31st Space Symposium, Technical Track, Colorado Springs, Colorado, United States of America
Presented on April 13-14, 2015

DEVELOPING AND DISTRIBUTING A CUBESAT MODEL-BASED SYSTEMS ENGINEERING (MBSE) REFERENCE MODEL

Primary Author
David Kaslow
Consultant
david.kaslow@gmail.com

Co-authors

Louise Anderson
DigitalGlobe
lweezy@gmail.com

Sharan Asundi
Tuskegee University
Dept of Aerospace Engineering
asundi@mytu.tuskegee.edu

Bradley Ayres
The Aerospace Corporation
bradley.ayres.ctr@afit.edu

Curtis Iwata
The Aerospace Corporation
curtis.iwata@gmail.com

Bungo Shiotani
University of Florida
Space Systems Group
bshiota@ufl.edu

Robert Thompson
Air Force Institute of Technology
robert.thompson@afit.edu

ABSTRACT

Model-Based Systems Engineering (MBSE) is a key practice to advance the systems engineering discipline, and the International Council on Systems Engineering (INCOSE) has established the MBSE Initiative to promote, advance, and institutionalize the practice of MBSE. As part of this effort, the INCOSE Space Systems Working Group (SSWG) Challenge Team has been investigating the applicability of MBSE for designing CubeSats since 2011. The goal of the team is to provide a sufficiently complete CubeSat Reference Model that can be adapted to any CubeSat project.

At the core of MBSE is the development of the system model that helps integrate other discipline-specific engineering models and simulations. The team has been working to create this system model by capturing all aspects of a CubeSat project using the Systems Modeling Language (SysML), which is a graphical modeling language for systems engineering. In the past three phases of the project, the team has created the initial iteration of the reference model, applied it to the Radio Aurora Explorer (RAX) mission, executed simulations of system behaviors, interfaced with commercial simulation tools, and demonstrated how behaviors and constraint equations can be executed to perform operational trade studies.

The current fourth phase has been focused on the next iteration of the CubeSat Reference Model. The CubeSat Domain has been scaled to accommodate the entire project life cycle. Each phase (or stage) of the life cycle is further divided to capture all aspects of the project. For example, the CubeSat Operational Domain is divided into the CubeSat Mission Enterprise, the External Environment, and the External Constraints. Additionally, the model incorporates the concept of operation, which includes launch, early operations, and normal operations. Finally, the team is investigating options to share the model so that it reaches the hands of CubeSat designers.

INTRODUCTION

Traditional Systems Engineering – Documents and Ad-Hoc Models

The engineering practice has been using documents and models as the preferred mode of capturing and communicating complex information. To design a single system, such as a CubeSat, a design team will need to produce many engineering artifacts including requirements and interface documents, presentations of how the system operates, models of system performance, and models of subsystem components and payloads. It is a challenge to maintain consistency across the documents and models because they exist as separate entities with no integration between these different engineering artifacts. Table 1 illustrates the role of technical documents and models in the traditional systems engineering methodology.

Documents serve as an important tool to record, store, and convey information, and although its physical form has evolved with the available technology, it continues to be widely used. Before the modern computer was invented and widely distributed, technical documents were created by typists who transcribed hand-written notes by engineers. Changes were proposed by marking up the hardcopy document and submitted to the control board. These were then reviewed by the committee to assess the impact, and once approved, the changes were incorporated into the technical documentation. Hardcopy change pages were then sent out to the document custodians. This process is largely unchanged despite the introduction of computers, networking, databases, and collaboration software.

Models are useful in providing insight into a problem because it enables exploration within a logical set of constraints, but its scope is limited. Modeling probably started with equations on paper to support requirements generation and analysis, and the tools of the trade evolved from slide rules and calculators to mainframes and supercomputer clusters. The initial space-ground system modeling was confined to the operation of collecting mission data, and the model supported the generation of system requirements and analysis of system performance. The system model was not a model of the physical space and ground systems and subsystems, but instead, it captured the system's desired performance. The space subsystems components and payloads were modeled in support of design and development, but this was separate from the system model. Models were created in an "ad-hoc" or as-needed basis, and these models were not integrated. Furthermore, the models were generated and executed through software codes, and there was little visibility of what was modeled and how it was modeled. The model reviews were based on an engineer's presentations and descriptions.

In the process of designing a system, many documents and models are created, and they are not consistently synchronized and updated due to the burden and seemingly low value. Inconsistencies are easy to manage and rectify in a relatively simple project using processes and policies, but such non-technical solutions do not scale very well when developing systems of sufficient complexity. To overcome this challenge, a paradigm that utilizes a combination of the two entities is emerging, and this is known as Model-Based Systems Engineering.

INCOSE's Model-Based Systems Engineering Efforts

Model-based approaches have been in use by other engineering disciplines including electrical, mechanical, and software, and this paradigm is now being adopted by the systems engineering practice as well. INCOSE defines MBSE as follows:

Model-based systems engineering (MBSE) is the formalized application of modeling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases. [1]

The objective of MBSE is to develop a model of a system that evolves with the project from start to decommission. It is an integration of discipline-specific engineering models and simulations, and it is based on a modeling language and graphical-based modeling tools where requirements and interfaces are not contained in isolated engineering artifacts as done previously but are now integrated within a single system model. This allows for a full exploration of the model, and changes are automatically propagated and reflected in every view of the system. The goal is to generate documentation from the model and not vice versa, and this is only if it necessary in the first place. Moreover, the physical system should be built using the model and not to the documentation.

Figure 1 shows the evolution of the International Council on Systems Engineering's (INCOSE) MBSE initiative and the genesis of the Space Systems Working Group (SSWG) MBSE Challenge Project. In 2007, INCOSE established its Systems Engineering Vision 2020 [1]. The vision included demonstrating the applicability of MBSE paired with the Systems Modeling Language (SysML) to several engineering disciplines. The MBSE Initiative started at the 2007 International Workshop [2]. INCOSE established the MBSE Roadmap shown in Figure 2 and a set of MBSE Challenge Teams including one for space systems modeling [3]. The Roadmap defines the high-level, long-term vision for the maturation and acceptance of MBSE across academia and industry.

SysML was developed jointly by INCOSE and the Object Modeling Group (OMG) to support MBSE [4]. Figure 3 contains an overview of the SysML modeling elements. There are modeling elements for requirements, structure, behavior, packages, and parametrics. Structure diagrams consist of block definition diagrams and internal block diagrams. Behaviors describe how a block deals with inputs and outputs and changes to its internal state. Behavior diagrams describe what the system must do to meet requirements. Behavior diagrams consist of Activity, State Machine, Sequence, and Use Case diagrams. Parametric diagrams represent the mathematical formulations used by the formal models and simulations. SysML is used to specify, analyze, design, optimize, and verify systems including hardware, software, information, personnel, procedures, processes, and facilities. But SysML is just a language and not a methodology or tool.

Figure 3 also illustrates that MBSE consists of not only a modeling language but also requires an engineering methodology and the incorporation of other modeling tools. INCOSE published *Survey of Model-Based Systems Engineering (MBSE) Methodologies* which states that “A MBSE methodology is a collection of related processes, methods, and tools used to support the discipline of systems engineering in a ‘model-based’ or ‘model-driven’ context” [5]. INCOSE also established a post-survey web site that listed additional MBSE methodologies [6].

SSWG MBSE CHALLENGE PROJECT EARLY PHASES

The SSWG MBSE Challenge Team has completed four phases (Phase 0 to 3), and Figure 4 shows an overview of these phases. The initial (Phase 0) effort focused on the modeling of a hypothetical FireSat space system. FireSat is a low-Earth orbit spacecraft for detecting, identifying, and monitoring forest fires. This space system is used as an example in the widely used and accepted Space Mission Analysis and Design (SMAD) textbook [7].

