# Model Based Systems Engineering on the Europa Mission Concept Study

**Todd J. Bayer, Seung Chung, Bjorn Cole, Brian Cooke, Frank Dekens, Chris Delp, I. Gontijo, Kari Lewis, Mehrdad Moshir, Robert Rasmussen, Dave Wagner**
**Jet Propulsion Laboratory, California Institute of Technology**
**4800 Oak Grove Dr.**
**Pasadena, CA 91109**
**818-354- 5810**
**Todd.J.Bayer@jpl.nasa.gov**

*Abstract*— **At the start of 2011, the proposed Jupiter Europa Orbiter (JEO) mission was staffing up in expectation of becoming an official project later in the year for a launch in 2020. A unique aspect of the pre-project work was a strong emphasis and investment on the foundations of Model-Based Systems Engineering (MBSE). As so often happens in this business, plans changed: NASA's budget and science priorities were released and together fundamentally changed the course of JEO. As a result, it returned to being a study task whose objective is to propose more affordable ways to accomplish the science. As part of this transition, the question arose as to whether it could continue to afford the investment in MBSE. In short, the MBSE infusion has survived and is providing clear value to the study effort. By leveraging the existing infrastructure and a modest additional investment, striking advances in the capture and analysis of designs using MBSE were achieved. In the process, the need to remain relevant in the new environment has brought about a wave of innovation and progress. The effort has reaffirmed the importance of architecting. It has successfully harnessed the synergistic relationship of architecting to system modeling. We have found that MBSE can provide greater agility than traditional methods. We have also found that a diverse 'ecosystem' of modeling tools and languages (SysML, Mathematica, even Excel) is not only viable, but an important enabler of agility and adaptability. This paper will describe the successful application of MBSE in the dynamic environment of early mission formulation, the significant results produced and lessons learned in the process.**

TABLE OF CONTENTS

1. INTRODUCTION .....1
2. MBSE APPROACH .....1
3. SUMMARY OF MODELING RESULTS .....4
4. ARCHITECTURE DESCRIPTION.....4
5. MISSION DOMAIN VIEW.....5
6. SYSTEM DECOMPOSITION VIEWS .....6
7. SCENARIOS .....8
8. ANALYSES AND REPORT VIEWS.....9
9. INFRASTRUCTURE, TOOLING, CONFIGURATION MANAGEMENT .....12
10. LESSONS LEARNED .....13
11. FUTURE WORK .....16
12. CONCLUSION .....16
13. REFERENCES .....17
14. BIOGRAPHIES.....17

978-1-4577-0557-1/12/$26.00 ©2012 IEEE

## 1. INTRODUCTION

*Proposed Europa Mission*

Jupiter is the archetype for the giant planets of our solar system and for the numerous giant planets now known to orbit other stars. Jupiter's diverse Galilean satellites—three of which are believed to harbor internal oceans—are central to understanding the habitability of icy worlds.

Europa and Ganymede are believed to be internally active and harbor internal salt-water oceans. If extrasolar planetary systems are analogous, then icy satellites could be the most common habitats in the universe—probably much more abundant than Earth-like environments which require very specialized conditions to permit surface oceans.

In 1995 Galileo - the spacecraft - arrived at Jupiter to conduct its follow-up on the key Voyager discoveries, especially at Europa. Galileo made many discoveries in the Jovian system including an induced magnetic field at Europa, which is considered strong evidence of a near-surface global ocean. The Juno mission launched in August 2011 will focus on Jupiter's deep interior and magnetosphere but will not address key science questions for the Galilean satellites and the integrated Jovian system. Thus, a new mission to Jupiter and its satellites was proposed to address top priority scientific questions.[1]

*Study Team Organization and Environment*

The formulation team was assembled in the fall of 2010. Most members were systems engineers and the level of involvement of many was at part time. A unique aspect of the pre-project work was a strong emphasis and investment in laying the foundations of integrated model-centric engineering. In what follows we discuss the approach taken to employing Model-Based Systems Engineering (MBSE) in this endeavor.

## 2. MBSE APPROACH

*Starting Points*

As the architecting effort and tooling got under way, the foundations of a SysML (Systems Modeling Language)

MBSE environment to support the project team became necessary. It was clear that these foundations would take time to put in place, so it was important to choose an achievable target availability date. The resulting plan called for the MBSE environment to be in place and ready to support the project team as it entered Phase A, which was planned to be a year in the future.

Prior to this time significant foundations had been developed by JPL's institutionally-funded Integrated Model-Centric Engineering (IMCE) initiative. The IMCE Concept of Operations [2],[3] explored key scenarios and use cases for how flight projects would actually use MBSE in their execution. This work allowed the modeling team to start off in the right direction and avoid many pitfalls and dead ends.

Other institutional activities benefited this study, including the deployment of a collaborative SysML tool environment (MagicDraw from 'NoMagic Inc.' and TeamWork), the development of ontologies specific to our business of robotic space exploration[4] and the incorporation of these ontologies into the SysML tools via the MagicDraw plug-in support. Finally, institutionally-funded training and support had produced a group of several dozen competent practitioners of MBSE on which the project could draw, as well as a curriculum covering MBSE, SysML and MagicDraw which was adapted to the specific needs of the modeling team.

