Model Based Systems Engineering (MBSE) Applied to Radio Aurora Explorer (RAX) CubeSat Mission Operational Scenarios

Sara C. Spangelo
James Cutler
University of Michigan
1320 Beal Street
Ann Arbor, MI 48104
saracs@umich.edu
jwcutler@umich.edu

Louise Anderson
Elyse Fosse
Leo Cheng
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91109
louise.anderson@jpl.nasa.gov
elyse.fosse@jpl.nasa.gov
leo.y.cheng@jpl.nasa.gov

Rose Yntema
Manas Bajaj
InterCAX
75 Fifth Street NW, Suite 312
Atlanta, GA 30308
rose.yntema@intercax.com
manas.bajaj@intercax.com

Chris Delp
Bjorn Cole
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91109
chris.delp@jpl.nasa.gov
bjorn.cole@jpl.nasa.gov

Grant Soremekum
Phoenix Integration
1715 Pratt Drive, Suite 2000
Blacksburg, VA 24060
grant@phoenix-int.com

David Kaslow
Analytical Graphics
220 Valley Creek Blvd.
Exton, PA 19341
dkaslow@agi.com

Abstract—**Small satellites are more highly resource-constrained by mass, power, volume, delivery timelines, and cost relative to their larger counterparts. Small satellites are operationally challenging because subsystem functions are coupled and constrained by the limited available commodities (e.g. data, energy, and time). Furthermore, additional operational complexities arise because small satellite components are physically integrated, which may yield thermal or radio frequency interference.**

In this paper, we extend our initial Model Based Systems Engineering (MBSE) framework developed for a small satellite mission by demonstrating the ability to model different behaviors and scenarios.

We integrate several simulation tools to execute SysML-based behavior models, including subsystem functions and internal states of the spacecraft. We demonstrate utility of this approach to drive the system design process. We demonstrate the applicability of the simulation environment to represent realistic satellite operational scenarios, which include the energy gathering and the data acquisition and downloading to ground stations.

The integrated modeling environment enables users to extract feasibility, performance, and robustness metrics and enables visualization of both the physical (e.g. position, attitude) and functional states (e.g. operating points of various subsystems) of the satellite for representative mission scenarios.

The modeling approach presented in this paper offers satellite designers and operators the opportunity to assess the feasibility of vehicle and network parameters, as well as the feasibility of operational schedules. This will enable future missions to benefit from using these models throughout the full design, test, and fly cycle. In particular, vehicle and network parameters and schedules can be verified prior to being implemented, during mission operations, and can also be updated in near real-time with operational performance feedback.

978-1-4673-1813-6/13/$31.00 ©2013 IEEE.
[1] IEEEAC Paper #2170, Version 1, Updated 06/01/2013.

TABLE OF CONTENTS

1. INTRODUCTION

MBSE Applied to CubeSats

This paper extends the work reported in our 2012 IEEE Aerospace conference paper [**?**]. The paper reported on using Model Based Systems Engineering (MBSE) and the Systems Modeling Language (SysML) to model a standard CubeSat, and applied that model to an actual CubeSat, the Radio Aurora Explorer (RAX) mission [**?**].

A CubeSat is a type of miniaturized satellite with a standard form factor based on cubes with dimensions 10^3 centimeters and weighing less than one kilogram. CubeSats typically consist of one to three cubes.

RAX is the first CubeSat funded by the National Science Foundation (NSF) [**?**]. It has is a space weather mission designed to study plasma field-aligned irregularities in the ionosphere. It has enabled undergraduate students, graduate researchers, engineers, and scientists to be involved in the design, building, and operations of a satellite.

INCOSE MBSE Challenge Project

This project is a key part of the International Council on Systems Engineering (INCOSE) MBSE Challenge project. The Challenge project was initiated at the January 2007 INCOSE International Workshop [**?**]. The MBSE Roadmap, Figure **??**, was created to define the high-level, long term vision for the maturation and acceptance of MBSE across academia and industry.

Several MBSE Challenge teams were established to promote MBSE, advance the state of practice, and share lessons learned related to a diverse range of:

- MBSE applications
- Model scope
- Model quality and robustness
- Modeling standards
- MBSE process, methods, tools, and training

Space Systems Challenge Team

The INCOSE Space Systems Working Group (SSWG) established the Space Systems Challenge team. The Challenge team included aerospace students and professors from Massachusetts Institute of Technology and Georgia Institute of Technology. The initial focus was on the modeling of a hypothetical FireSat space system [?]. FireSat is a satellite for detecting, identifying, and monitoring forest fires. This system is used as an example in the widely used and accepted Space Mission Analysis and Design (SMAD) textbook [?]. Much was learned from modeling FireSat.

Our follow-on CubeSat project was initiated in April 2011 to model an actual space system, a standard CubeSat, with the RAX satellite being the point design.

The team now includes University of Michigan Aerospace graduate students and a departmental professor; the INCOSE SSWG, including engineers from NASAs Jet Propulsion Laboratory (JPL) and from modeling and simulation tool vendors InterCAX, Phoenix Integration and Analytical Graphics.

The collaborative environment includes a CubeSat - MBSE Google group, a MBSE Google documents collection, a No Magic Teamwork server for SysML modeling, and biweekly or weekly Web conferencing through the JPL-hosted Meetingplace server.

Advancement and Demonstration of MBSE State of Practice

Our Challenge team and project was created to assess, advance, and demonstrate the application of MBSE to the space systems domain.

We are developing a SysML model that incorporates several COTS tools:

- MagicDraw
- Cameo Simulation Tool Kit
- ParaMagic
- Systems Tool Kit
- PHX Model Center
- MATLAB

We are executing the model to analyze:

- Communication subsystem signal to noise ratio
- Solar energy collection and subsystem power consumption
- Activity flow including behaviors and interactions

2. MBSE AND SYSML

MBSE is the formalized application of modeling to support system requirements, design, analysis, optimization, verification and validation, beginning in the conceptual design phase and continuing throughout development and later life cycle phases [?].

The MBSE goal is to eventually replace the document-centric system engineering approach starting in the acquisition phase of a project and continuing on into operations.

Our application of MBSE uses SysML as the modeling language. SysML is a graphical modeling language for modeling systems. It is used to specify, analyze, design, optimize, and verify systems and their hardware and software components. SysML was developed by INCOSE and the Object Management Group(OMG)[?].

Figure **??** illustrates the SysML diagram types. A system is described in terms of:

- Structural block diagrams illustrating the constituent elements of a system and their connections
- Behavioral activity and state diagrams describing operational behaviors
- Parametrics definitions for operational constraints specified by values and/or equations
- Requirements text based requirements in the model that can be traced to design, analysis, and verification elements

SysML is used to model all aspects of a system either directly or through an interface with other models. It enables systems engineers to create and evolve models in an integrated, collaborative, and scalable environment. It enables building models that can be used in early design stages and that can support specification and design updates. Using models to define, develop, and ultimately operate a system is known as Develop With What You Fly With (DWWYFW).

Figure **??** illustrates that the MBSE environment is an integration of modeling tools and design tools along with viewing and report generation tools. This integration facilitates the analysis of alternative design models, and supports robust design optimization.

The ability to integrate, collaborate, and scale is centered around having a model repository. The repository is an information resource that is accessible through basic web-based technologies in addition to desktop applications. A variety of model editors can be integrated with such a repository, enabling engineers of all disciplines to collaborate. This integration is facilitated by the use of standard SysML approaches. Using Internet technologies to implement this approach provides a nearly unlimited ability to scale.

3. CUBESATS

CubeSats are type of low-cost, standardized nanosatellite (where a 1U is a cube 10 cm^3 on a side and approximately 1 kg) [?]. These small satellites are typically launched as secondary payloads. They have enabled the university community to design, build, and launch satellites using primarily off-the-shelf components. More recently, the worldwide community has adopted the CubeSat standard as a means