The results from the initial effort were reported first in December 2007 then in a series of INCOSE workshops and symposiums, and INCOSE INSIGHT articles. The results demonstrated that a space system could be modeled in SysML. Modeling FireSat demonstrated the application of SysML but the hypothetical nature of FireSat precluded anyone from actually building a real space system from the model. Therefore the practical use of the model could not be demonstrated or verified.

The SSWG MBSE Challenge Project then initiated the MBSE CubeSat Project in April 2011 to demonstrate the application of MBSE to a realistic mission in the space systems domain. A CubeSat is a type of miniature spacecraft with a form factor based on the standardized unit cube, which is 10-centimeters on a side, and it weighs approximately one kilogram per cube. CubeSats typically consist of one to three units with some up to six units.

Phase 1 consisted of developing a SysML reference model of a CubeSat and applying it to the Radio Aurora Explorer (RAX) mission [8]. RAX is a three-unit CubeSat developed jointly by SRI International and the Michigan Exploration Laboratory at the University of Michigan [9]. The RAX mission is to study the formation of magnetic field-aligned electron density irregularities in the Earth’s ionosphere, which are known to disrupt tracking and communication between Earth and orbiting satellites.

During each pass over a ground-based radar station, RAX receives and processes the scattered radar signal and then downlinks the payload and telemetry data to a ground station. The modeling of RAX was to prove the applicability of MBSE for modeling operational space missions. It was not intended to be an accurate model of the RAX satellite. The phase 1 RAX SysML model defined the logical and physical architecture of the flight and ground systems.

Table 1 - Traditional Document-Based and Ad-Hoc Modeling Systems Engineering Methods

	Document-Based	Ad-Hoc Modeling
Intent	Systems – Subsystems – Components Operations concepts, requirements, interfaces, verification, …	In support of generating and analyzing requirements Mission data collection Space subsystems and payloads
Evolution	In the beginning hardcopy only Computers, word processors, and databases - transition to softcopy	Equations, slide rules, calculators Mainframe computers Desktop computers
Shortcomings	Still mostly hardcopy produced for reviews and design specifications	Code running on a computer Little visibility to the what and how of the models Hardcopy PowerPoint and Visio diagrams created as need for reviews

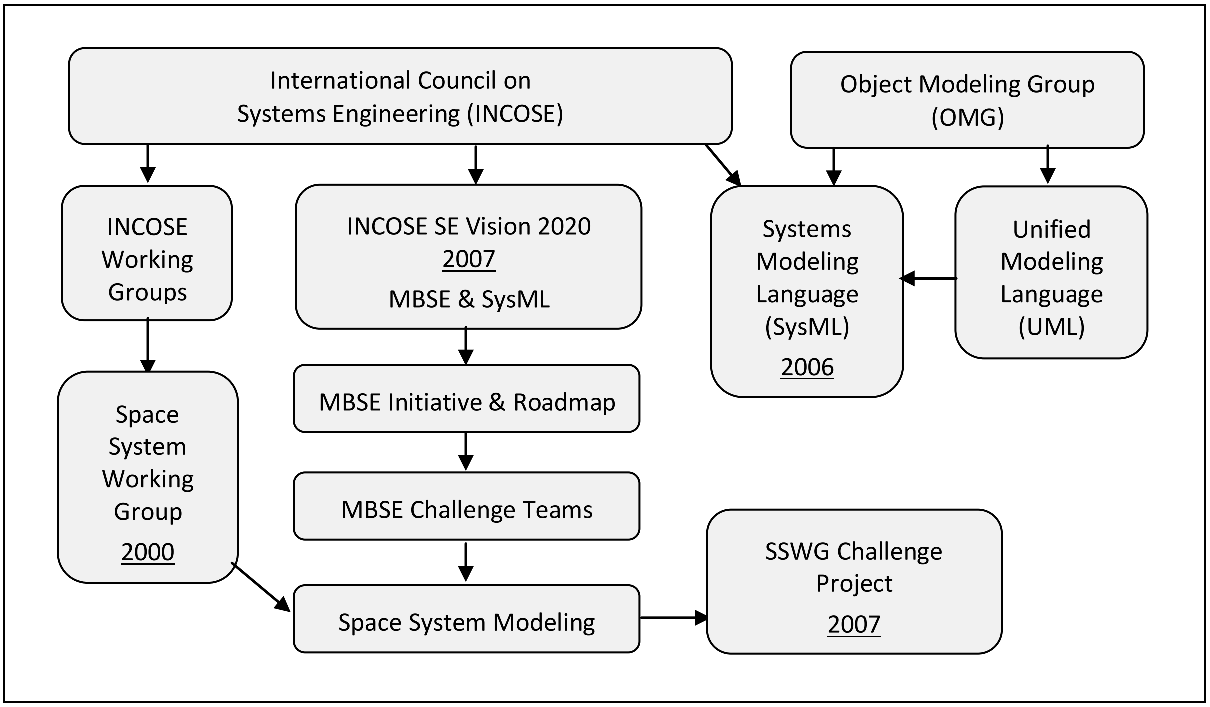

Figure 1 - Context of the INCOSE MBSE Initiative and SSWG Challenge Project

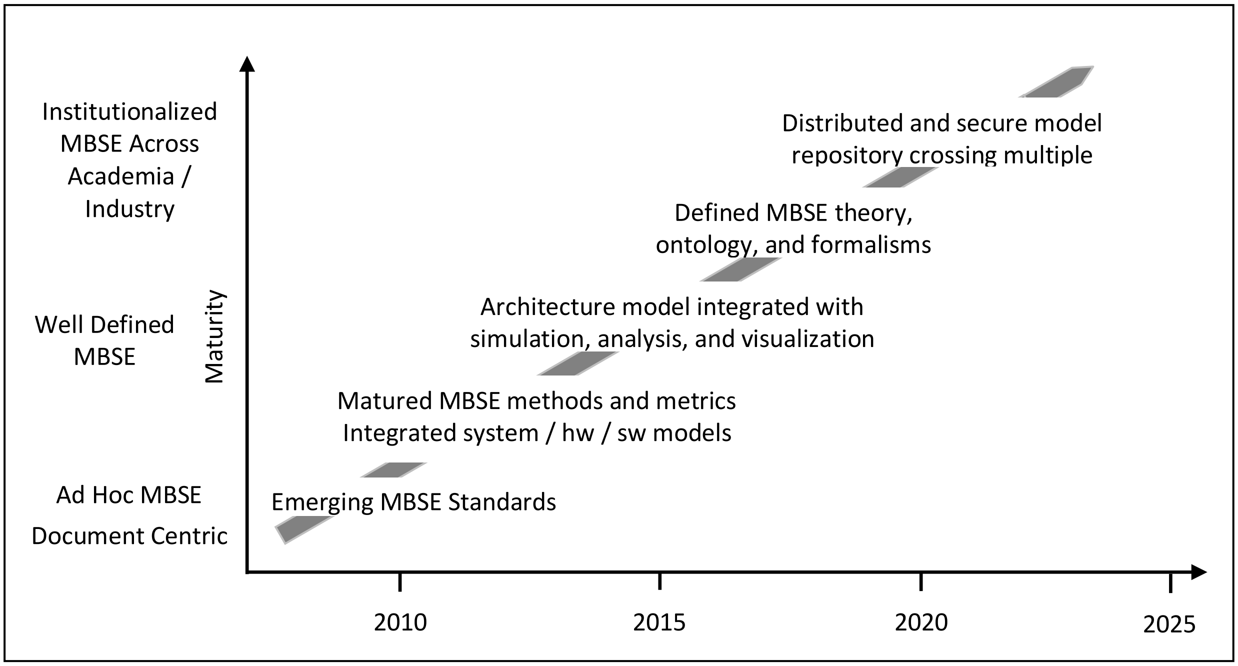

Figure 2 - Model-Based Systems Engineering Roadmap. Adapted from [3]