*Architecting*

Architecting is a broad term for the meandering path we take from system objectives to design. It is an exploration of concepts, addressing a typically intricate and evolving variety of concerns and aspiring eventually to converge upon an architecture that is harmonious from all points of view. Architecture itself is not design, but rather defines a space of designs, including both the variations that are allowed as well as allowance for variation that must be preserved.

While broad, interwoven and creative in their scope, the established product of such systems engineering endeavors nonetheless falls narrowly in a set of stilted requirements, each dutifully accompanied by its own isolated statement of rationale. While this imperative, staccato style is necessary for a clear and verifiable direction to designers, it largely belies the complex reasoning involved in arriving at a balanced architecture. The perfunctory representation of requirements "flow-down" does not convey a comprehensive understanding of how the integrated architecture is intended to meet its objectives. Assorted supplemental information that might offer such insight tends to be treated less formally, posing the likelihood of a fragmented and cursory architecting effort.

In any event, requirements alone are not ordinarily expected to address the full range of obligations and constraints under which a project works — hence the role of policies, principles, procedures, budgets, reviews, and so on. In such a disjoint collection of architectural artifacts lies the danger that the resulting design will fall substantially short in the eyes of some stakeholders. Failure to consider all the angles and instill this architectural thinking in order to preserve the relationships among concerns, concepts and design is a recurring theme in criticisms of systems engineering. Omissions, misinterpretations, inconsistencies, and poor foresight regularly lead to problems later in development or during operations. Without strong guidance from a sound architecture, attention can be drawn unwittingly to a blinkered point design, where the big picture gets lost.

Wanting to avoid such pitfalls, this mission concept study decided to cast the architecting effort in a more structured form, so that a clear line of reasoning could be followed from each stakeholder's concern to system requirements and other project directives. Specifically, the decision was made to write the rationale *first*, in coherent narratives tailored to each concern. Requirements, policies, and so on would *then flow from* this unified rationale in a manner somewhat reminiscent of the "literate programming" approach to software development.[5] The imposed architectural structure would then enable these narratives to be carefully interwoven for consistency and balance.

Such an imposition of structure immediately suggests modeling formalism. Indeed, there was strong motivation to align the architecting effort and supporting tools as closely as possible to larger interests in system level modeling in this study. After all, a structured architecture description is effectively a model, fitting neatly with the resulting requirements into the "Four Pillars" view of system modeling advocated by the Object Modeling Group in its approach to SysML.[6]

The need to launch architecting early, while the modeling effort gathered momentum, argued for a simplified approach: one that could be learned easily and adapted quickly into available tools. Thus, it was important to stick with mostly familiar systems engineering concepts, readily adaptable to comfortable forms of expression, but with just enough structure to accomplish the primary ends of the approach. Care was also necessary to ensure both that near term links to the growing system modeling activity would be straightforward, and that architecting information could migrate easily into the more involved and comprehensive system model, once it had matured. Nothing off the shelf quite fit the bill, so a modest architecting framework (described below) was developed and subsequently adapted using an open source web development tool for collaborative databases. The tool resulting from this development is referred to as the Architecture Framework Tool (AFT).

A further innovation of interest was a more disciplined approach to the architecting effort, which sometimes can seem a bit improvised. Commitment to structure in architecting artifacts has the additional advantage of

instigating similar diligence in other aspects of systems engineering, permitting for instance the productive use of workflow tools. It also encourages an iterative, incremental approach to development, establishing a regular and reviewable cadence to the effort (3-4 month cycles).

Up to the present stage in this concept study, the notions of architecture framework, collaboration tools, and staged development have worked reasonably well. An associated cultural shift at the institutional level is the more daunting hurdle, but buy-in from both project and line management can have the role of the essential catalyst.

*The Dynamic Nature of Formulation*

In March 2011 the President's budget and NASA's science priorities were released and together fundamentally changed the course of this formulation effort, turning it back into a study task whose objective is to propose more affordable ways to accomplish the science at the Jovian system. This is a common and even typical, characteristic of early mission formulation. Stakeholders, priorities and funding all can and often do change and it is essential that the formulation team be agile in responding to a changing environment. Likewise, agility was important in tailoring the MBSE infusion effort to adapt to the evolving needs and resources.

In this particular case the formulation team came up with an innovative proposal to split the original JEO mission into two independent elements. Each mission element would have approximately half of the original science and would be a standalone mission, launched independently. The splitting was done such that each element could obtain sufficient science to justify itself, so that if only one of them was built it could still be successful scientifically. Keeping the science objectives as the guiding principle, it was decided to have one of the mission elements in low orbit around Europa, carrying the science instruments appropriate to that platform. The other element naturally evolved into a Jovian orbiter with a trajectory that includes multiple low altitude flybys of Europa, having a payload complement amenable to such an observing scenario. Each of these elements by itself is expected to be significantly less expensive than the JEO orbiter.

Later on in the study, at the direction of the sponsor, a third mission element was added. The new element has the objective of delivering a science package to the surface of Europa.

*Modeling Plan*

In order to inform the decision on how to adapt the MBSE infusion to the changing circumstances, a Modeling Plan was created and some of its key elements are summarized below.

The modeling plan has an important difference from the original infusion plans in that, to be valuable to the study, engineering products would need to be produced sooner, even if they were more modest in scope. Also, the new plan takes note of the significant enabling infrastructure already in place due to previous investments by the JEO Pre-Project and by IMCE, the institutional initiative.