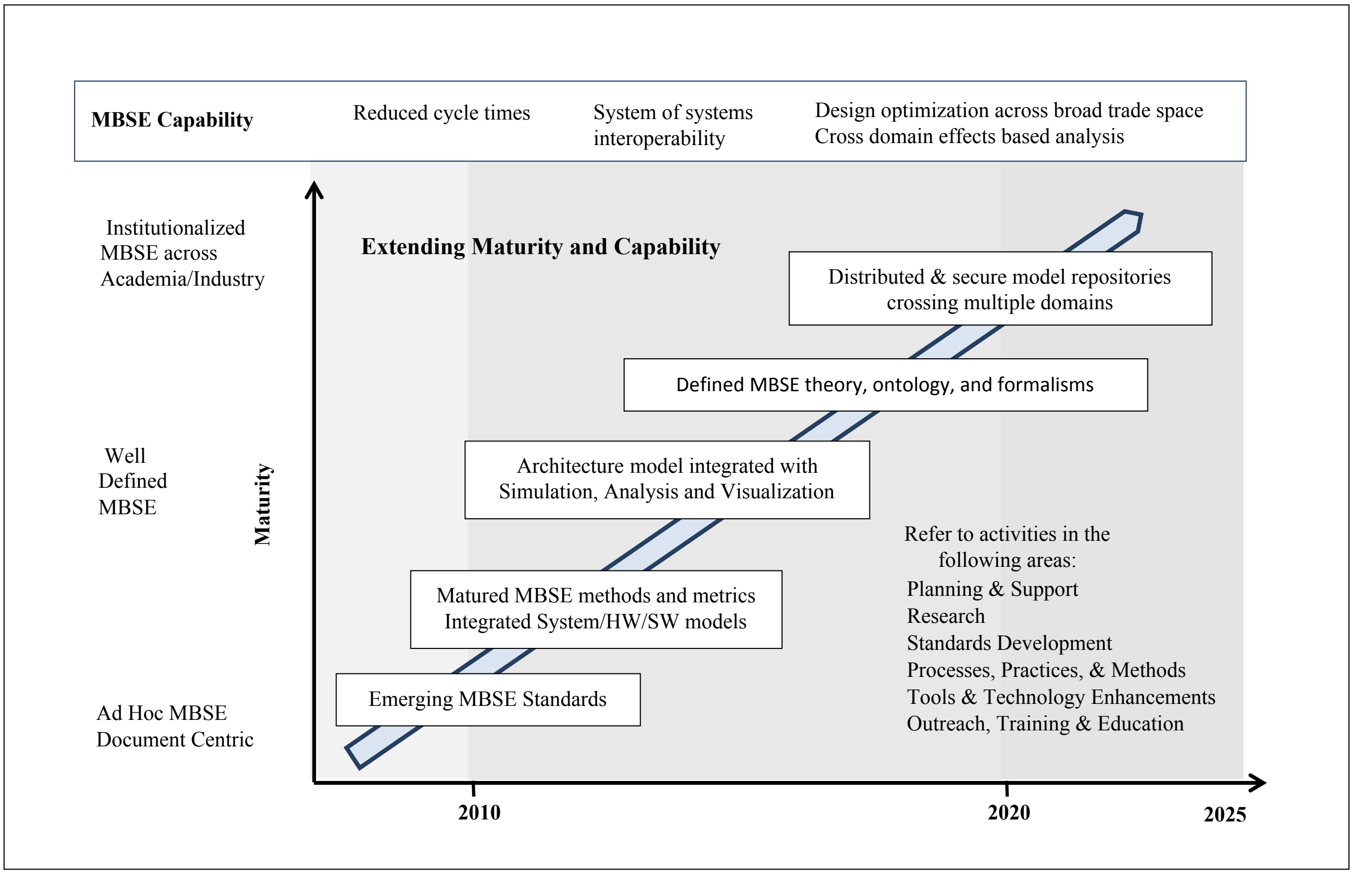

Figure 1. MBSE Roadmap

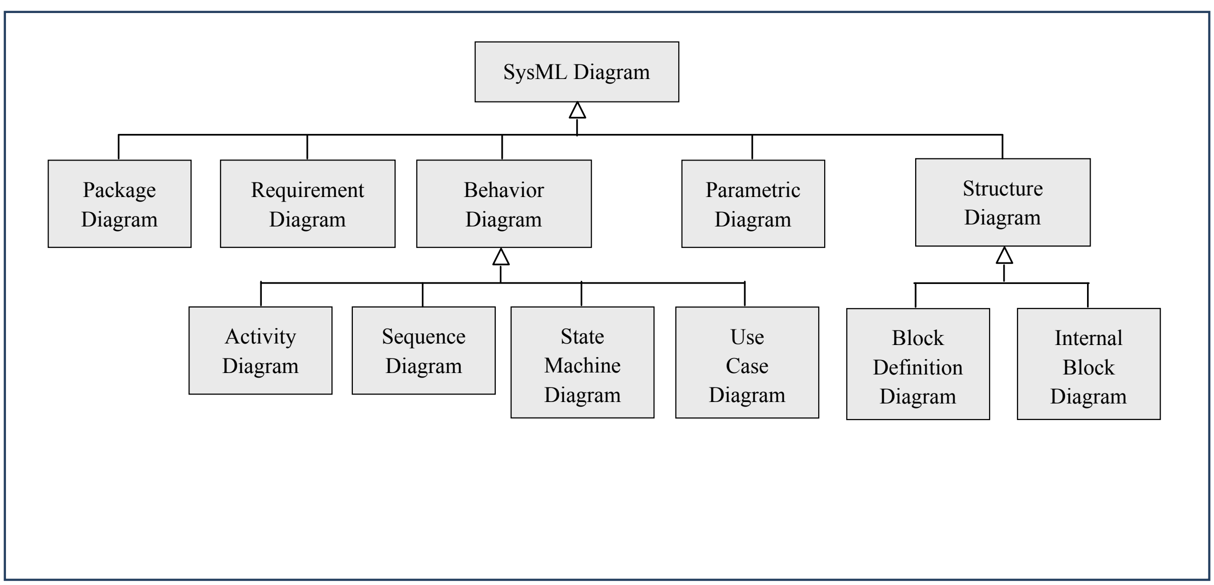

Figure 2. SysML Diagram Types, ⓒSanford Friedenthal, Elsevier Inc., 2012. All Rights Reserved

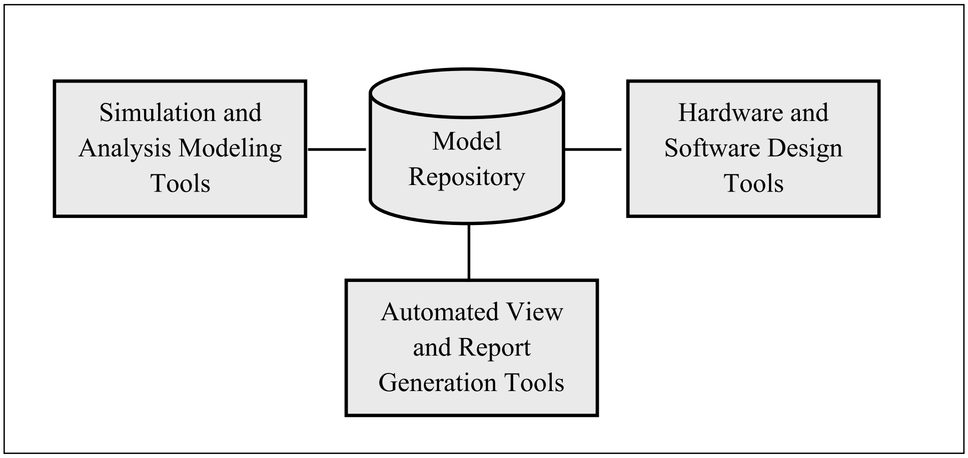

Figure 3. MBSE Environment

of performing novel scientific, surveillance, and technology demonstration missions at significantly reduced cost and with short development timelines.

Current Approach to CubeSat Design

The current approach to design and operational planning for CubeSat missions is largely intuition-based, often relies on simplified trade-studies that usually do not explore the complete design space [**?**]. Furthermore, ad-hoc and often unverified approaches are used to combine multiple simulation environments that often neglect elements of the mission dynamics. Designing he satellite at an early stage and neglecting key operational parameters can be problematic because decisions made in early design stages can have a significant impact on mission operations. For example, if a battery is sized prior to performing operational simulations, it may be of insufficient capacity to sustain the satellite throughout eclipse and be unable to satisfy mission operations requirements.

MBSE Approach to CubeSat Design

Our 2012 IEEE Aerospace conference paper delineated the CubeSat modeling objectives [**?**].

The current modeling effort is well under way, and has developed many of the early work products as indicated below. Our overall plan is to develop work-products for the CubeSat community that will include:

- A CubeSat meta-model describing CubeSat specific concepts and a modeling framework. The framework provides SysML structural and behavior models for the:
  - Mission
  - Mission elements which are systems that achieve the mission objective
  - Mission environment, e.g. space particles and fields as well as Earths atmosphere layers and magnetic fields
  - Flight system
  - Ground system
- An example CubeSat model that existing and future teams can use as a template for describing and modeling their own satellites, optimizing satellite design, and evaluating mission operations.
- The model will include:
  - The entire satellite mission including flight system, ground system, and targets of interest
  - Key satellite structure, including systems, subsystems, and components and their interfaces
  - Key satellite system and subsystem behaviors
  - Key satellite constraints and measures of effectiveness

The model will provide the techniques to interface CubeSat SysML models with a diversity of COTS modeling, analysis, and visualization tools. These tools can extract the portion of the information necessary to solve a problem or analyze a relevant part of the system and then integrate the solution back into the mission specification. For example, an optimization algorithm which takes as inputs satellite position and opportunities to collect energy and data, and then generates operational schedule can be interfaced with the SysML model.

The model will provide the capability to ensure that design updates comply with mission requirements and to communicate design updates to all engineers working on the mission.

Ultimately the models will be used by mission operators to evaluate mission planning, scheduling, and operations strategies considering position, attitude, on-board energy, data, and thermal states. This is will of paramount importance when responding to satellite component degradation and anomalies.

4. Integrated Tool Environment

Next we describe the simulation and analysis tools of the MBSE Environment shown in Figure **??** that enables us to analyze and optimize system performance. The simulation environment brings to life the models described in the previous section, where various aspects of the system model (parametrics, activities, and state machines) can be executed.

Conventional approaches often consist of simulators that are patched together in an ad-hoc manner, or require manual and time-consuming tasks when passing information between simulators. Unlike these approaches, our simulation environment enables the flow of information between simulators in an automated way, enabling users to easily evaluate different design configurations or reconfigure the analysis for different mission scenarios.

We use the following simulation tools to bring the SysML model to life:

- MagicDraw®from No Magic is a graphical SysML modeling tool that enables the analysis and design systems databases

- Cameo Simulation Toolkit®from No Magic enables different MBSE behavioral models such as SysML State Machines and Activity Diagrams to be executed within MagicDraw.

- STK®from Analytical Graphics is a tool that supports high fidelity simulation and visualization of satellite behavior including orbital dynamics and satellite subsystems models for power, thermal, sensors, attitude control, and telemetry.

- MATLAB®provides powerful numerical computing for evaluating equations, evaluating functions, executing algorithms, and plotting results. MATLAB can also interface with other optimization toolboxes and solvers.

- ParaMagic®is a SysML parametric solver and integrator for MagicDraw. It provides the ability to execute SysML parametric models and perform system trade studies from the earliest stages of system development. ParaMagic can execute constraint relationships that are math equations or wrap externally-defined models such as MATLAB/Simulink®, Mathematica®, and Excel. ParaMagic leverages the acausal nature of SysML parametric relationships to execute models in different causalities (swap inputs and outputs on-the-fly). It can detect and solve complex SysML block and parametric model structures, such as complex aggregates, recursion, and property and constraint redefinitions in the model hierarchy. Equivalent tools Melody®, Solvea®, and ParaSolver®are available for Rhapsody®, Enterprise Architect®, and Artisan Studio®respectively.

