecraft System						99.0	0.28	126.6
C&DH						41.6	0.25	52.0
	Product-4	**10.4**	**0.25**	13.0	4	41.6	0.25	52.0
Power						57.4	0.30	74.6
	Product-5	**25.5**	**0.30**	33.2	2	51.0	0.30	66.3
	Product-6	**3.2**	**0.30**	4.2	2	6.4	0.30	8.3

Model Correctness

Because all of the models are integrated and formal, we are able to check for correctness of the model. As an example,we can check that the following correctness conditions are satisfied:

- Every hardware product has exactly one mass characterization
- Every mass characterization has exactly one mass contingency, CBE and MEV
- Every product is supplied by exactly one work package
- Every work package is authorized by at most one work package.

These are some of the correctness properties that are automatically checked within the EHM modeling workflow.

MEL AND MASS MARGIN ANALYSIS VIEWS

In this section we describe how the models are used to generate views and how models can be updated through views. Because models are specified in SysML, models can be viewed and edited using SysML views. In addition, JPL has created a framework called DocGen [6] that is capable of creating customized views from models and is capable of updating models through the customized views.

MEL Views and Viewpoints

The MEL is one of our customized views. For the MEL, we have created two views: a *bill-of-materials table* and a *deployment table*. The bill-of-materials table view is a traditional MEL view. An example of this view is shown in Table 1. This view organizes the "leaf-level" products, i.e. the products that are not assemblies, by the work package hierarchy. This view provides the total and per unit mass. It also provides the total mass of the products that are associated with each work package. To an experienced engineer, this view provides a sense of correctness of the system design and also a sense for targets for improvements. Note that in the MEL table views shown in Table 1 and Table 2, the bolded numbers represent the user specified values, while unbolded numbers represent derived or calculated values.

The deployment table view is a not a traditional MEL. However, it provides information that the bill-of-materials table cannot. An example of a deployment table is shown in Table 2. A deployment table shows the assemblies into which the products are deployed. The first column of indented products portrays the product deployment hierarchy. This view provides the total and per unit mass of the assemblies. Unlike the bill-of-materials table, this view allows an engineer to gain insights into the mass characteristics of the assemblies and their makeup. This view also shows which work package supplies each of the products, thus providing a means to check that all products are supplied by some work package. Note that the WorkPackage column could be displayed in a separate view since it does not address a concern relating to mass.

The DocGen capability provides a means to generate and edit the aforementioned views through viewpoint specification and view modeling. In DocGen, a viewpoint is modeled in SysML. Then, once a SysML view conforms to the desired viewpoint and imports the necessary model, the desired view can be automatically generated. As an example, Fig. 8 shows a view model that generates Table 1. Once this view is modeled, the table can be viewed and edited in a Java-based window and published to a Portable Document Format (PDF) or HyperText Markup Language (HTML). Given a view of interest, DocGen provides this capability first by searching (through the dependency stereotyped $\ll$conform$\gg$) the viewpoint that

Table 2: The deployment table of the Flight System shown in Fig. 3(b).

Products	Mass per Unit			Num of Units	Total Mass			WorkPackage
	CBE (kg)	Cont.	MEV (kg)		CBE (kg)	Cont.	MEV (kg)	
Flight System	106.4	0.28	136.0					Flight System
Product-1	**2.5**	**0.25**	3.1	2	5	0.25	6.25	Payload System
Product-2	47.5	0.28	60.7	2	95	0.28	121.42	C&DH
Product-3	**1.2**	**0.30**	1.6	1	1.2	0.30	1.56	Payload System
Product-4	**10.4**	**0.25**	13.0	2	20.8	0.25	26	C&DH
Product-5	**25.5**	**0.30**	33.2	1	25.5	0.30	33.15	Power
Product-6	**3.2**	**0.30**	4.2	2	6.4	0.30	8.32	Power

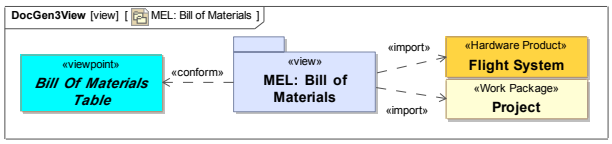


Fig. 8: The view model for the bill of materials table.

the view conforms. DocGen also searches (through the dependency stereotyped ≪import≫) for all of the imported models. Then, DocGen executes the behavior of the viewpoint (specified using SysML Activity) with the imported models as the inputs to the behavior. The result of the behavior execution is a view, such as the bill-of-materials table in Table 1.