The System Model to be created would have the scope and contents required for answering the more immediate system questions only, and no effort would be made to make it more complete for its own sake. The first modeling task specified in the plan was capturing the top level system description in SysML Block Definition and Internal Block Diagrams (BDD's and IBD's respectively). Adding simple analyses to the descriptive models was the second step.

The modeling plan also emphasizes that "System Model" does not imply a single representation and specifically it does not imply that a model has to be in SysML to be considered part of the "System model". The System Model should consist of a federation of models, each in its best representation and with the most appropriate owner, so that it provides the needed level of integration to do the required analyses and not more. Despite that, it was recognized that a certain level of tool standardization would be beneficial to the project and the AFT and SysML/MagicDraw were chosen as the two main tools to create descriptive models of the mission concepts, as they are compatible by design through their use of a common set of terms.

The AFT was to serve as the central Content Management System, capturing the full thread of the architectural description narrative through a combination of original material created in it and links to information created and stored externally. This enabled the AFT to be used to produce most of the actual study report products.

SysML/MagicDraw would be used for the more graphical/analytical tasks to show how events are sequenced, how elements are related, how relationships are computed and where necessary, how conceptual elements map to realizational or deployment instances.

For analytical models a variety of tools are available, among others Excel, Mathematica, Maple, Satellite Tool Kit (STK), and MagicDraw, not to mention the large set of existing domain modeling tools. In the interest of reviewability, preference was given to models that can be expressed in symbolic math (using tools such as Mathematica and Maple, rather than Excel where possible) and which are graphical and have semantic meaning (SysML rather than Powerpoint or Visio).

In this initial formulation study, the key methodologies were: Architecture Development and Modeling; Design Space Exploration / Trade Studies; and Cost estimating. As the external environment evolved, adaptation became necessary, by tuning the rigor and depth of the application of these methodologies.

Model Verification and Validation is critical for a credible mission concept, but here again the scope the effort was

tailored to be consistent with needs and resources available in this study phase. Existing domain models were assumed to have been validated by their home organizations. Models created for this study were validated by review with team leadership, to make sure the right model had been created. Verification followed a more informal peer review process, to check for correctness. Limited automated verification was introduced where it was deemed practical.

Another interesting aspect of the modeling plan was the general principle that the models, and therefore the modeling team, must contribute to the larger systems engineering effort. The models were intended to become the authoritative engineering artifacts. In addition the modelers, being also capable systems engineers, were assigned other engineering tasks beyond those involving modeling.

This approach did not require that all team members actively engage with AFT or with SysML. Instead, each modeler was allowed to use the tools he was most comfortable with and the work was divided up accordingly. This plan was approved by the project in late May 2011 and an intensive and focused modeling activity got underway immediately.

## 3. SUMMARY OF MODELING RESULTS

By leveraging the existing infrastructure and a modest additional investment, striking advances in the capture and analysis of designs using MBSE were achieved. The modeling artifacts became the single source of authoritative information, once the capturing phase was completed. Another breakthrough happened with the realization that after capture, more time was being spent using the models to do engineering than doing modeling work itself. In the following sub-sections multiple areas in which progress has been made are described. Two of the high level highlights are: SysML models of all three mission concept elements have been created consisting of physical decomposition, system and subsystem block diagrams and mass reports which are automatically generated and served via web pages. At the same time, the mission conceptual architecture description is hosted in the AFT and will be used to produce most of the final report for this study.

As a guide to the reader, the full list of modeling results to be discussed is included below:

- Architecture Description
- Mission Domain View
- System Decomposition Views
    - Flight System Deployment
    - Work Breakdown
- Scenarios
    - System Integration and Planetary Protection
- Analyses and Report Views
    - Equipment List and Mass Margin
    - Power Margin and Energy Balance
    - Data Balance
    - Radiated Equipment Lifetime and Margin
    - Science Margin
    - Cost Estimation
    - Sizing Analysis
    - Automated Report Generation and Web Publishing
- Infrastructure, Tooling and Configuration Management

## 4. ARCHITECTURE DESCRIPTION

The architecting framework used in this study was inspired largely by IEEE 1471-2000 (ISO/IEC 42010)[7] — a reasonable model, considering the software-intensive nature of modern space developments. Balancing hardware, software and operational aspects of description in customary terms for systems engineers suggested only a few basic extensions to ensure that the full architecting story could be told. The result appears in the diagram in Fig. (01), which is described in some detail below.

Blocks in the diagram define categories (metaclasses) of items requiring exposition in the architecture description. Accompanying each category is a template (not shown) specifying the sorts of information required (the attributes of metaclasses) for each member of that category. Members are then linked in a variety of ways, according to the relationships described by connecting lines in the diagram.

Stakeholders and their Concerns are the drivers for everything else in the architecture, i.e., they can be considered the 'entrance points' to explore the framework in Fig. (01). Among the information required are engagement plans for Stakeholders and success criteria for Concerns. Views describe the architecture responding to these Concerns, their form, approach, and so on. In addition, views draw together a variety of other information for consideration from a particular Viewpoint, often tailored to meet a particular Stakeholder's needs. Especially important are the Trades explored in arriving at the results reported in Views, including any Principles of note that may have been invoked in their outcome.