- PHX ModelCenter®allows users to create and execute simulation workflows by integrating various types of simulation models like Excel spreadsheets, STK scenarios, and MATLAB scripts. Once a simulation workflow is created, PHX ModelCenter executes the workflow, automatically transferring data from one model to the next. Users are able to execute multi-run studies by employing a rich set of trade study algorithms, including design of experiments, optimization, and reliability analysis. PHX ModelCenter can also be used to execute parametric models developed in MBSE tools like MagicDraw and Rhapsody, making it easier to evaluate performance and verify requirements throughout the design process.

The reason for using great set of diverse tools in the simulation environment is three-fold. First, we wanted to demonstrate how diverse tools could be integrated into a common framework. Second, we wanted to use the most appropriate simulators or mathematical engines environments for each particular simulation, and when possible, integrate existing code. Third, we wanted to test and determine which tools worked well for different applications (and could interface with other tools), thus we utilize and test a significant set of tools. However; a different, or smaller set of simulation or calculation tools could be utilized to accomplish similar goals.

5. RAX CubeSat

Mission Description

RAX is a space weather mission designed to study plasma field-aligned irregularities in the ionosphere [?]. It performs experiments using a bi-static radar configuration which utilizes a high-powered ground-based radar station. The primary station is PFISR, located in Poker Flat, Alaska, as shown in Figure **??**. The ground-based radar sends a high powered signal that reflects off the irregularities and are measured by RAX. On-board timing is provided by a GPS and position knowledge is provided by ground-based tracking systems.

RAX is passively magnetically aligned with the Earth's magnetic field using on-board fixed magnets, as shown in Figure **??**. This type of attitude control system enables RAX to have its antennas pointed towards the Earth when it passes over the experimental zone near the North Pole. Furthermore, the GPS antenna was installed on the opposite satellite face such that the antenna faces the GPS constellation during experiments, when accurate timing is critical. Oscillations are dampened with hysteresis material.

RAX-1 was launched in October of 2010 and RAX-2 was launched in November of 2011. RAX-2 is still performing experiments and being operated on a daily basis from the University of Michigan ground stations in Ann Arbor and ground station partners located around the world.

RAX SysML Model

The RAX satellite SysML models are based on the operational satellite framework developed in Ref. [?].

The SysML representations of the RAX model in this section provide a visual representation of how the system behavior can be evaluated using the simulations and then generates performance metrics based on the evaluations.

Figure **??** shows the RAX Block Definition Diagram (BDD) consisting of the RAX Launch System, RAX Environment, and RAX Mission. The majority of this paper focuses on the RAX Mission. However RAX Launch System and RAX Environment are also important in capturing the overall system. The RAX Mission model consists of both logical and physical models.

The logical models consider the operations of the system while the physical models consider the physical components. The decomposition strategy is typically used by CubeSat designers to separate functionality into subsystems that correspond to logical concepts.

For the CubeSat model, logical subsystem models describe the different concepts required to define the desired behavior of the system. The physical models specify the hardware and software that realize the logical design.

For example, one of the Power subsystem functions is to store energy. The physical battery hardware implements that functionality. Developing both logical and physical models allows the CubeSat systems engineer to clearly define the difference between the functionality (using logical models) and the hardware that supports this functionality (using physical

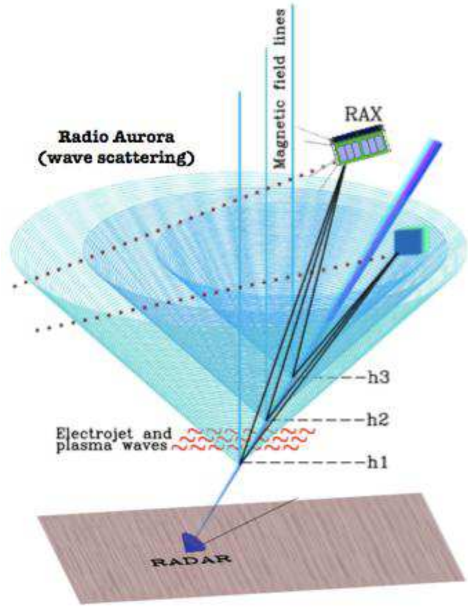

Figure 4. Radio Aurora Explorer (RAX) Satellite Mission

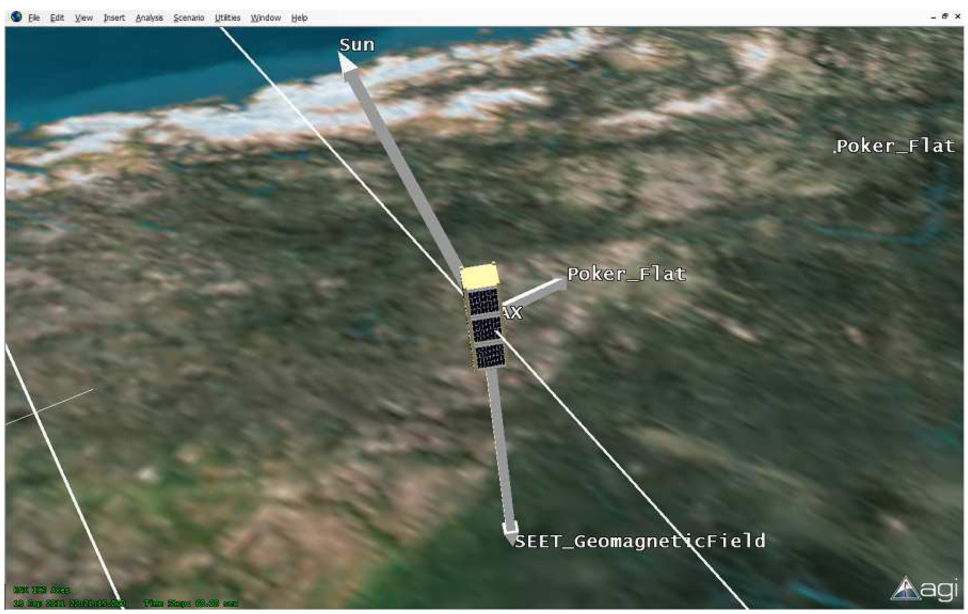

Figure 5. Schematic of RAX spacecraft with vectors pointing towards the experimental zone, Poker Flat, AK, the sun, and along the Earth's magnetic field (which the spacecraft long axis is aligned with). The figure is generated using STK.

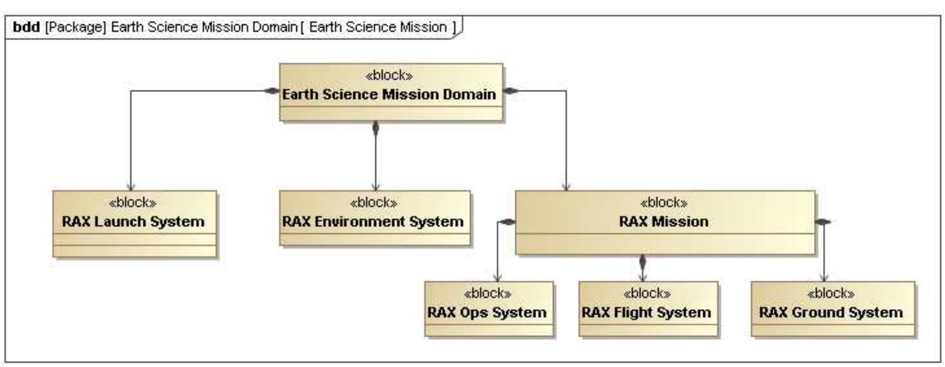

Figure 6. RAX Mission Block Definition Diagram (BDD)

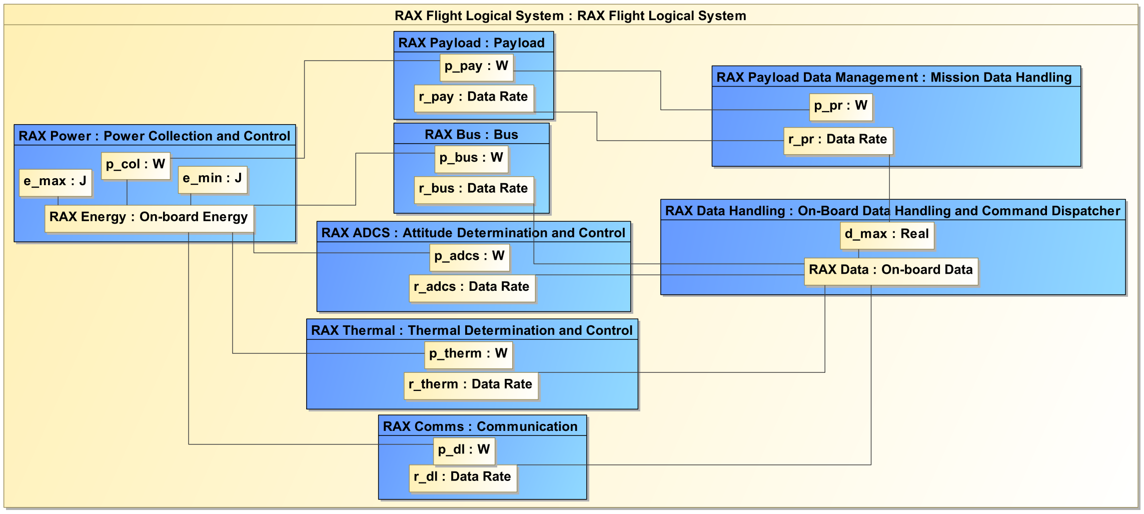

Figure 7. RAX Mission Internal Block Diagram (IBD) with Subsystems and Interactions

models).

The focus of this paper is on the operations of the RAX system, thus we focus on the logical models. As described in Ref. [**?**], RAX has several functional subsystems, each supporting at least one critical part of the mission or other subsystems. These subsystems are detailed in Ref. [**?**]. The Internal Block Diagram (IBD) shown in Figure **??** illustrates how the subsystems for the RAX Logical Flight System interconnect along with some of the key properties for each of the subsystems that are used in the analysis.