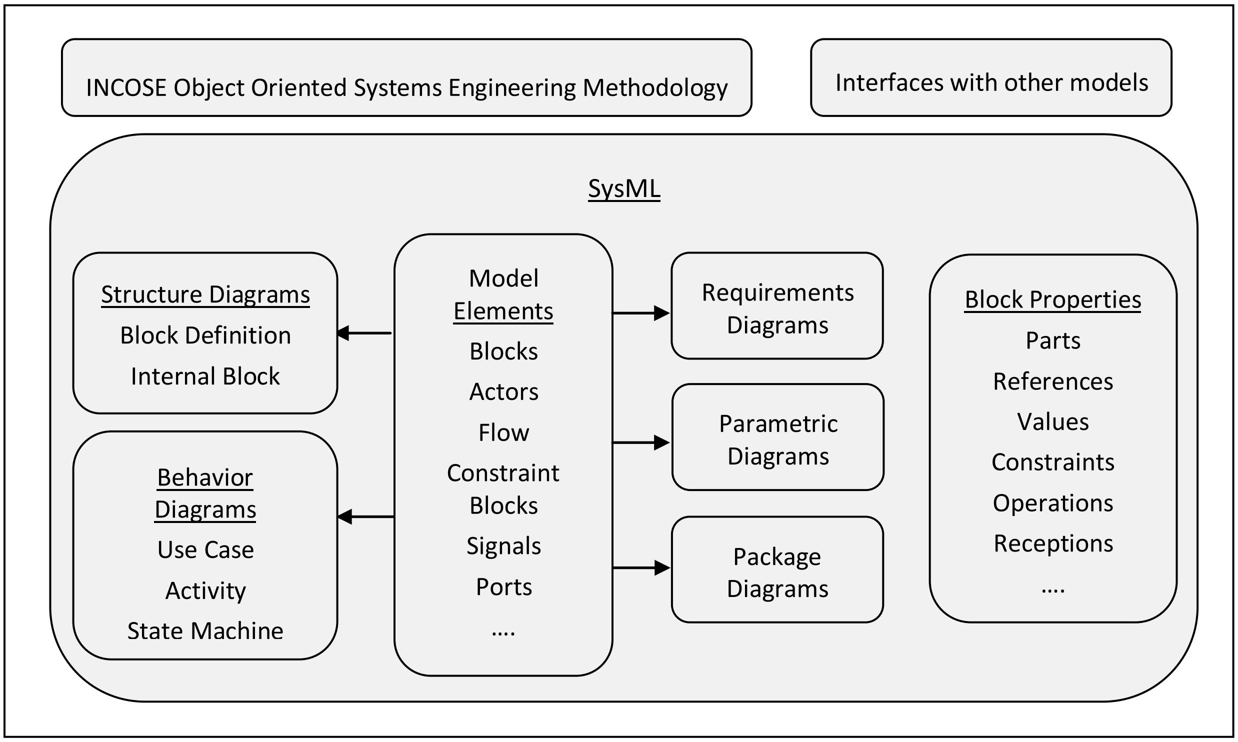

Figure 3 - Model-Based Systems Engineering Components

Phase 2 focused on expanding the RAX CubeSat model to include behaviors [10]. Mission activities and states were modeled such as: opportunities to collect mission data, download data, and collect solar energy. A power model accounted for solar energy collection and subsystem power consumption. A communication model accounted for data download rate, available power, and signal to noise ratio taking into account gains and losses due to communication components, atmosphere, and length of propagation path.

Phase 3 consisted of two efforts. The first was to develop the beginnings of an enterprise-level model for a generic CubeSat [11]. The second effort was to a develop a new model with the capability to time-step through a scenario and to capture the energy collection and usage processes as well as data collection, storage, and downlink [12]. Two trade studies were demonstrated: 1) On-board energy level as a function of solar panel area and maximum battery capacity and 2) quantity of data downlinked as a function of orbital altitude and ground station network.

CURRENT PHASE

Development and Description of the CubeSat Reference Model

The current phase of the project is focused on developing the CubeSat Reference Model, and Figure 5 illustrates the SSWG's development approach. The SSWG team consists of students, professors, engineers, and software developers from across the industry including NASA centers, aerospace companies, and modeling and simulation tool providers. Team members are continually recruited from symposiums and workshops. There is interaction with the INCOSE Object-Oriented Systems Engineering Method (OOSEM) and Model-Based Concept Design (MBCD) working groups.

The architecture of the CubeSat Reference Model is founded on MBSE and OOSEM as presented in the *INCOSE Systems Engineering Handbook* [13] and *A Practical Guide To SysML* [14]. The specific methodology used to create the architecture is not critically important to the development of the reference model. It only matters that the model provides the foundation for a user to create a physical architecture. The decomposition of the system into space and ground subsystems is consistent with the *NASA System Engineering Handbook* [15] and *Space Mission Engineering – The New SMAD* [7]. The Cal Poly CubeSat Design Specification provides the foundation for specifying a CubeSat's physical, mechanical, electrical, testing, and operational requirements [16].

The CubeSat Reference Model starts with an identification of potential stakeholders. A stakeholder is any entity that has an interest in the system including sponsor, end user, procurer, supplier, and regulatory agencies. The each stakeholder's needs, objectives, constraints, and measures of effectiveness are incorporated in the reference model. Constraints are those items fixed and not subject to trades such as mission budget and schedule.

The reference model supports all life cycle stages employed by the organization utilizing the model. For example, for an academic CubeSat project, the likely stages would be concept, development, production, operational, and retirement. The model supports all phases of operations: launch, early operation, and normal operations.

The CubeSat Reference Model is built using No Magic Inc. products, including MagicDraw and Cameo Systems Modeler, as the modeling tool set. Importing to, and exporting from, other vendors modeling tools may result in compatibility issues between different elements of the model. Some additional effort may be required in order to resolve these issues. However, vendors are increasingly developing functionality that allows the importing of models developed by other vendors' tool sets. Additionally, exporting to a common file format, like the XML Metadata Interchange (XMI), may reduce these interface issues.