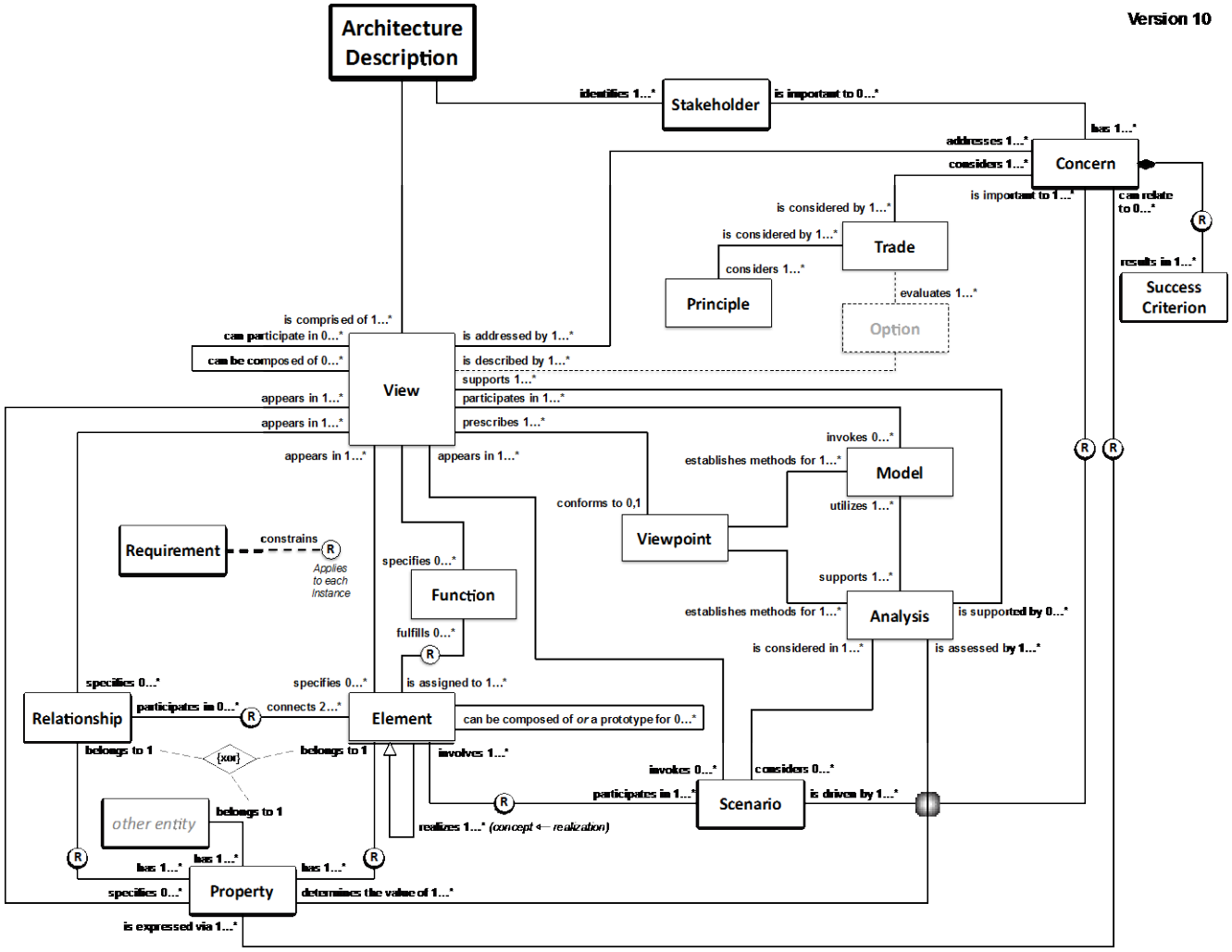


**Figure (01) - The Architecture Framework Tool (AFT) for Architecture Description.**

The system itself, plus aspects of the environment and supporting services with which it interacts, are described in assorted compositional or prototypical hierarchies of Elements. An Element is essentially any item in the decomposition of the system to which Functions or Properties can be assigned. These may be the Elements of architectural concepts, described in comparatively abstract form, or they may be the actual Elements that realize these concepts, given allocations. Interfaces or other interactions between Elements are described in Relationships. (A block diagram would thus be one way to describe a set of Elements and their Relationships.) Both technical and programmatic structures are captured in these hierarchies. By supporting a variety of conceptual and realizational hierarchies, the tyranny of dominant decomposition is avoided in favor of a far more insightful separation of concerns with exposition of the resulting relationships across domains.

Models, the Analyses performed on them, and the Scenarios that drive them round out the description. As with each of the other categories, templates of content plus associated guidelines ensure systematic coverage when they are applied methodically.

The framework above is defined in UML (Unified Modeling Language)[8] and has been integrated into the OWL ontology (Web Ontology Language)[9] guiding the larger modeling effort. The AFT has now been in use for nearly a year, validating for the most part the basic ideas behind the approach. Subsequent development will take the lessons learned from this activity to inform the eventual absorption of architecting into the overall modeling enterprise.

*Ontologies*

As a result of the accelerated pace in this study, some ontologies that had not yet been developed by IMCE were needed and had to be developed in this effort. As a result they are now being integrated into the IMCE ontologies for institutional use by other modeling groups.