The Power Collection and Control subsystem is responsible for acquiring energy from body-fixed solar panels, distributing power to support ongoing operations, and storing excess energy for future use in an on-board battery. The On-board Data Handling and Command Dispatcher subsystem is responsible for dispatching commands, and managing the storage of on-board data.

The Mission Data Handling subsystem is responsible for processing, compressing, deleting, and filtering data for the satellite payload. The Communication subsystem receives commands from and downloads data to the Earth ground stations.

The Attitude Determination and Control, Thermal Determination and Control, Structures and Mechanism subsystems are self-explanatory, and are passive for the RAX satellite (i.e. are not active).

6. Analytical Model and Results

Figure **??** illustrates the application of the RAX system model to analyze:

- Communication subsystem signal to noise ratio
- Power
- Flight System Behavior

Communication Subsystem Signal to Noise Ratio (SNR) Analysis

Due to the importance and challenges of communication in the design and operation of small satellites, we provide a detailed view of the communication subsystem in this section. The SysML model presented in this section is based on the model in Ref. [**?**].

The main purpose of the communication subsystem is to download data from the satellite to ground stations. In this case there is assumed to be only one ground station. We want to analyze the signal-to-noise ratio, SNR , of the communication link established between the communication subsystem and the ground station, which must be greater than a minimum level, SNR_{min} , based on the error rate acceptable in transmission.

The SNR Analysis block in Figure **??** represents the SNR analysis that we want to perform. The link equation used for the analysis uses design variables specific to the communication subsystem (Communication block), network of ground stations (Ground Network block), atmosphere (Atmosphere block), and the satellite trajectory (Orbital Elements block).

Figure **??** shows the parametric model for the SNR Analysis block. The parametric model shows the link equation (calcSNR constraint property) which relates the SNR analysis variable to the system design variables owned by the communication subsystem, ground stations, atmosphere, and satellite trajectory. The parametric model also shows the space loss equation (calcLS constraint property) that relates the space loss (L_s) to propagation path distance (L_p). These equations are represented in log form according to industry practice.

SysML parametrics are acausal in nature. The mathematical constraints in the parametric model are represented in a declarative manner. This implies that there are no fixed inputs and outputs specified at the parametric model level. The same parametric model can be solved with different combinations of inputs and outputs such as y=kx and where we solve for y given x and k. Or x=y/k where we solve for x given y and k. We can solve the equations with different combination of dependent and independent variables.

Figure 8. RAX Models and Simulations as related to various simulation environment

So the intent is to use the given parametric model in three different analysis scenarios:

- Analysis Scenario 1: Given the data download rate (r_dl) and the available power (p_dl), compute the SNR for the communication link.

- Analysis Scenario 2: Given the data download rate (r_dl) and the desired SNR, compute the power required (p_dl).

- Analysis Scenario 3: Given the available power (p_dl) and the desired SNR, compute the data download rate (r_dl) that can be achieved.

ParaMagic leverages the SysML standard to execute parametric models in the context of block instances, where each instance represents a specific design alternative or configuration or scenario in this case. With ParaMagic, we can execute a given parametric model for different causalities - input and output variables can be switched on-the-fly.

Figure **??** shows the SysML instance structure (block definition diagram)for an analysis of a specific design configuration with specific values of the properties of the design. Figure **??** shows the ParaMagic browser for Analysis Scenario 1. As shown in the figure, all of the value properties have assigned values except for SNR and L_s. SNR is assigned target causality as the value of interest for Analysis Scenario 1 and L_s is left with undefined causality which means it will be solved only if needed to find the target value. Figure **??** shows the solved value of SNR, boxed in red. As SNR_{min} = 13 dB, this value is acceptable and therefore the power allotted in the design is sufficient for the specified data download rate and acceptable error rate. The Update to SysML button at the right of the browser allows the user to update the solved values to the instance model and diagram.

Figure **??** shows the ParaMagic browser for Analysis Scenarios 2 and 3 (SysML instance structure not shown). It shows that SNR has been assigned given causality and value 13, equal to SNR_{min} . For Analysis Scenario 2 (LHS),

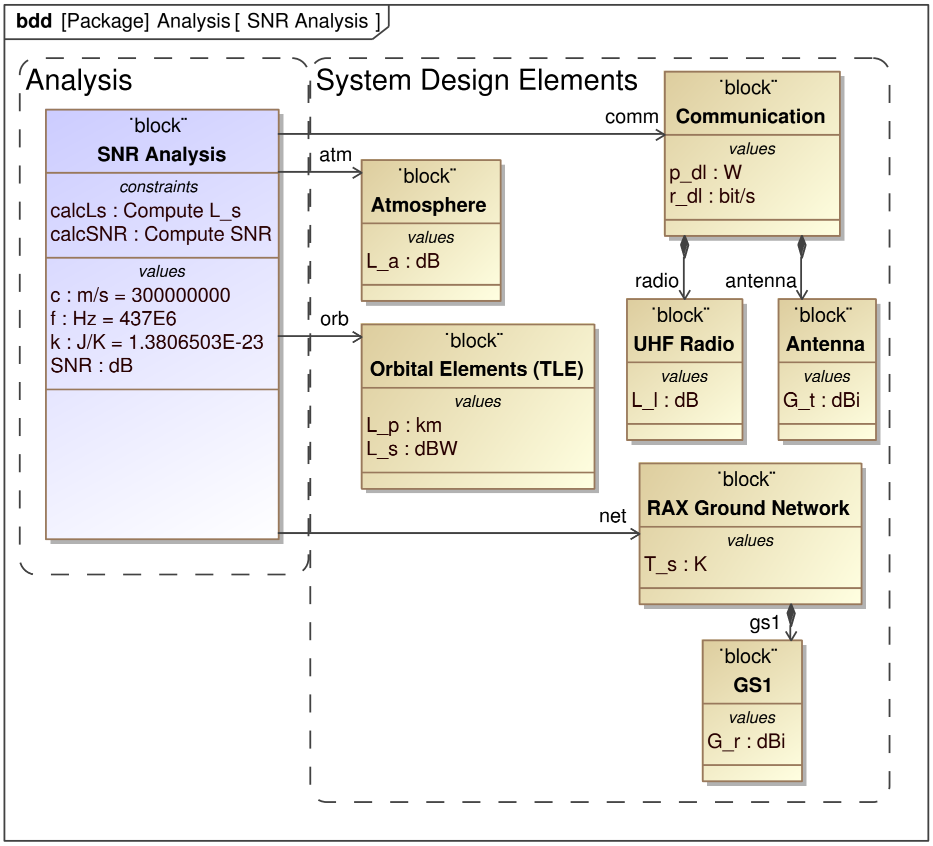

Figure 9. SysML BDD illustrating the SNR Analysis model setup

the power required for download p_dl is computed given the minimum acceptable SNR and data download rate. For Analysis Scenario 3 (RHS), the data download rate r_dl is computed given the minimum acceptable SNR and available power.

Simulation Model and Results

Power Analysis—To capture realistic power scenarios, we have developed a simulation that consists of PHX ModelCenter as the glue that ties together simulations and analysis components from STK, SysML, and MATLAB. We model the dynamics of opportunities to collect energy and download data and how this impacts the time history of the satellite states, including the on-board energy and data, and the amount of downloaded data.

We create a workflow for an example mission scenario, which includes data and energy collection, on-board operations, and data download over a specified ground station. The simulation is executed during a specified scenario time. These state dynamics are a function of performed operations, including nominal, payload, and download operations, and available energy collection from the sun. We implement the RAX-specific scenario by combing the MagicDraw parametric model in Figure **??** with an orbital scenario from STK®and custom analysis MATLAB scripts using PHX ModelCenter, as shown in Figure **??**.

The simulation is a workflow that is created graphically by dragging and dropping reusable components and combining them using if-else branches, loops, and other flowchart-like constructs (using PHX ModelCenter). The graphical link editor is used to specify what data should be passed from one application to the next when the model runs. Through a graphical user interface accessible from within the MBSE tool or PHX ModelCenter, we then execute a PHX ModelCenter model defined by a SysML parametric diagram.

With this simulation environment, we can evaluate design configurations, perform trade studies, and check requirements compliance. Analysis can also be automatically re-run with updated the attribute values.

We execute the power scenario in Figure **??** using the simulation workflow created in PHX ModelCenter, which automatically executes the workflow one or more times, utilizing parallel computing resources as needed. When instructed, each component is executed automatically, transferring information between components. Using the simulation environment described above, we can perform a parametric study using the multi-dimensional data visualization tools in PHX ModelCenter ®to help interpret and analyze the results.

Flight System Behavior Analysis—Cameo Simulation Toolkit was used to analyze the RAX behavior and interactions. Simulation in this context means to execute the model so that an understanding of the RAX System interactions and behaviors can be understood. Since a model is a simplified representation of the actual System, in this case RAX, creating a model that allows for simulation to be useful for analysis was an iterative process.