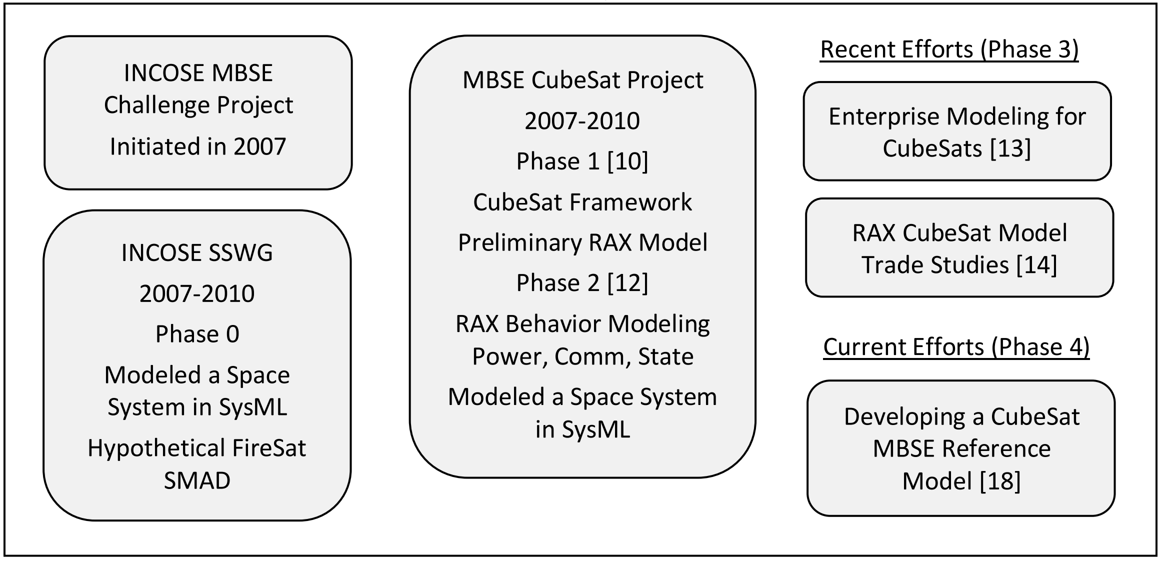

Figure 4 - SSWG MBSE Challenge Project Phases

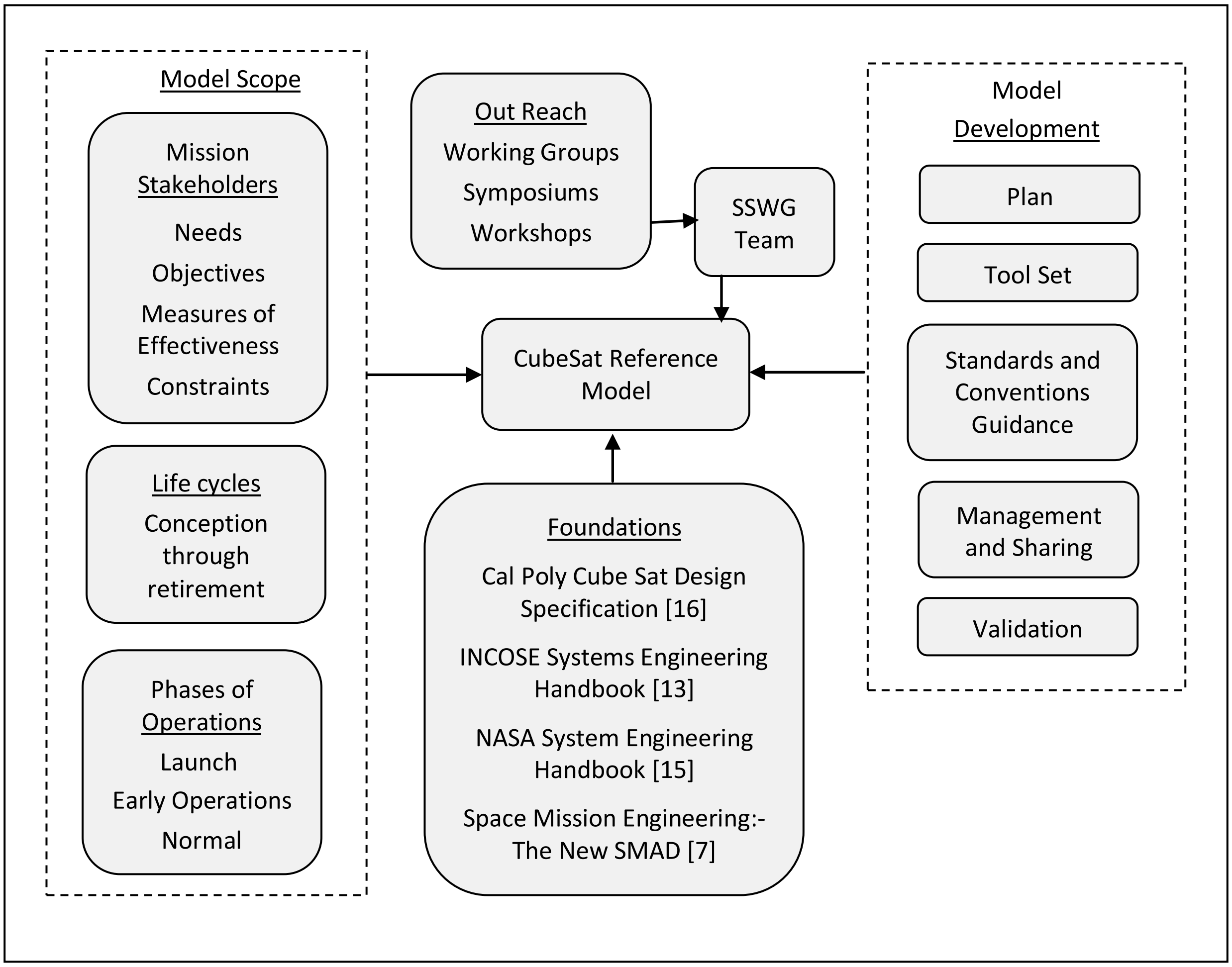

Figure 5 - Development of the CubeSat Reference Model

A copy of the model is made available to members of the SSWG team for review and comments. The model is updated according to the discussions that happen during the weekly teleconference. Even though team members are participating on their own time, each team member contributes in accordance with their organization's proprietary information policies. If a team member provides model components that were developed using the organization's modeling tools, the mechanism and approval process for providing the components should be well documented. This could still be an issue if the organization changes its position with the worst case being the model becomes proprietary to the organization.

Distribution of the CubeSat Reference Model

The CubeSat Reference Model is approaching a sufficient level of maturity to begin sharing with the rest of the MBSE community. There have also been requests to use the model for educational purposes and to evaluate it for actual CubeSat projects. In response, the SSWG team is developing a model distribution process to share to external entities, and this is illustrated in Figure 6. In addition to the model file, the distribution will include video recordings that explain and walk-through of the model and its components.

The distribution will be with the expectation that feedback will be provided to the SSWG team so that the reference model can be improved. The feedback mechanism will follow the guidelines used for internal feedback but there will be an increased level of scrutiny with respect to organization intellectual property and release process.

The minimum level of model structure to be incorporated before the reference model will distributed outside the SSWG includes:

- Representative stakeholder needs, objectives, measures of effectiveness, and constraints
- Properties and parametrics at subsystem level for power and weight
- Data and information flow for plans, schedules, commands, telemetry, and mission data

Since the reference model is being developed by the SSWG team effort, there is an obligation to protect the investment of time and knowledge of each team member. There also needs to be a licensing environment that is conducive to a user organization supporting the development of and use of the model.

There will be a license that allows for non-commercial use and prohibits incorporating the model or model features into a commercial product. Whether there will be a prohibition against using the model in a commercial enterprise has not been determined.

For the needs of the team, Creative Commons Organizations licenses provide sufficient protection and flexibility. Creative Commons is a nonprofit organization that enables the sharing and use of creativity and knowledge through free legal tools [17]. Since the licenses work in concert with copyrights, the CubeSat Reference Model will need a copyright. If INCOSE designates the model as INCOSE product, then it likely will be an INCOSE SSWG copyright.

Four options are being considered to make the model available outside the SSWG, which are the following: MagicDraw ZIP file, XMI file, through a hosting of JPL's OpenMBEE residing on Amazon Web Services, and Teamwork Server hosted on by No Magic. These methods are dependent on the user's capability to access the model through the organization's firewall or to import the model with a USB drive. The MagicDraw ZIP file and the Teamwork Server approaches are probably the easiest and least prone to translation error because the reference model is being developed using MagicDraw. The use of XMI will be evaluated by exporting the model to Sparx System's Enterprise Architect and IBM's Rational Rhapsody.

OpenMBEE is an open-sourced, model-based infrastructure that is being developed by JPL, and it is capable of storing all information in one location. There are access points for system modelers using any number of modeling tools and systems engineers that do not have the time to learn the various modeling tools but need a repository for their design information. OpenMBEE supports the interaction of modeling tools through the reading, processing, and writing of information. A model can be exported from MagicDraw to the OpenMBEE database, edited in the database, and imported back into MagicDraw. OpenMBEE also provides access control as well as configuration management and versioning.