## 5. MISSION DOMAIN VIEW

This view is based on a viewpoint borrowed from the "Object Oriented Systems Engineering Method"[10] that sets

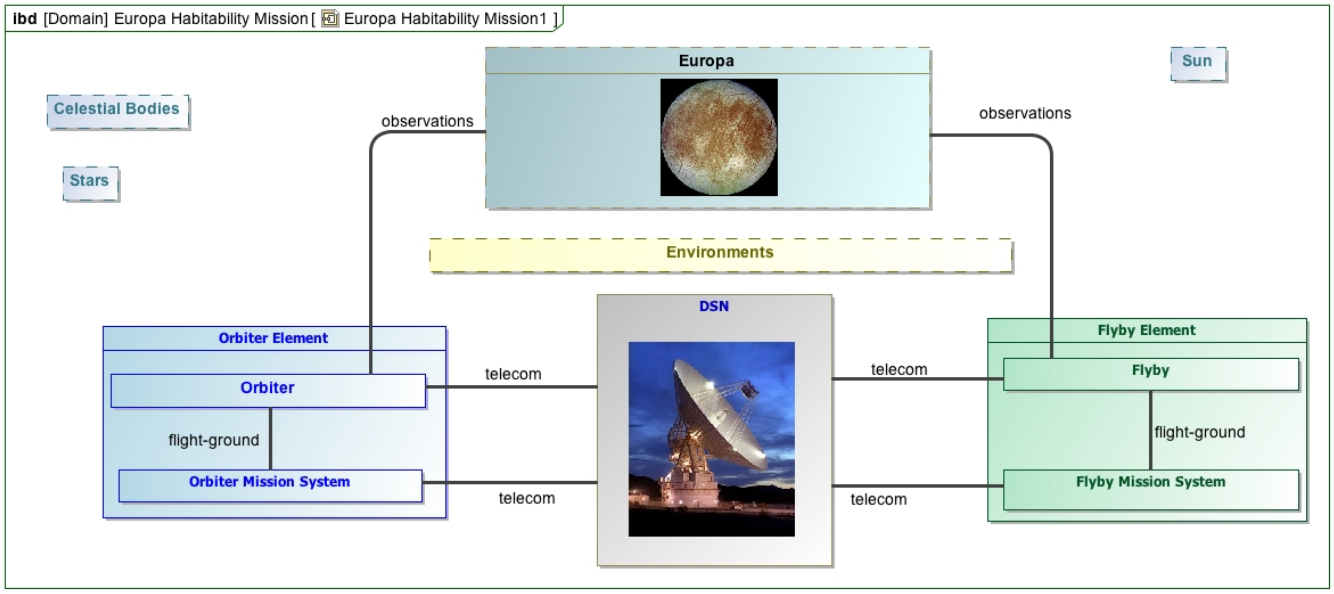


**Figure (02) - Mission domain for the Europa Habitability Mission study concept.**

the broadest context for the "system under study" and the "environment" or anything else the modeler considers to be in the model scope. For instance the environment, the celestial bodies that might have an influence on the success of the mission and the interplanetary telecommunications infrastructure are some of the items included in the mission domain. In this conceptual study, elements such as an orbiter or flyby element are the "system under study" and their interfaces need to be included in the model as well. Figure (02), a SysML Internal Block Diagram (IBD), is a possible description of the mission domain for a Europa conceptual study. It shows graphically the items and interfaces that are in scope and set the context for further modeling.

## 6. SYSTEM DECOMPOSITION VIEWS

Complex systems usually are described by recursive decompositions into subsystems that are then further decomposed in a top-down approach. A single hierarchy normally is used to describe all aspects of a conceptual architecture and the organizations responsible for each subsystem. This often results in oversimplifications and the failure to capture important relationships. Realistically, in many cases several inter-related hierarchies that differ significantly in structure are needed to model properly a complex system and to address different viewpoints.

Two primary types of hierarchies have been defined in this concept study so far. First, deployment hierarchies were modeled, in which composition is defined mainly in terms of modularity and integration order. The second hierarchy conforms closely to the work breakdown structure, which is decomposed into "work packages". Each product or the deployment of products is the responsibility of a particular work package. Work packages in turn are typically assigned to institutional (or business) entities, which are also organized hierarchically. The business hierarchy has not been modeled yet and is not part of this study. (Also to be modeled in the future is the functional decomposition.) The important point to notice is that neither the work breakdown hierarchy nor the institutional/business hierarchy matches perfectly any of the deployment hierarchies. Therefore several types of hierarchies must be modeled separately and then joined with appropriate relationships to better capture the architecture.