CAMEO Simulation Toolkit provides the ability to execute,

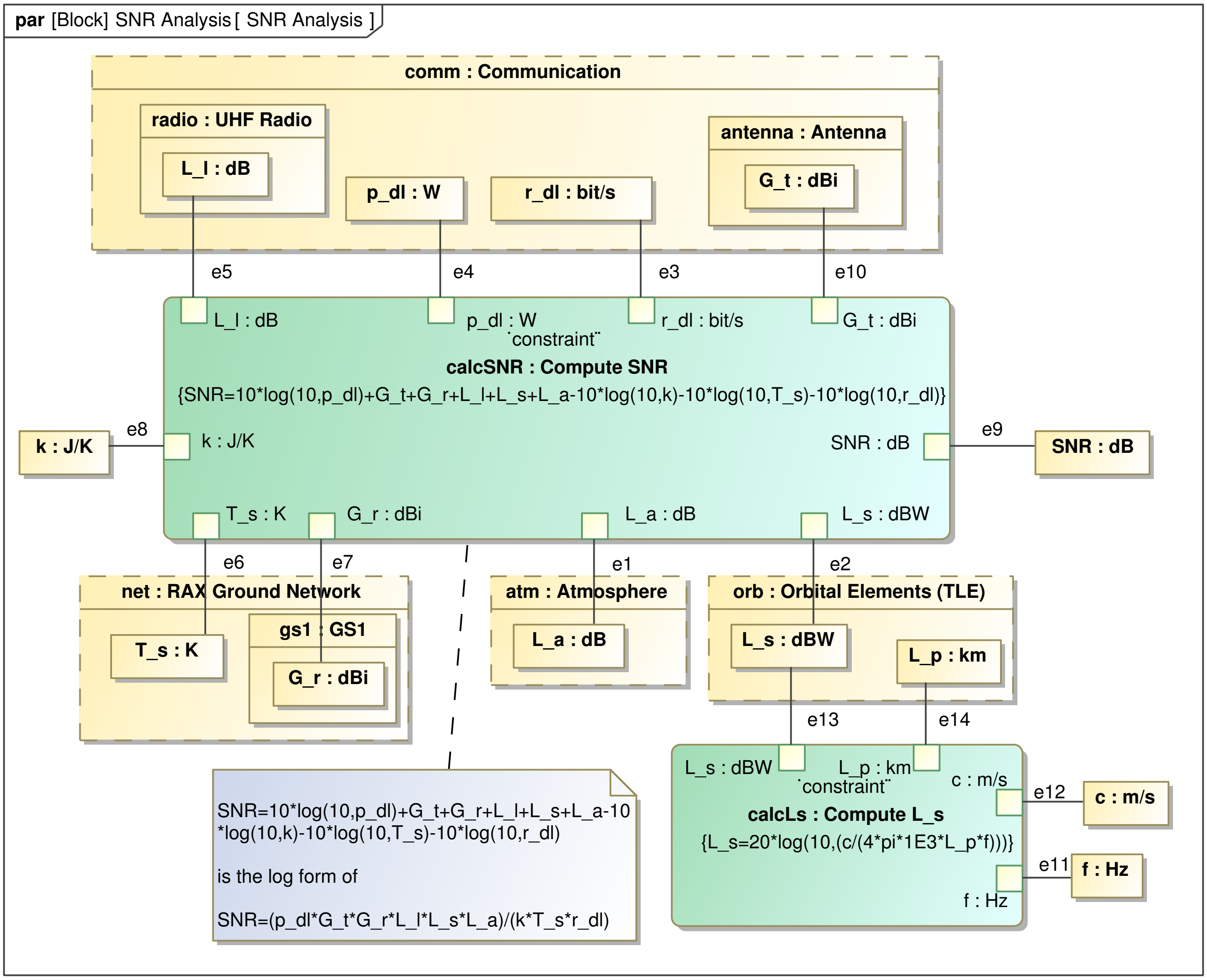

Figure 10. SysML parametric diagram showing the communication link analysis model

animate, and debut state machines and activity models. The sequence of steps is to run a simulation, view the behavior by the model, and update the design appropriately if a different behavior is needed. This type of functionality also supports verification and validation of the system.

The Mission Operations System (MOS) consists of the hardware, software, procedures, and personnel that enable control of the Flight System as well as analysis of the Flight System behavior. The MOS operation team generates sets of commands that are to be executed on-board the Flight System. For RAX the on-board computer (OBC) is the main handler for processing commands and sending them out to the relevant subsystems for execution. This process was simulated in the RAX Model as described below.

Figure **??** shows the interface between the RAX Flight System and the RAX Ground System. The sets of commands are uploaded to the Flight System and provide the schedule on when and how to perform an experiment. For the RAX spacecraft, the experiment times are based on when the spacecraft is over the target of interest and there is the predicted level of energetic activity.

The upload consists of sending a command signal from the ground station that traverses the Flight-Ground Interface as is then received by the OBC. The OBC has knowledge of the time and can dispatch the command information to the appropriate subsystems when a command approaches execution time.

Figure **??** shows the states for the Main Flight Computer. Also shown are states that have underlying behavior that is pertinent to that state. In this case the Command Processing State has underlying behavior for dispatching commands.

Figure **??** depicts the behavior that the OBC performs in order to analyze command files sent from the ground. In this snippet of the process shown, the OBC determines what subsystem is being affected and whether or not this system is going to upload or download mode. Once the determination is made, the OBC sends the final signal data to the Communications Subsystem, shown in Figure **??**.

In Figure **??**, the states for the Communications Subsystem are shown. Nominally the system is in the beaconing mode, but once a signal is received from the Main Flight Computer