Development of a Mission Specific Model

Figure 7 illustrates the approach to developing a mission specific CubeSat model from the CubeSat Reference Model. The current state of development of the CubeSat Reference Model is shown in Figure 8 through Figure 14 and described in [18]. This will be the starting point for a mission specific CubeSat team.

Figure 8 shows the CubeSat Domains or the scope of the reference model. Providing a fully-formed CubeSat Reference Model could be overwhelming for a team that is early in the system life cycle and unfamiliar with MBSE and SysML. Having models for each life-cycle phase should make it easier for a team to modify, add, and remove SysML elements when constructing their mission specific model.

Figure 9 shows a representative set of stakeholders, which can be modified depending on the project. Identification of the stakeholders is important and should be done at the start of a project. Particular attention must be paid to the regulatory agencies. CubeSat projects are pursued internationally, but the licenses and regulations that cover its activities are administered at the national level. Consequently, the rules, the process to gain specific permissions, and the responsible regulatory bodies varies between countries. For example, in the United States, the Federal Communications Commission (FCC) regulates the frequencies, the Orbital Debris Assessment Report contains guidelines for limiting orbital debris, and the National Oceanic and Atmospheric Administration (NOAA) regulates remote sensing. Since most CubeSats use amateur frequency bands, the desired amateur frequency must first be coordinated through the International Telecommunication Union (ITU) and then the International Amateur Radio Union. American CubeSat projects must also adhere to the International Traffic in Arms Regulations and Export Administration Regulations. The timelines for requesting and receiving approval must be well understood at project start.

Figure 10 to Figure 12 show the scope of the reference model with each stage of the life cycle further divided into lower elements. Specifically, Figure 10 shows the CubeSat mission enterprise, external environment, and external constraints. Figure 11 and Figure 12 shows the details of each element of Figure 10. The details include shipping, launch, and communications support and services.

The space and ground systems are represented by logical and physical architectures. The logical architecture consists of abstractions of components that implement the system, and it also defines the interactions between these components that are needed to accomplish the system-level actions and operations. The logical components are then allocated to the physical components that are implemented as hardware, software, data, procedures, and operator actions. The physical architecture is then defined in terms of these physical components and their interactions [14]. Figure 13 and Figure 14 illustrate the logical space and ground system reference models.

The CubeSat Reference Model will be a generic logical space-ground system architecture, suitable for use by a wide variety of CubeSat developments. The CubeSat development team can use it to create the mission specific physical architecture. This starts with the CubeSat development team identifying all the stakeholders and their needs, objectives, measures of effectiveness, and constraints and populating them into the structure of the reference model. Figure 15 illustrates that a mission objective refines a mission need and that a mission use case refines a mission objective. Then the mission objectives trace to mission requirements which trace to system requirements which trace to subsystem requirements. Once the stakeholder needs, objectives, measures of performance, and constraints have been incorporated into the logical model as requirements, the CubeSat development team can create the mission specific physical architecture that meets the specific system requirements.