In order to have a glimpse of the complexity involved in the hierarchical decompositions, consider the properties of mass and power which are defined for hardware product types. Particular instances of those products that appear in a deployment will have mass and power values assigned. In addition, totals for subsystem deployments are typically allocated to work packages and these are then assigned to

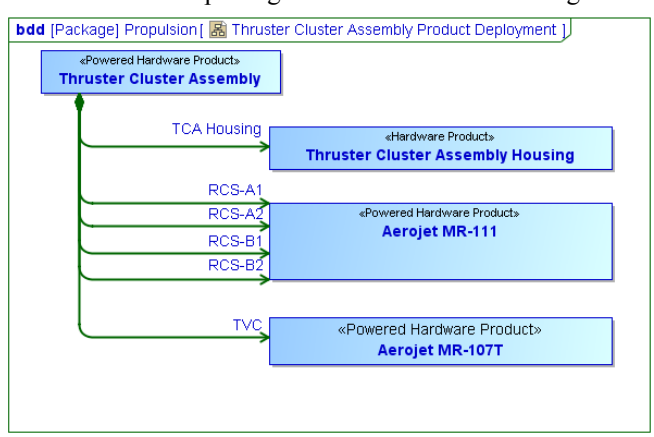


**Figure (03) - An excerpt of a generic system-level deployment diagram.**

individual business/institutional entities so that they can be developed and the accountability for mass/power estimate accuracy can be tracked. Cost is another interesting property, with its complex allocation to work packages and institution. By modeling these three hierarchies (deployment, work package and business/institutional) separately, it is possible to address the inter-related issues of specification, allocation, deployment, and other complex questions without duplicating information and running into problems of keeping them synchronized.

*Flight System Deployment*

A major part of the project architect's vision for architectural capture was the idea of product deployment. In terms of the architecture framework (Figure 01), this is a view showing the physical decomposition, in this case of the flight system. Elements (products) are composed into hierarchical assemblies which have relationships (interfaces) defined between them and properties (mass, power). Rather than talk about subsystems, assemblies, parts, or modules, the system is described by a simple composition hierarchy. Products (e.g., propellant tanks, electronic cards, primary structure elements) are defined by their specification, with the understanding that often they are also made of other products. The connecting of atomic components to make a composite is called a deployment in this hierarchy. Many deployments are possible; for example a star tracker may be deployed to the flight system, or it may be deployed to a test bed or to an inventory for spare parts. Even the flight system is also deployed in various configurations during integration and test, which are different from the flight deployment. Some items show up earlier, while others require substitutes for certain tests. Additionally, there are different testbeds with different needs and support equipment will be a required part of many test deployments. Extender cables, breakout boxes or other non-flight variants may be needed as well. Also, there may be different deployments in flight (as in a staged system) that require separate descriptions. In all cases, variation from one deployment to the next is in the complement of components involved, the interfaces among the ones represented, and the functions assigned to that configuration. These are defined by specification, not by instance. The usual subsystem hierarchies, on the other hand, are related more to assignment of development and delivery responsibility for product instances.

This concept study developed deployment hierarchies for the flyby, orbiter, and lander elements of the flight system, but test beds and sparing have yet to be considered in the model. These deployments are presented to the engineering team primarily as traditional block diagrams, with interconnections captured in SysML IBDs. These deployments allow the team to discuss the collection of electronics into a single box or separate boxes, functional vs. block redundancy and the flow of integration for example. An excerpt of a system-level deployment diagram is shown in Fig. (03). In the diagram below, the text above the arrows is the name of a given deployment. For example, the Thruster Cluster Assembly has four Aerojet MR-111 class thrusters deployed into it. The positions of these thrusters are named "RCS-A1," "RCS-A2," etc.

*Payloads*

For each of the mission elements there is an instrument suite, referred to as the payload. The modeling team found it convenient to set up a separate module that contains all the instruments needed by the mission elements as a library of instruments to pick and choose from. This allows different mission elements to use common instruments or for an element to use multiple copies of any particular instrument. It also allows for easily moving instruments from one element to another.

The model decomposition of each instrument was chosen in part due to the way some instruments are de-composed into a front-end sensor and back-end electronics, but also due to cost. The cost modeling caused an additional splitting of the sensor and its electronics from the shielding around them as shown in Fig. (04). Back-end electronics shielding is handled by the Command and Data Handling (C&DH) sub-system, which will have the responsibility of shielding the entire electronics. For institutional reasons instruments are costed using the NASA Institutional Cost Model (NICM) while shielding is costed using PRICE-H or SEER - two standard cost models from the aerospace industry.

Each instrument is modeled in a BDD which shows its physical decomposition into components. An example of the Laser Altimeter characterization diagram is shown in Figure (04), which is constructed from table 4.2-1 in the "Jupiter Europa Orbiter Mission Study 2008: Final Report" publicly available in the internet.[11] For each of the lower level elements in the decomposition, relevant parameters are defined, such as mass Current Best Estimate (CBE), mass allocation, contingency, mass margin, etc. Analogous parameters are defined for power. Scripts were then created to traverse the decomposition structure and obtain mass sub-totals for subsystems and the total mass for the instrument. It became very important to differentiate which mass values are calculated and updated by the script from those that are filled in by the modelers. A flag 'IsDerived' was placed on all mass properties that the script is allowed to update, thus allowing the automated script to perform the mass roll-up at any time on demand. This is a key property of our system model: it was built to accommodate behavior description & analysis. We emphasize this point, as it highlights the value of avoiding the "tyranny of rigid hierarchy".

As science scenarios are developed, different instruments will be in "On" or "Off" states at different times. A generic "State" parameter model was created to handle this. At different times in the mission, higher-level scenarios can then track in which state each instrument is. By doing this, the hardware-decomposition part of the model can accommodate the behavioral description and analysis part of the model. The same notion is used for the Mass, even though the mass of instruments does not change. This is useful at the system level, as it allows the assignment of

different Mass "states" for propellant at different times, such as Launch, Jupiter and Europa Orbit Insertions.