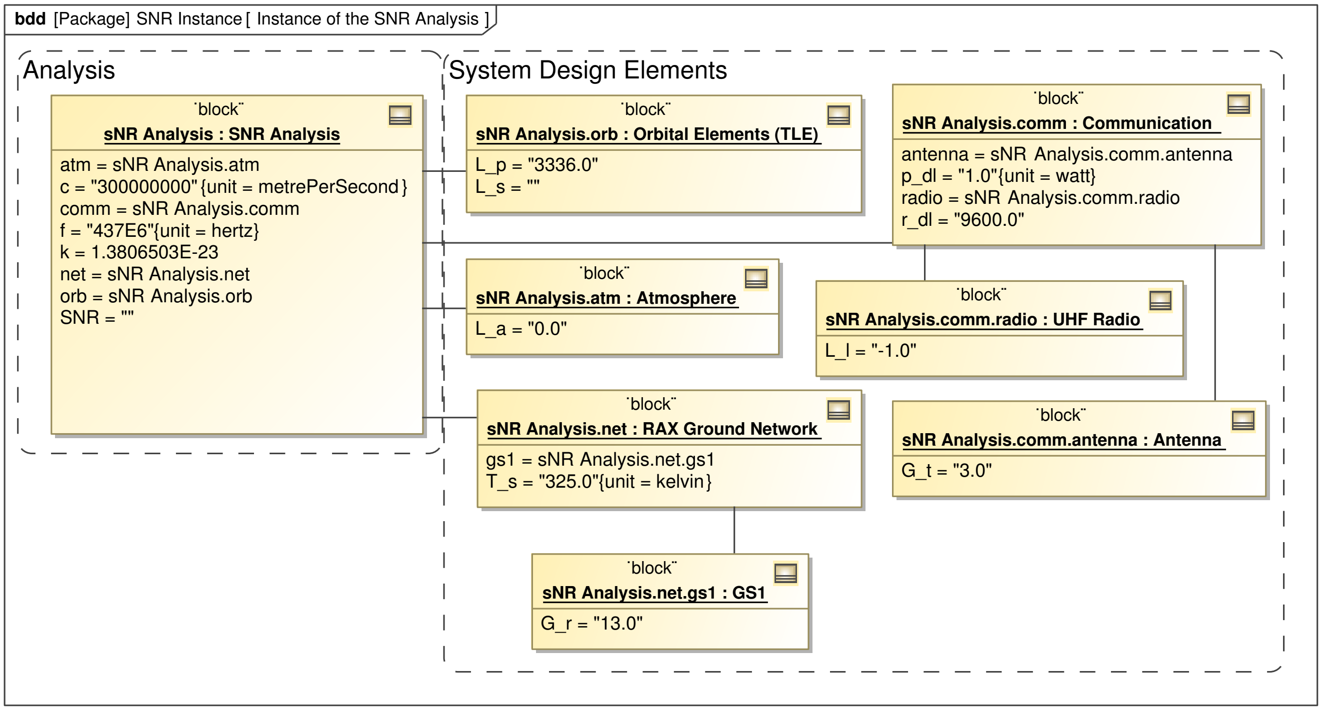

Figure 11. SysML BDD illustrating the instance structure setup for Scenario 1

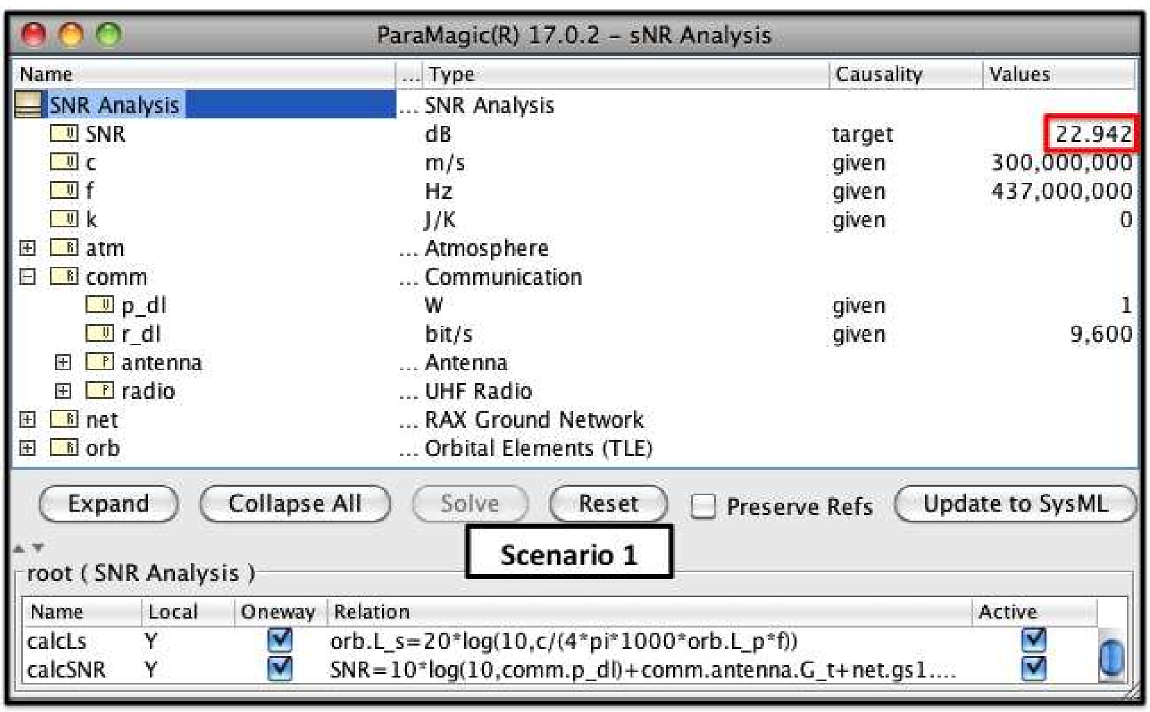

Figure 12. ParaMagic Browser showing results for Analysis Scenario 1

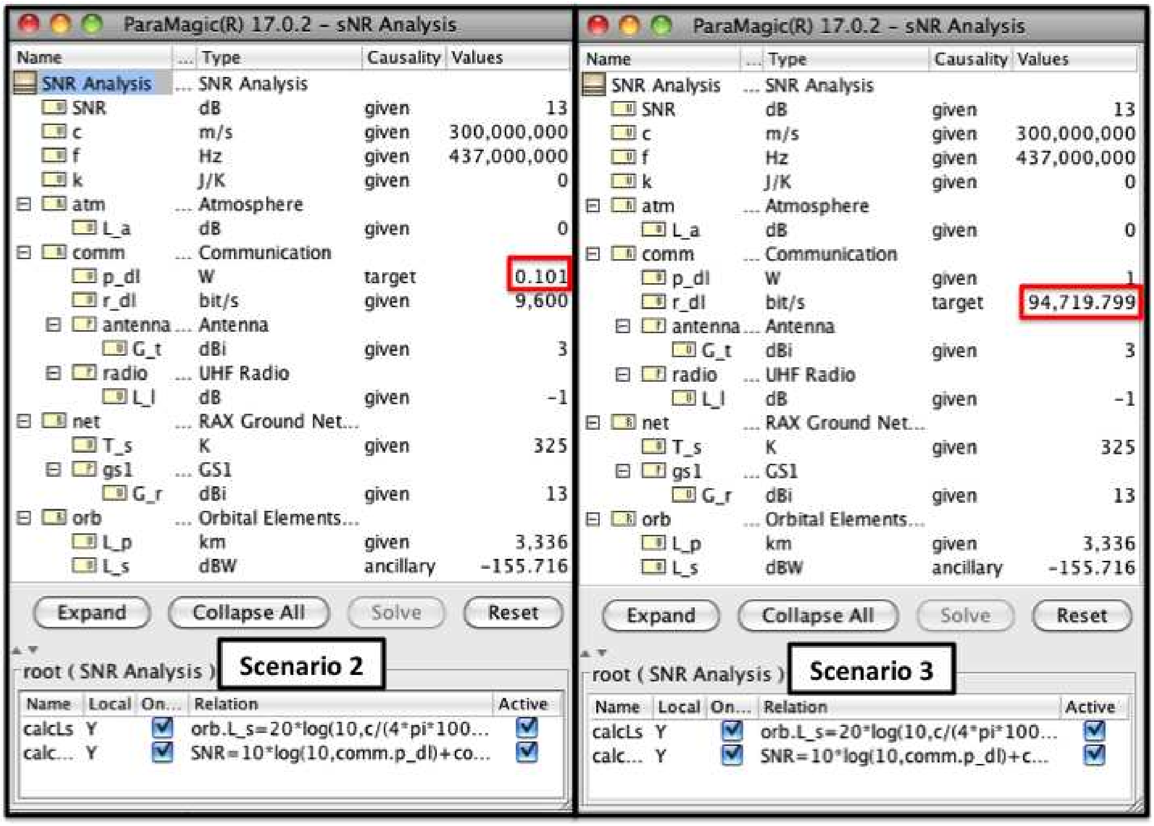

Figure 13. ParaMagic Browser showing results for Analysis Scenarios 2 and 3

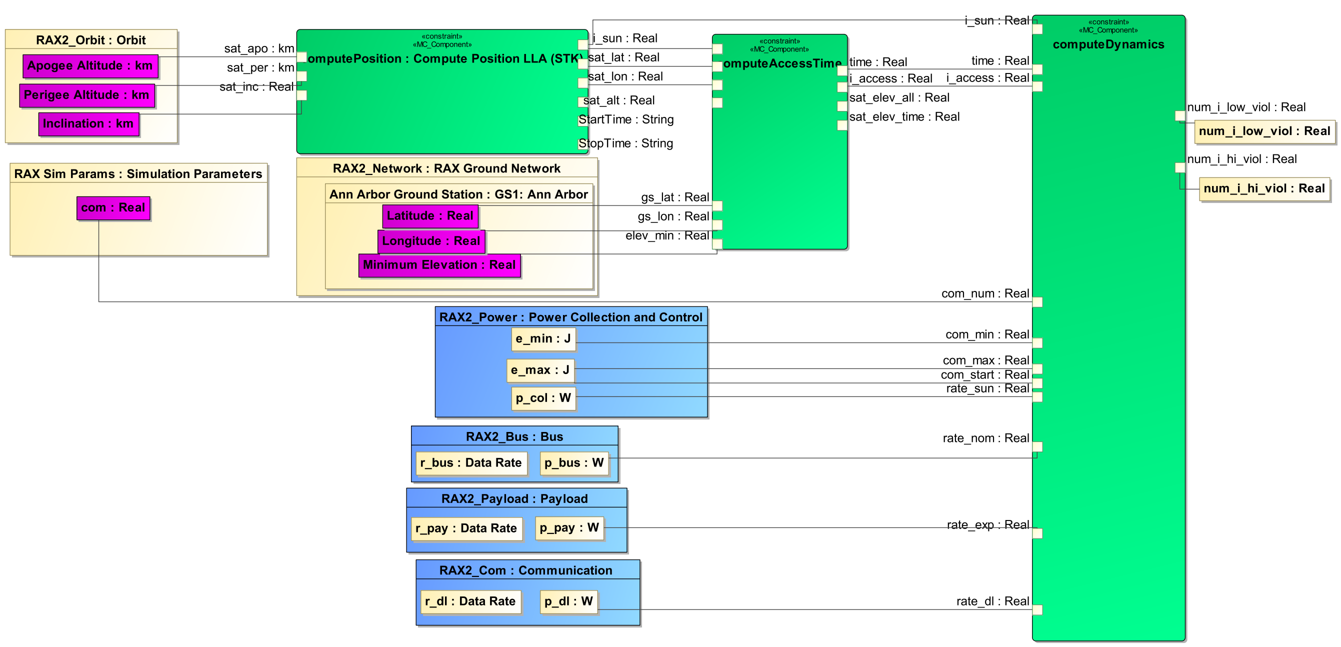

Figure 14. Parametric Diagram showing RAX Power Scenario in MagicDraw

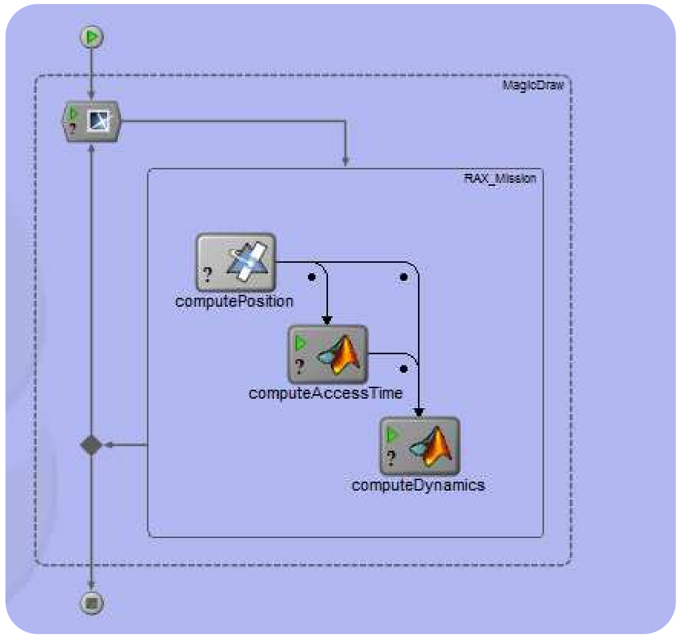

Figure 15. RAX Power Scenario in PHX ModelCenter integrated with STK and MATLAB

that indicates whether the Flight System is uploading or downloading data, the communications system transitions to the relevant state.

Using CAMEO Simulation Toolkit allows for the interfaces to the different systems of the RAX Mission System to be analyzed and the actual information exchange between systems to be depicted and tested. The expected behavior as well as on-flight observed behavior can be compared against what the model is saying will occur. If a model is developed in the early phases of the Mission, these types of simulations will allow for verification and validation of the mission software and interfaces.

7. CONCLUSION

Summary

The RAX model described in this paper demonstrates the utility in using a standards-based approach for modeling the system design and analysis using a ”develop as you fly” philosophy. The BDD and IBD diagram structures of SysML are the starting point, establishing the fundamental relationships and interfaces between the components of our system. Going beyond traditional static system representations, we add parametric diagrams to enable interactive analysis of the design based on established physical principles (e.g. communications link margin, power constraints). Furthermore, time evolution of our system introduces the various states the flight and ground system undergoes. These states are defined in the State Machine diagrams. Block representation, parameterization, and state definition all serve as the glue that ties the system together, and provides the framework for integrating the design model with the analytical models.

The role of the systems engineer is to understand all parts of the system in order to describe how the whole system works. Unlike traditional requirements approach using declarative ”shall statements”, the formalized descriptive language of SysML is not only human readable, but also allows for machines to read and interpret the description. This capability allows for the integration of seemingly disparate analysis tools (e.g. Excel, Mathematica) into an integrated modeling environment.

We developed the MBSE simulation environment presented in this paper using a modular approach, which enabled easy growth of the model and multiple modelers to simultaneously contribute to the model. We first identified key framework elements, such as the subsystems, states, and their interactions. All modeling elements were introduced in the context of building or executing an analysis or simulation, which ensured they were required and minimize the complexity of the model. The framework is thus easily extended to include additional modeling elements, higher fidelity simulators, or more interactions between the components. We also integrated existing software code into the simulator. A variety of modelers with different levels of expertise (ranging from beginner to expert SysML user) contributed to the model. Beginners found the learning curve reasonable, as they were building off the work of the experts and thus learning as they contributed. Beginners found working with SysML as the beginning easier if they had experience with the CubeSat system itself or other simulator.