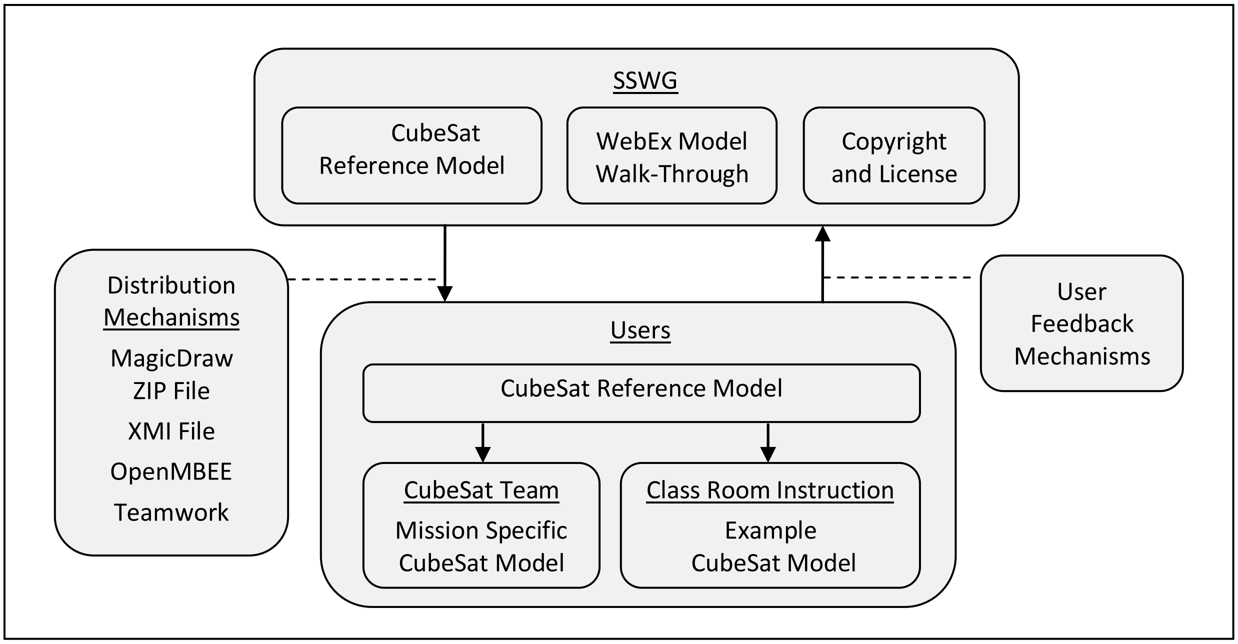

Figure 6 - Distribution and Use of the CubeSat Reference Model

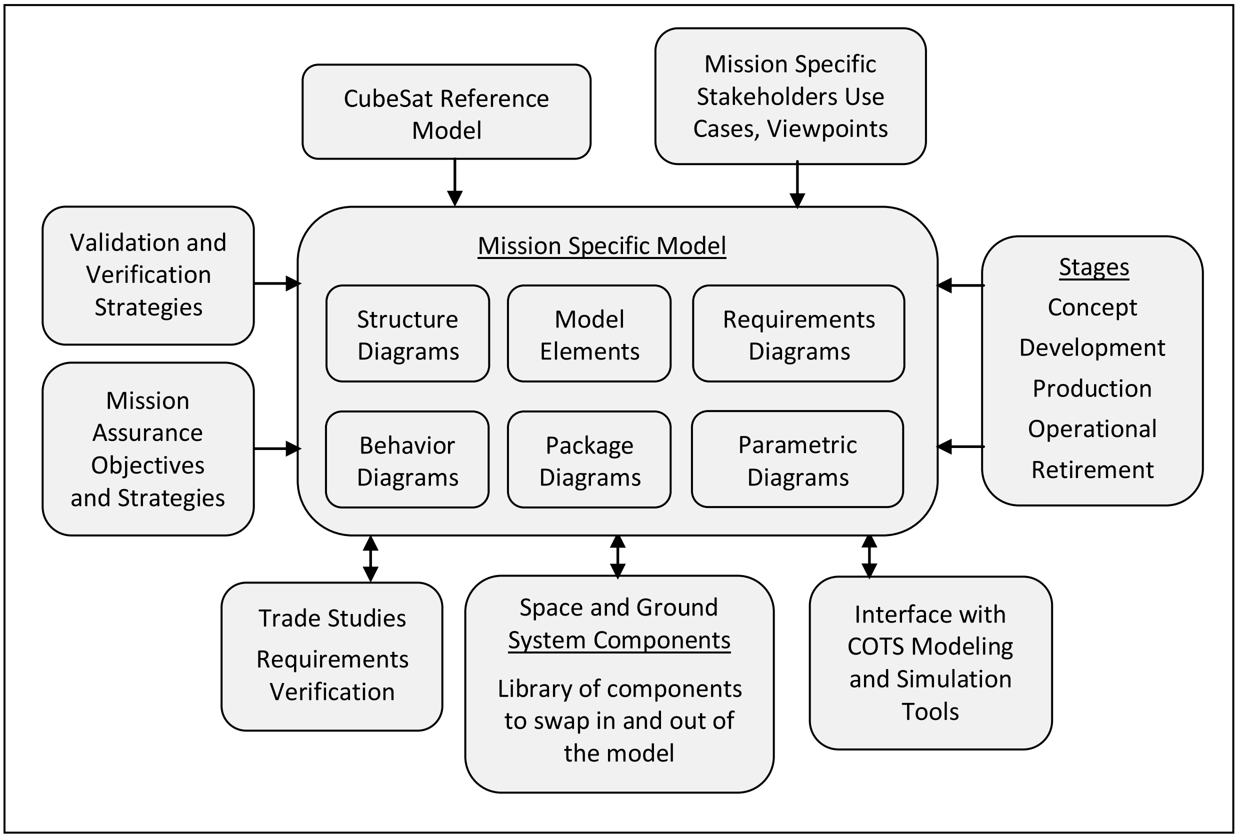

Figure 7 - Development of a Mission Specific CubeSat Model

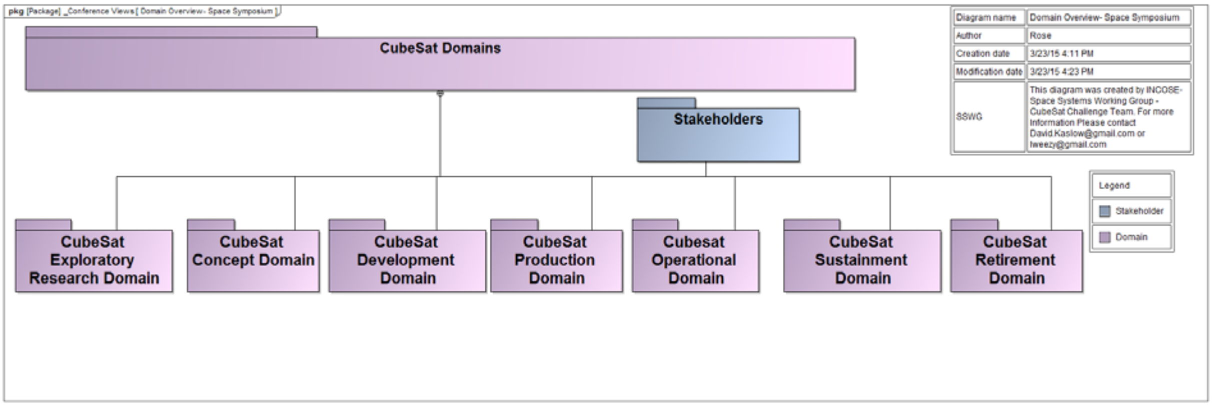

Figure 8 - CubeSat Domains

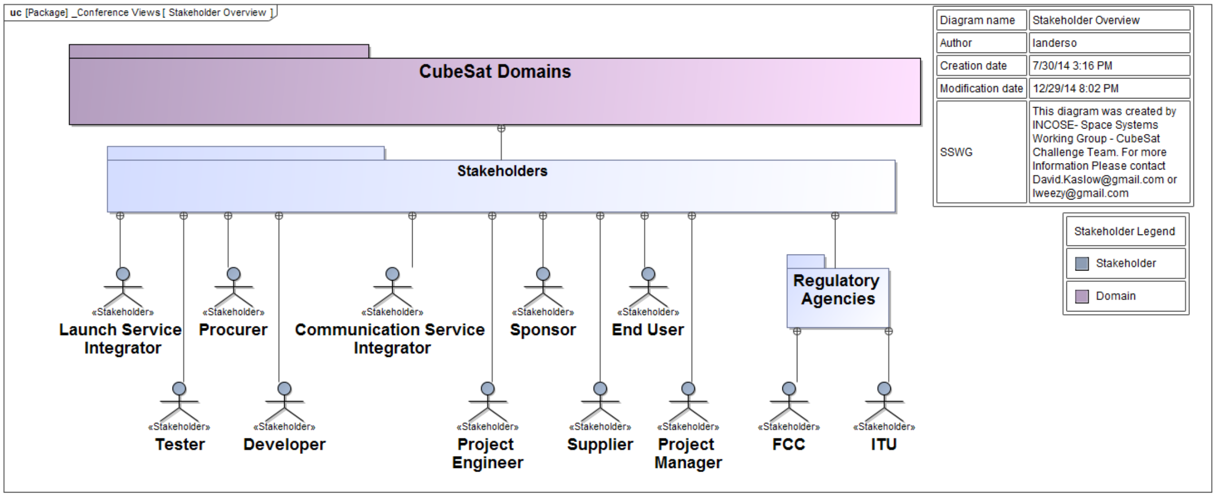

Figure 9 - CubeSat Stakeholders

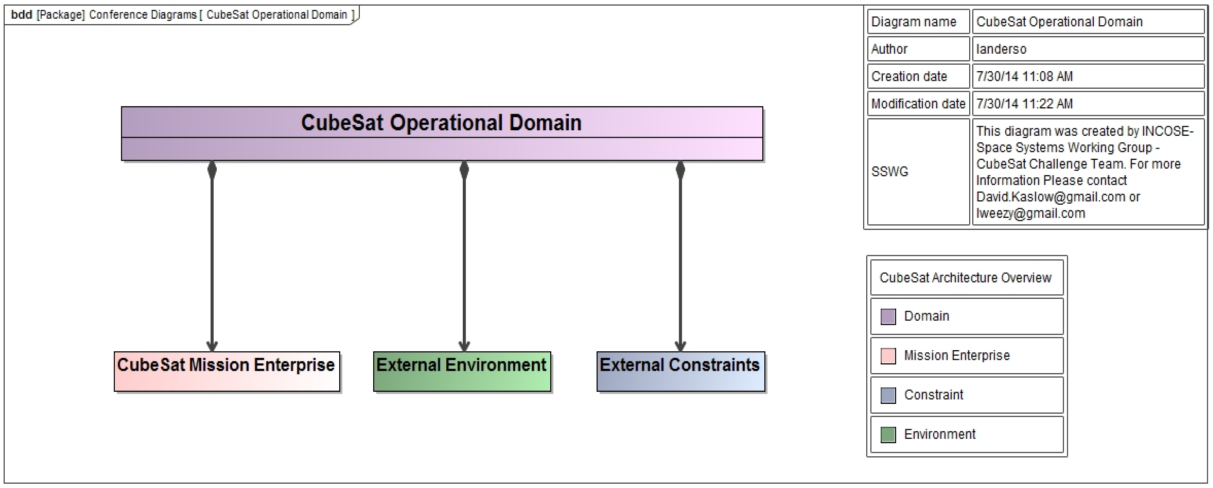

Figure 10 - CubeSat Operational Domain

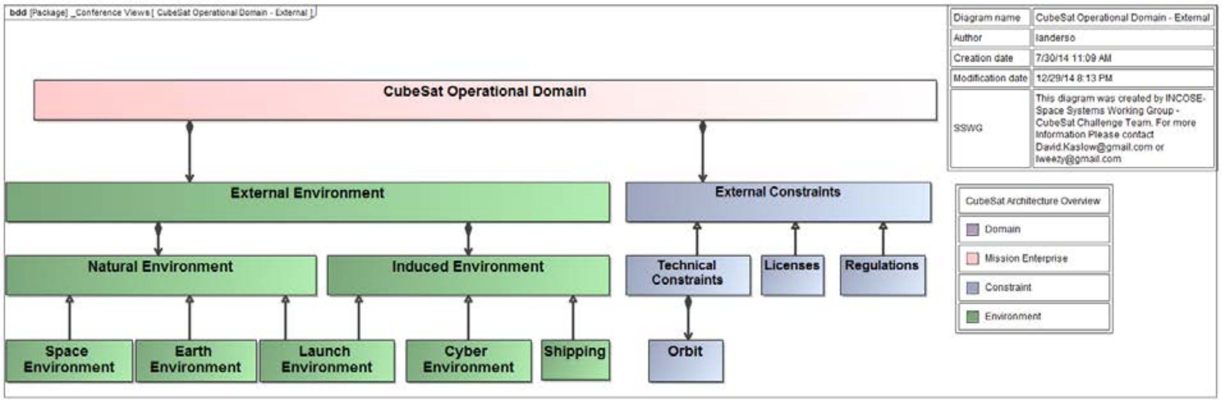

Figure 11 - External Environment and External Constraints