*Work Breakdown*

This is a view showing the relationship between elements (products) and other elements (work packages). The relationship is "supplies" as in "work package supplies product." It is in this view that the contents of a "subsystem" are defined.

## 7. SCENARIOS

*System Integration and Planetary Protection*

A major challenge of this concept study is the need to meet the strict planetary protection requirements inherent in orbiting Europa. Because the putative Europan ocean could potentially support life, the spacecraft must be effectively sterile inside and out, when it enters into Europan orbit. The main method to meet this sterility requirement is to bake out the spacecraft at a high temperature for a specific length of time, killing any contaminating organism. One of the challenges is to determine at what point in the integration flow to perform this bake out sterilization and to properly plan and maintain sterility post bake, prior to Europan Orbit Insertion. It is important to understand the sensitivities to cost, schedule, and spore count of the different bake out options available during the integration and test activities.

The modeling team captured the integration and test storyboard in SysML and assigned attributes to enable the calculation of cost, schedule and spore count of the spacecraft. Although more work is required for a fully functional implementation, the plan is to create the functionality to manipulate activities in the integration flow and determine the deltas to the cost, schedule, and spore count. This would allow the optimization of the integration and test flow, while meeting the spore count restrictions. A small excerpt of a comprehensive SysML Activity diagram that models system integration and test is shown in Fig. (05).