Lessons Learned: Challenges and Successes

Throughout developing the models and simulations in this paper, we have experienced several lessons learned that are listed below:

- We were able to extract time-dependent parameters in PHX ModelCenter using a specific post-processing script and vendor support. This was a great advantage for executing the dynamic power system scenario.

- We were able to setup and execute SNR analyses for the communication sub-system for different scenarios using ParaMagic. It enabled us to setup the parametric model once and execute it for different causalities, e.g. computing SNR given available power and data download rate, and

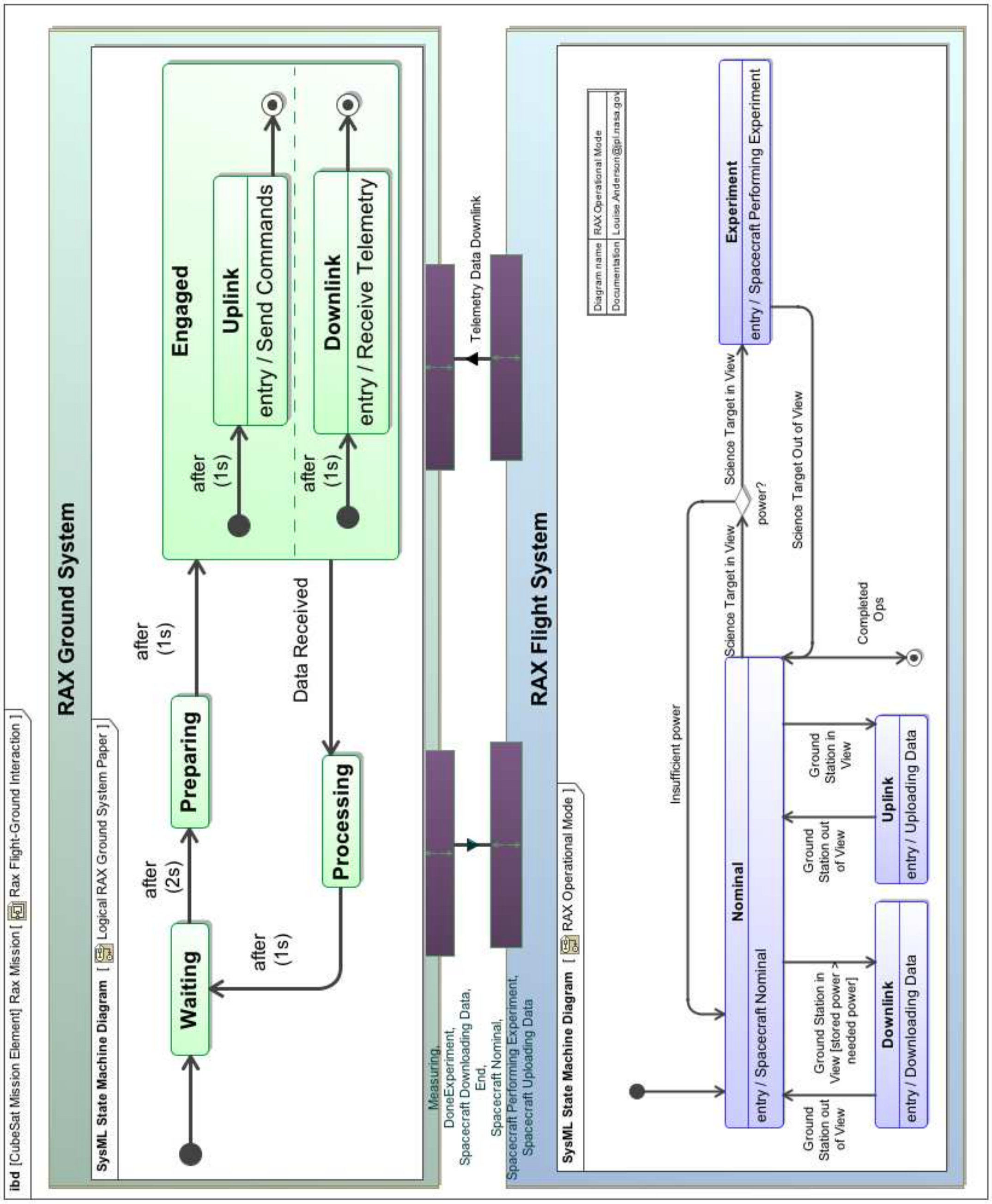

Figure 16. Internal Block Diagram - RAX Flight System - RAX Ground System

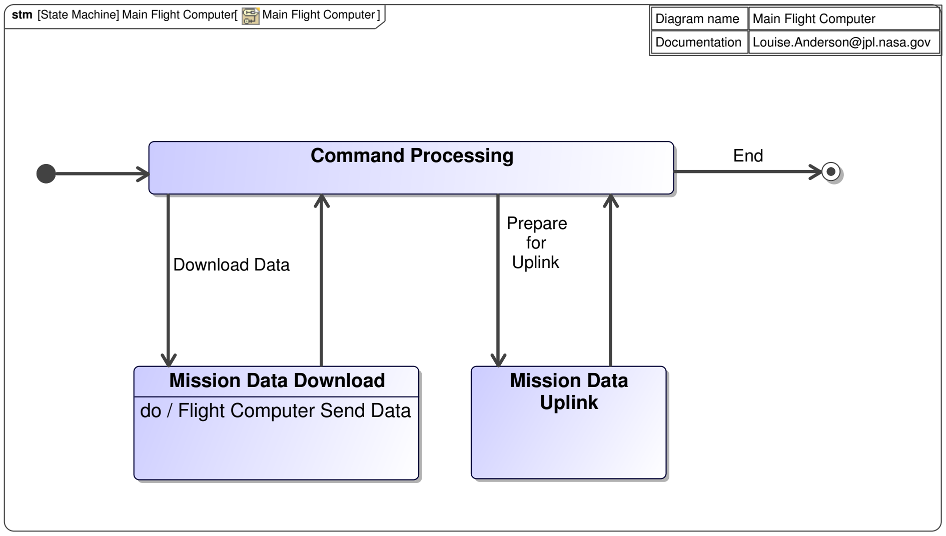

Figure 17. State Machine - Main Flight Computer

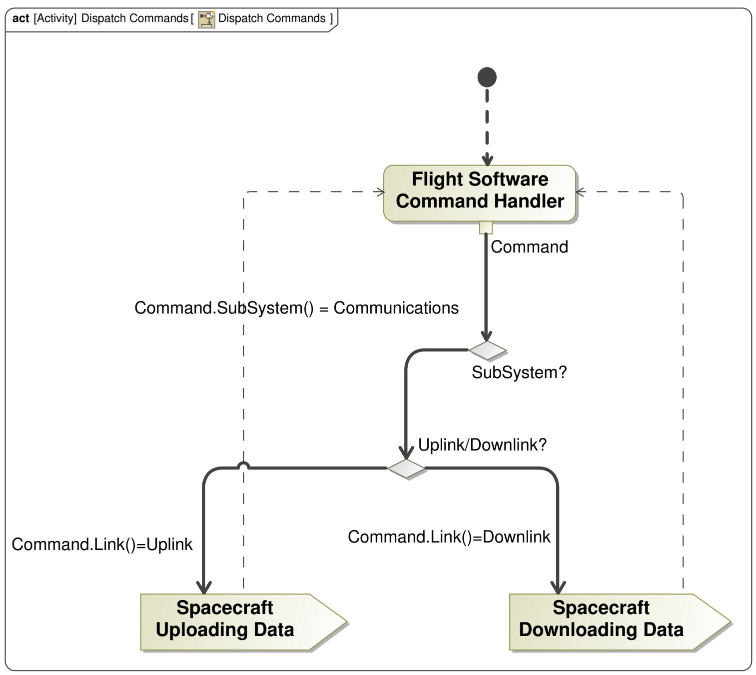

Figure 18. Activity Diagram OBC Dispatch Commands Behavior

computing required power given acceptable SNR and data download rate.

We also encountered several challenges, listed below:

• Appropriate licenses are required for all simulation tools, which can be challenging, and required vendor support.

Future Work

Beyond the models, simulations, and analyses demonstrated in this paper, there are additional ways to extend this work to more sophisticated analyses that can aid in both vehicle and mission operation design. Extensions include:

• Using ParaMagic to execute parametric models, such as compute different performance parameters, during state machine simulations in a given state and during transitions.

• Wrapping STK models and AGI components as parametric constraints and execute using ParaMagic. This capability is in a prototype stage right now.

• The simulations currently allow the model to be stepped through in time to aid in visualizing what is occurring with spacecraft behavior. In the future, extending this approach to include constraint-based solving would give the full analysis picture. With simulation the different states can be constrained from occurring based on value properties received from the constraint modeling. With both methods working

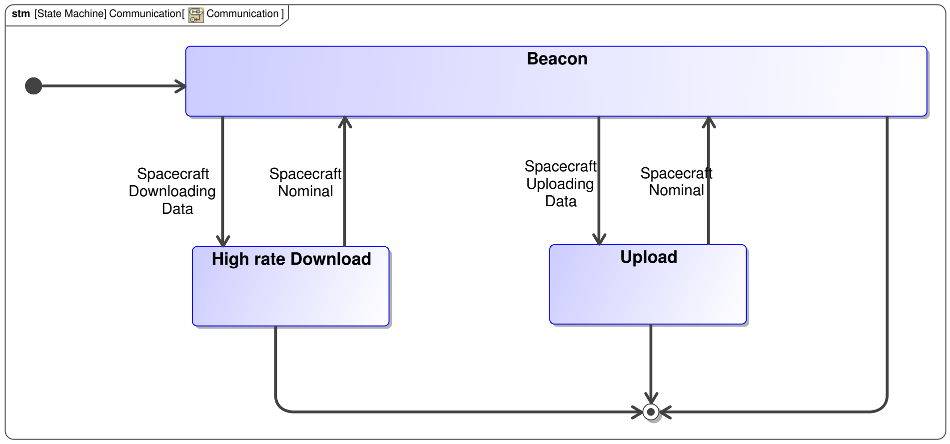

Figure 19. State Machine - Communication Subsystem

together, a dynamic approach of changing input values could be used to evaluate the equations and to visualize the behavior of the spacecraft based on input values.