Mass Margin Analysis View

Although EHM has yet to implement this view, the same approach used to generate the MEL table views will be used to automatically generate a Mass Margin Analysis and Report view, simply by implementing the behavior of the associated viewpoint. For now, we are using the MEL views for the dry mass, and we compute the mass margin separately based on the dry mass, the expected maximum ΔV, and the launch vehicle's maximum payload mass capacity. As we plan on implementing this capability, we need to develop the modeling patterns for ΔV, the propulsion system type, and launch vehicle payload mass capacity characterizations. The Mass Margin viewpoint must implement the rocket equation to compute the wet mass and the mass margin equation to finally compute the margin. We expect the final view to resemble Table 3.

CONCLUSION

Under the EHM project, a mission concept study, we are taking a MBSE approach to manage the MEL and the mass margin, among other technical margins. We use models as the *single-source-of-truth* under which we have no duplicate information. Because of the model-based nature of the single-source-of-truth, we are able to check for the correctness of the models. Because the views are generated from a single-source-of-truth, we are able to ensure consistency of the information that are duplicated across the views. Based on our experience in this work, we were able to proceed with implementing other technical resource management-related viewpoints and views, including power equipment list (PEL) viewpoints, power/energy margin analysis and report viewpoints, and data margin analysis and report viewpoints..

Due to the efficiency of this approach, EHM was able to create block diagrams/schematics, MEL and mass margins for three different Europa mission concepts in just three months with a high degree of confidence in the consistency, correctness and even completeness. This represent a significant contribution to the state of the practice in development of NASA mission concept study reports, and to systems engineering of complex space systems in general. The result of this effort can be found in the final report describing the EHM studies [3].

ACKNOWLEDGMENTS

This work was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. © 2012 California Institute of Technology. Government sponsorship acknowledged.

Table 3: The mass margin table from the Europa Multiple-Flyby Mission study [3, pg. C-109]

Flyby Mass Margin			
T. Bayer 24 Apr 2012	LAUNCH		
Flyby Model - Final Report Update	Flight System Mass, kg		
	CBE	Cont.*	MEV
Ion & Neutral Mass Spectrometer	24	50%	36
Ice Penetrating Radar	33	50%	50
ShortWave IR Spectrometer	21	50%	31
Topographical Imager	7	50%	11
Payload	85	50%	127
Power	59	21%	72
C&DH	39	30%	51
Telecom	98	29%	126
Structures	529	27%	673
Thermal Control	44	30%	57
Propulsion	175	28%	224
GN&C	68	23%	84
Harness	70	50%	105
Radiation Monitor	8	30%	10
ASRGs (4)	174	45%	252
Spacecraft	1264	31%	1655
Flight System Total Dry	1349	32%	1782
Bipropellant	860		1277
TVC Monopropellant	75		75
ACS Monopropellant	40		40
Pressurant	6		6
Residual and Holdup	24		35
Propellant	1005		1432
Flight System Total Wet	2354		3214
Capability (21-Nov-21 VEEGA)	Atlas V 551:		4494
System Margins			
JPL DWP (Capability - Max Prop - CBE Dry) / (Capability - Max Prop)			48%

REFERENCES

[1] T. J. Bayer, M. B. Bennett, C. L. Delp, D. Dvorak, J. S. Jenkins, and S. Mandutianu. Update - concept of operations for integrated model-centric engineering at JPL. In *Proceedings of Aerospace Conference, 2011 IEEE*, Big Sky, Montana, March 5–12 2011.

[2] T. J. Bayer, S. Chung, B. Cole, B. Cook, F. Dekens, C. Delp, I. Gontijo, K. Lewis, M. Moshir, R. Rasmussen, and D. Wagner. Early formulation model-centric engineering on NASA's Europa mission concept study. In *Proceedings of 22nd Annual International Symposium (IS2012)*, Roma, Italy, July 9–12 2012.

[3] Europa Study Team. Europa study 2012 report. *National Aeronautics and Space Administration*, JPL D-71990, May 1 2012.

[4] S. Friedenthal, A. Moore, and R. Steiner. *A Practical Guide to SysML: The Systems Modeling Language*. Morgan Kaufmann OMG Press, 2009.

[5] ISO/IEC/IEEE. Systems and software engineering — architecture description. *ISO/IEC/IEEE 42010:2011*, December 1 2011.

[6] M. Jackson, C. L. Delp, D. L. Bindschadler, M. A. Sarrel, R. Wollaeger, and D. Lam. Dynamic gate product and artifact generation from system models. In *Proceedings of Aerospace Conference, 2011 IEEE*, Big Sky, Montana, March 5–12 2011.