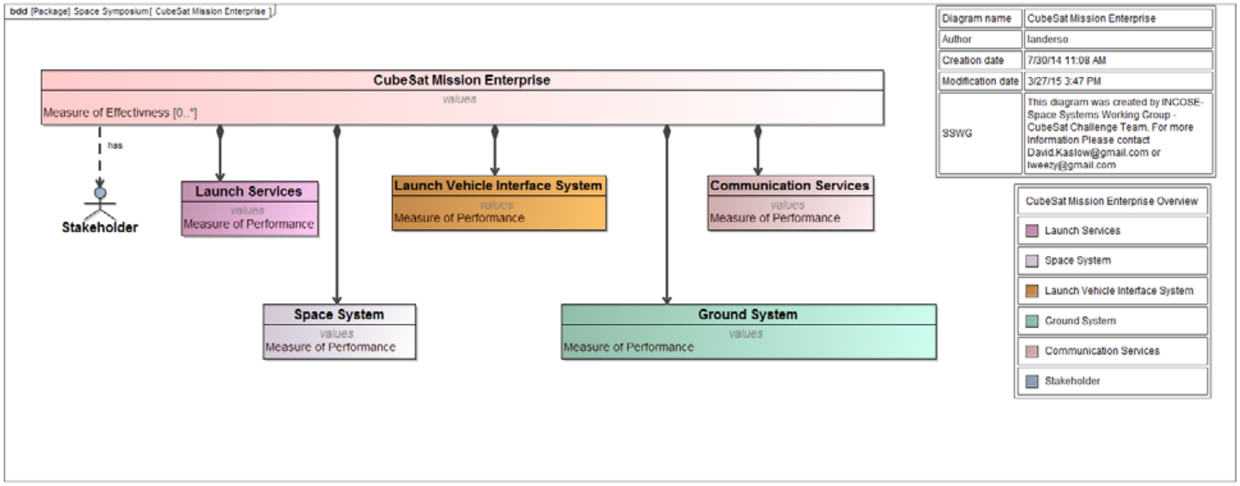

Figure 12 - CubeSat Mission Enterprise

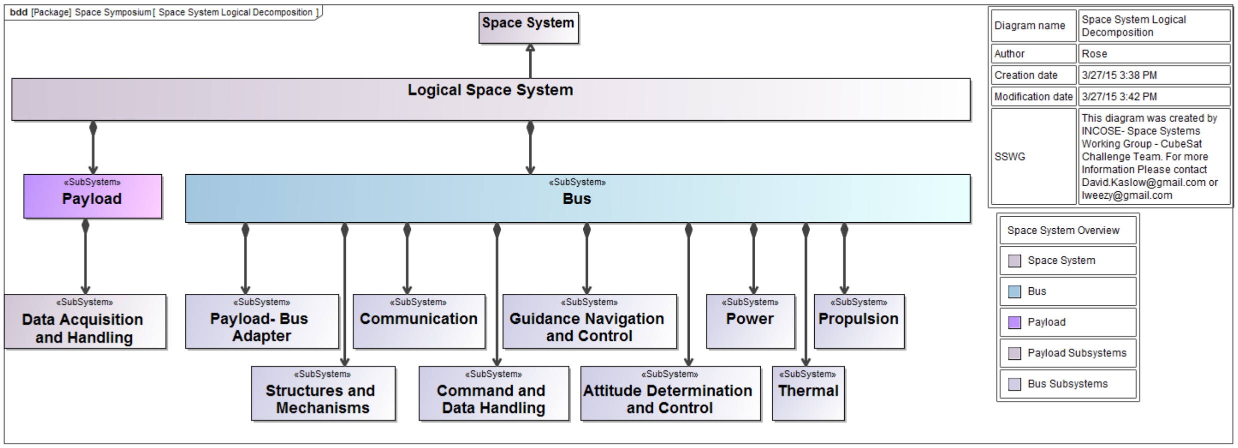

Figure 13 - Logical Space System Model

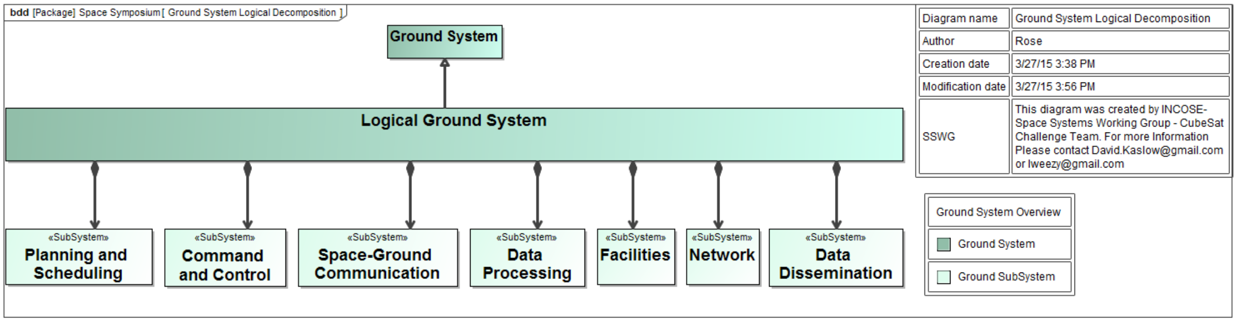

Diagram name	Ground System Logical Decomposition
Author	Rose
Creation date	3/27/15 3:38 PM
Modification date	3/27/15 3:56 PM
SSWG	This diagram was created by INCOSE- Space Systems Working Group - CubeSat Challenge Team. For more Information Please contact David.Kaslow@gmail.com or lweezy@gmail.com