• The various simulations in this paper currently execute individually. Future work will bring these simulations together such that broader simulations can be performed, for example the power and communication systems could be analyzed and optimized simultaneously.

• The ability to verify optimal scheduling algorithms in the simulation environment would be extremely useful, as there is currently no unified environment where this can be done efficiently. In particular, it would be helpful to be able to assess the robustness of operational schedules to perturbations in various input parameters, which can likely be achieved with the Monte Carlo analysis in PHX ModelCenter.

• Beyond demonstrating mission scenarios and performing trade studies, this environment may also be useful for combined vehicle and operations optimization. In particular, due to the ease of identifying inputs and outputs, it is possible to vary specific parameters (assigned as inputs) and monitor how they impact other parameters (assigned as outputs). This type of analysis can be extremely useful for designing ground systems, sizing satellite components, etc. Furthermore, these types of trades can be useful for developing the simulation system that will be used for later design trades and flight simulations, along the lines of develop with what you fly with.

• Finally, comparing results from our simulation environment based on our modeling framework to real data from operational missions such as RAX will provide a means to verify, validate, and improve the models.

ACKNOWLEDGMENTS

We thank the entire University of Michigan and SRI International RAX Teams for their contributions. We also thank Analytical Graphics, Inc (AGI), Phoenix Integration, and InterCAX for their generous support of our work. We thank the Natural Sciences and Engineering Research Council of Canada (NSERC) and Zonta International for their support.

We would also like to thank Sandy Friedenthal for his invaluable review, contributions, and guidance of this work and paper.

Parts of this research were carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

REFERENCES

[1] 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 *IEEE Aerospace Conference*, Big Sky, MT, March 2012.

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

[3] T. Moretto, “Cubesat Mission to Investigate Ionospheric Irregularities,” in *Space Weather: The Journal of Research and Applications*, November 2008.

[4] I. C. on Systems Engineering (INCOSE), “Mbse initiative,” January 2007. [Online]. Available: https://connect.incose.org/tb/MnT/mbseworkshop/

[5] J. Wertz and W. Larson, *Space Mission Analysis and Design*, 3rd ed. Microcosm Press, 1999.

[6] I. C. on Systems Engineering (INCOSE), INCOSE Website. [Online]. Available: http://www.incose.org/ProductsPubs/products/sevision2020.aspx

[7] O. M. G. (OMG), OMG Website. [Online]. Available: http://www.omgsysml.org/

[8] K. Woellert, P. Ehrenfreund, A. J. Ricco, and H. Hertzfeld, “Cubesats: Cost-effective science and technology platforms for emerging and developing nations,” *Advances in Space Research*, vol. 47, no. 4, pp. 663 – 684, 2011.

[9] S. Spangelo and J. Cutler, “Optimization of single-satellite operational schedules towards enhanced communication capacity,” in *AIAA Guidance, Navigation, and Control Conference*, Minneapolis, MN, August 2012.

BIOGRAPHY

Louise Anderson *is an early career hire Software Systems Engineer at JPL. She's currently on the Ops Revitalization team in Multimission Ground System and Services (MGSS). Louise is also currently Co-Lead of the Modeling Early Adopters group at JPL. She graduated in May 2010 from the University of Colorado-Boulder with a degree in Aerospace Engineering. Previously she worked at the Laboratory for Atmospheric Space Physics in Boulder Colorado working as a Command Controller on the Mission Operations Team. She has worked on the mission ops team for Kepler, Sorce, AIM, Quikscat, and Icesat. Louise is a member of INCOSE, currently for INCOSE she is working on the Space Systems Working Group specifically on CubeSat Modeling. .*

Manas Bajaj, PhD *is the Co-Founder and Chief Systems Officer at InterCAX (www.InterCAX.com) where he leads the development of software applications for MBSE. He has successfully led several government and industry-sponsored projects. Dr. Bajaj has been actively involved in the development, implementation, and deployment of the OMG SysML standard and the ISO STEP AP210 standard for electronics. He is a Content Developer (author) for the OMG Certified Systems Modeling Professional (OCSMP) certification program, and coaches organizations on SysML and MBSE. Dr. Bajajś research interests are in the realm of SysML and model-based systems engineering (MBSE), computer-aided design and engineering (CAD/CAE), advanced modeling and simulation methods, and open standards for product and systems lifecycle management (PLM/SLM). He has authored several publications and won best paper awards. Dr. Bajaj earned his PhD (2008) and MS (2003) in Mechanical Engineering from the Georgia Institute of Technology, and B.Tech. (2001) in Ocean Engineering and Naval Architecture from the Indian Institute of Technology (IIT), Kharagpur, India. He is an INCOSE member and participates in the OMG and PDES Inc working groups.*

Bjorn Cole *is a systems engineer in the Mission Systems Concepts section of the Jet Propulsion Laboratory. His research interests are in the fields of design space exploration, visualization, multidisciplinary analysis and optimization, concept formulation, architectural design methods, technology planning, and more recently, model-based systems engineering. He is currently working on pre-Phase A mission formulation. He earned his Ph.D. and M.S. degrees in Aerospace Engineering at the Georgia Institute of Technology and his B.S. in Aeronautics and Astronautics at the University of Washington.*

Leo Cheng *is currently a member of the Flight Control Team for the Spitzer Space Telescope. In addition to his real-time operations experience, Leo has extensive experience in the area of science planning and operations, including leading the development of the science plan for the Cassini mission's Huygens probe descent and landing on the surface of Titan. Originally from the island of Guam, Leo holds a B.S. Degree in Physics from Cal Poly Pomona University and a M.S. Degree in physics from Rensselaer Polytechnic Institute.*

James Cutler *received a B.Sc. degree in Computer and Electrical Engineering from Purdue University, and M.S. and Ph.D. degrees in Electrical Engineering from Stanford University. He is currently an assistant professor in the Aerospace Engineering Department at the University of Michigan. His research interests center on space systems a multidisciplinary approach to enabling future space capability with particular emphasis on novel, nanosatellite missions. He is developing next generation communication capability, design optimization techniques, and space weather measurement missions. His research lab is developing and flying multiple nanosatellites for NSF and NASA.*

Christoper Delp *is the Systems Architect for the Ops Revitalization task in MGSS. He is also a Systems Engineer on the Europa Habitability Mission Model Based Systems Engineering Team. He is a founder of the Modeling Early Adopters grass roots Model Based Engineering working group. Previously he served as Flight Software Test Engineer for MSL and Software Test Engineer for the Tracking, Telemetry, and Command End-to-End Data Services. He also leads the INCOSE Space Systems Working Group's entry in the Model Based Systems Engineering Grand Challenge. Additionally, he has performed research on software verification and tools for Service-Oriented Architecture in support of the Deep-space Information Services Architecture. Prior to coming to JPL, he worked as a software engineer performing DO-178b Level FAA flight qualified software development and testing on Joint Tactical Radio System (JTRS) and the T-55 Full Authority Digital Engine Controller (FADEC). Chris earned a Master of Science in Systems Engineering from the University of Arizona where he studied Model Based Systems Engineering, Simulation and Software Engineering. Previous to graduate studies, Chris performed his duties as a systems engineer on Missile Systems Verification and Validation.*

Elyse Fosse *is a Software Systems Engineer for the Ops Revitalization task in MGSS. She also develops ground system cost models for deep space and Earth missions. She is also a member of the Multimission Ground Data System Engineering group at the Jet Propulsion Laboratory. Her interests include software and systems architecture, applications of model-based system engineering, and cost model implementation and analysis. Elyse is also a part of the INCOSE Space Systems Working Group's entry into the Model Based Systems Engineering Grand Challenge. Elyse earned her M.A. in Applied Mathematics from Claremont Graduate University and her B.S. in Mathematics from the University of Massachusetts Amherst.*

Dave Kaslow *is Director, Product Data Management at Analytical Graphics, Inc. He has thirty-nine years of experience in both the technical and management aspects of developing ground mission capabilities. He is co-author of "Defining and Developing the Mission Operations System", "Activity Planning", "FireSat" and "Spacecraft Failures and Anomalies" in Cost-Effective Space Mission Operations. He is also the author and co-author of papers for the International Council on Systems Engineering (INCOSE) Annual International Symposiums and for the IEEE Aerospace Conference.*

Grant Soremekun *has a background in engineering design optimization and software development. He holds a B.S. in aerospace engineering and an M.S. in engineering mechanics from Virginia Tech. For the past 9 years, Grant has worked for Phoenix Integration, a software company specializing in process integration, multidisciplinary analysis, and design optimization. During this time, he has fulfilled several roles, including software developer, application engineer, product manager, and is currently serving as the manager for Phoenix Integration western US sales..*

Sara Spangelo *completed her Master's in Aerospace Engineering at the University of Michigan, where she worked on optimizing trajectories for energy-efficient periodic solar-powered UAVs. She has been involved in the GPS and operational scheduling of the Radio Aurora Explorer (RAX) CubeSat Mission from 2009-2012. She is currently pursuing a Ph.D. in Aerospace Engineering at the University of Michigan The focus of her ongoing doctoral work is on developing models, simulators, and deterministic and stochastic optimization algorithms for scheduling small satellites and diverse heterogeneous ground networks towards enhanced communication capacity. She interned at JPL in 2012, where her work focused on integrating diverse simulation environments to enable Model-Based Systems Engineering of small spacecraft.*

Rose Yntema *is the Applications Engineer at InterCAX (www.InterCAX.com) where she applies MBSE techniques to complex systems in areas such as aerospace, energy, defense, and telecommunications. She is actively involved in the development of SysML parametric modeling and simulation software. Yntema earned her M.S. (2012) in Electrical and Computer Engineering from the Georgia Institute of Technology, and Sc.B. (2010) in Electrical Engineering from Brown University.*