Figure 14 - Logical Ground System Model

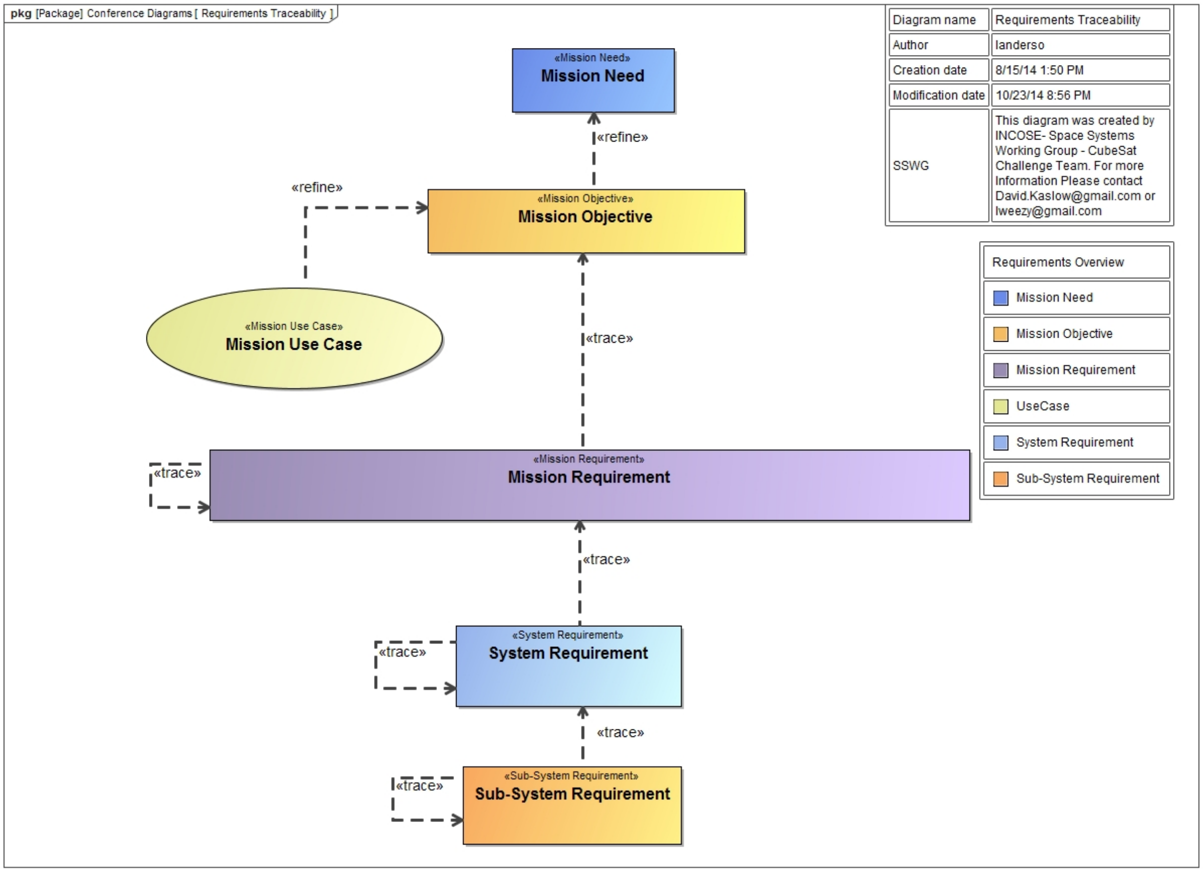

Diagram name	Requirements Traceability
Author	landerso
Creation date	8/15/14 1:50 PM
Modification date	10/23/14 8:56 PM
SSWG	This diagram was created by INCOSE- Space Systems Working Group - CubeSat Challenge Team. For more Information Please contact David.Kaslow@gmail.com or lweezy@gmail.com

Figure 15 - Flowdown of Stakeholder Mission Needs

NEXT STEPS

The effort to date has been focused on establishing standard nomenclature and definitions; incorporating the stakeholders and their needs, objectives, and measures of effectiveness; and defining the generic CubeSat architecture down to the logical subsystems. The next step is to determine the level of model definition at each of the life cycle stages and to create reference models for the concept and development phases.

CONCLUSION

After several phases of learning and applying MBSE to the CubeSat design process, the SSWG Challenge Team is now focused on developing a SysML model that will serve as a framework for future CubeSat developers. MBSE holds the promise of reducing the burden of systems engineering tasks, which is beneficial to small CubeSat teams, and a properly designed reference model can serve as a checklist to these teams and promote uniformity and consistency across future CubeSat models. The work on the CubeSat Reference Model is ongoing, and this product has been matured to the point where the Challenge Team is comfortable sharing with the rest of the MBSE community. Several avenues of distribution along with licensing and copyright options are being considered, and the team hopes to release the models publicly in the near future.

REFERENCES

[1] Systems Engineering Vision 2020, INCOSE-TP-2004-004-02, ver. 2.03,September 2007. [Online]. Available: http://www.incose.org/ProductsPubs/pdf/SEVision2020_20071003_v2_03.pdf

[2] International Council on Systems Engineering (INCOSE), “MBSE Initiative,” January 2007. [Online] Available: https://connect.incose.org/tb/MnT/mbseworkshop/

[3] MBSE Roadmap. MBSE Wiki, INCOSE MBSE IW 2012. MBSE Wiki. [Online} Available: http://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:mbse_iw_2012-introduction-2012-01-21-friedenthal-c.pptx

[4] Object Management Group (OMG), OMG Website. [Online]. Available: http://www.omgsysml.org/

[5] *Survey of Model-Based Systems Engineering (MBSE) Methodologies.* INCOSE-TD-2007-003-01, Ver B. 10 June 2008. [Online]. Available https://www.incose.org/ProductsPubs/pdf/techdata/MTTC/MBSE_Methodology_Survey_2008-0610_RevB-JAE2.pdf

[6] *Additional Methodologies Identified as Gaps since 2008 INCOSE Survey*. MBSE Wiki, Metrics and Methodologies [Online]. Available: http://www.omgwiki.org/MBSE/doku.php?id=mbse:methodology

[7] J.R. Wertz, D. Everett, and J. Puschell, Eds., *Space Mission Engineering: The New SMAD*, (Space Technology. Library, Volume 28), Hawthorne, CA, Microcosm Press, 2011.

[8] S. Spangelo, D. Kaslow, C. Delp, B. Cole, L. Anderson, E. Fosse, B. Gilbert, L. Hartman, T. Kahn, and J. Cutler, “Applying Model Based Systems Engineering (MBSE) to a Standard CubeSat,” in *Proceedings of IEEE Aerospace Conference*, Big Sky, MT, March 2012.

[9] J. Springmann, B. Kempke, J. Cutler, and H. Bahcivan, “Initial Flight Results of the RAX-2 Satellite,” in *Proceedings of the 26th Annual Small Satellite Conference*, Logan, UT, August 2012.

[10] S. Spangelo, L. Anderson, E. Fosse, L Cheng, R. Yntema, M. Bajaj, C. Delp, B. Cole, G. Soremekun, D. Kaslow, and J. Cutler, “Model Based Systems Engineering (MBSE) Applied to Radio Explorer (RAX) CubeSat Mission Operational Scenarios,” *Proceedings of IEEE Aerospace Conference*, Big Sky, MT, March 2013.

[11] L. Anderson, B. Cole, R. Yntema, M. Bajaj, S. Spangelo, D. Kaslow, C. Lowe, E. Sudano, M. Boghosian, R. Reil, S. Asundi, and S. Friedenthal, “Enterprise Modeling for CubeSats,” *Proceedings of IEEE Aerospace Conference*, Big Sky, MT, March 2014.

[12] D. Kaslow, G. Soremekun, H. Kim, S. Spangelo, "Integrated Model-Based Systems Engineering (MBSE) Applied to the Simulation of a CubeSat Mission", *Proceedings of IEEE Aerospace Conference*, Big Sky, MT, March 2014.

[13] *INCOSE Systems Engineering Handbook,* v. 3.2.2, October 2011, INCOSE-TP-2003-002-03.2.2.

[14] S. Friedenthal, A. Moore, R. Steiner, *A Practical Guide to SysML*, 2nd ed., Elsevier, Waltham, MA, 2012.

[15] *NASA Systems Engineering Handbook,* rev. 1, December 2007, NASA/SP-2007-6105 Rev1.

[16] *CubeSat Design Specification*, rev. 13, The CubeSat Program, Cal Poly SLO, February 2014

[17] Creative Commons Website. [Online]. Available: http://creativecommons.org/about

[18] D. Kaslow, L. Anderson, S. Asundi, B. Ayres, C. Iwata, B. Shiotani, R. Thompson, "Developing a CubeSat Model-Based System Engineering (MBSE) Reference Model – Interim Status", *Proceedings of IEEE Aerospace Conference,* Big Sky, MT, March 2015.