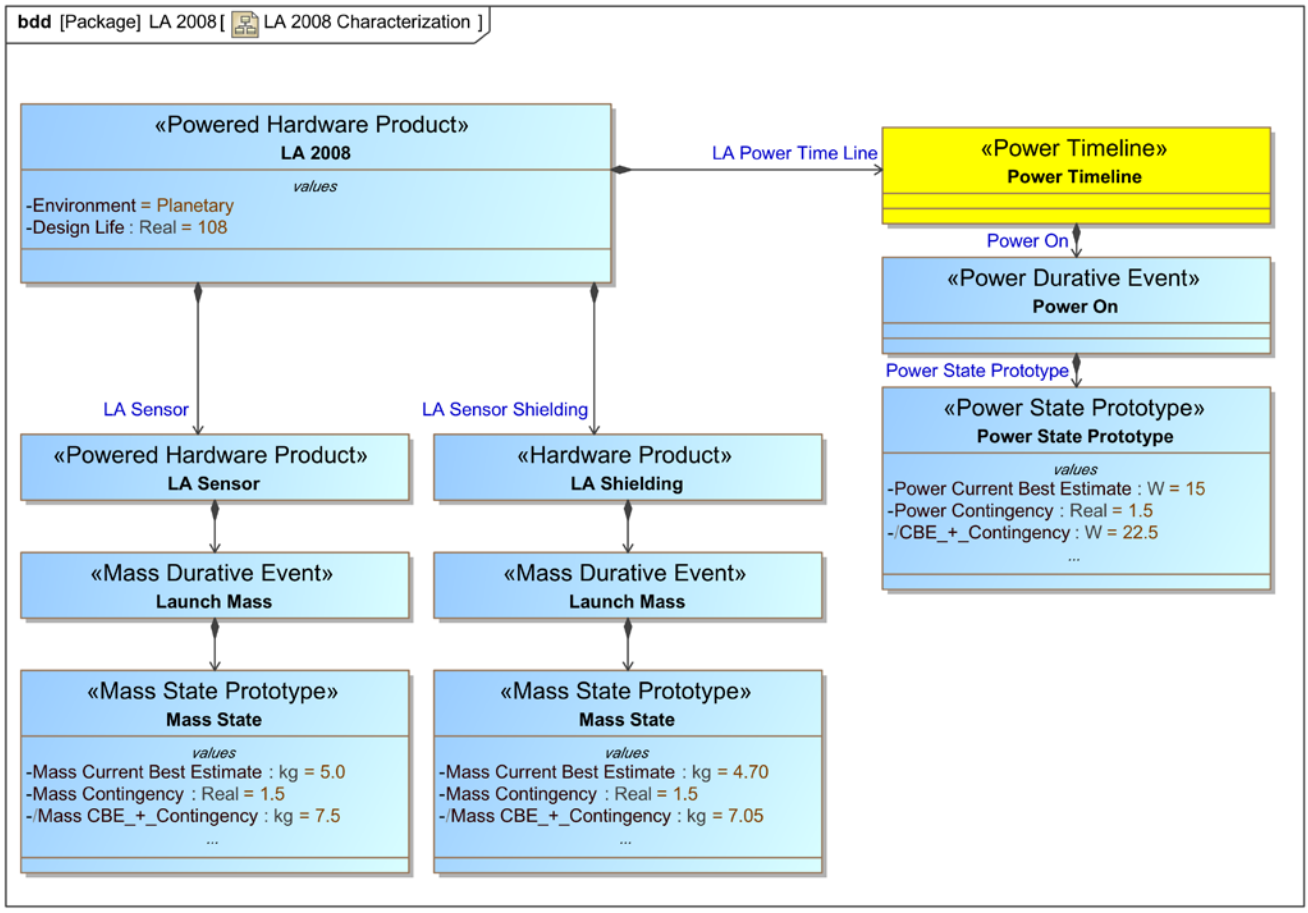


**Figure (04): Characterization of the Laser Altimeter, which is part of the Orbiter Payload**

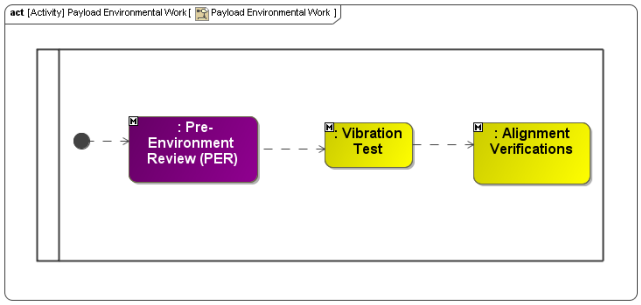


**Figure (05) - Small excerpt of a comprehensive activity diagram describing system integration & test.**

## 8. ANALYSES AND REPORT VIEWS

Analyses such as the computation of technical margins are crucial for verifying and validating a design against the mission objectives. Technical margins measure the difference of what is predicted and what is required throughout the lifetime of the mission. Generally they include mass, power, energy, DeltaV and data margins. In addition, due to the harsh radiation environment of the Jovian System, radiated equipment lifetime margin becomes crucial as well. Other analyses that are fundamental to science mission design are science margin, cost estimation and sizing of the flight system. In terms of the architecture framework (Figure 01), an analysis uses a model to form a view which conforms to a viewpoint. For example, the Mass Margin Analysis described below, calculates flight system mass margin using the elements, relationships and properties in the flight system model, to produce a view called "Equipment List and Mass Margin Report" which conforms to the viewpoint of the same name.

At an early stage of mission design, much of the aforementioned analyses are performed using spreadsheets. While spreadsheets are generally useful, they also have critical drawbacks and provide minimal visibility into the analysis methodology and the fundamental equations governing the analysis. Copying and pasting data into a spreadsheet without recording its provenance is also very common, which complicates future updates. This study concept opted to explore other methods for performing analyses, given that its system architecture is already captured as a SysML model. One approach uses scripts that operate directly on the SysML design specifications, thus reducing error prone and time-consuming data entry and inconsistent information. When the analysis is too complex for scripts that operate within the SysML MagicDraw tool, design specifications are exported to a mathematical engine such as Mathematica where the analysis methodology is explicitly documented and made visible. This requires the use of appropriate transformations, as described in another work presented in this conference.[12]

Documenting and reporting the analyses is as important as the analysis itself. Documentation is a means of communicating the design for the purpose of verifying and validating it against the mission objectives, or to facilitate its inclusion into a broader design. Generally, documentation has been created by copying and pasting the analysis results into a report or a presentation that is distributed as hard or electronic copies. While this approach was also used in this study, it has several limitations. To name a few, the copying and pasting process can involve erroneously choosing wrong parts or pasting into wrong places; the copied information may not be up to date or worse, inconsistent with the final result. Often this documentation process can be time-consuming and represent a bottleneck to an iterative design process. Other documentation approaches used here include DocWeb, which is capable of automatically extracting the design specification and analysis results from SysML directly into html or PDF online documentation. Another approach mentioned briefly before embeds documents directly into the analysis by using Mathematica.

*Equipment List & Mass Margin*

Capturing the "Mass Equipment List" is typically done using tools like Excel by making a list of equipment and their masses. In a System Model the deployed equipment and equipment types are captured as a structure composition of products (type of equipment) and deployments. This basic object oriented structure of deployed 'parts' is connected to the mission domain above, providing a framework for relating a variety of information to the composite structure of the flight system. There are several advantages to this approach; for instance, the properties that characterize the mass of the flight system are modeled complete with units and a temporal context. These mass characterizations then appear on every product in the flight system that has mass.

*Power Margin & Energy Balance*

A flight system generally is either power or energy limited. The present mission concept is most likely power limited and as such, much of the focus has been on analyzing the power margin. The power margin compares the predicted power consumption of the flight system over the mission lifetime to the power availability. Ideally the power margin should be computed for the whole mission lifetime, but to simplify the analysis, only the presumed power-critical events are considered. For each event, the power consumed by the deployed flight system components is compared against the produced power. Currently this analysis is performed using a spreadsheet. In the future, we hope to perform the power and energy margin analyses based on the overall mission scenarios. We expect to perform this analysis directly from the SysML design specifications by either using scripts or by exporting to tools such as Mathematica or Maple. This analysis process is similar to the process used for mass margin analysis.

*Data Balance Margin*

A key analysis in all space missions is the data balance. Remote sensing missions are fundamentally designed to send a payload of instruments to a destination where they can take measurements and return data to scientists on Earth. The data balance model considers the rate at which the system can produce data compared to its ability to move and store this data onboard and to transmit it over available radio links. It is often the case that even simple instruments can produce more data than can be returned due to the limitations of deep space communications. Therefore a model is needed to help explore the trades between observing strategies and instrument properties, transmitter power, antenna size and mass to develop an optimized mission design.

A recurring theme in the analysis performed in this study is the use of more sophisticated tools than the traditional spreadsheets. This modeling effort also began with a spreadsheet model of the data balance. Modifying the spreadsheet quickly to accommodate design and scenario variations was greatly time consuming because of the limited scope of the original model and the fact that so many strategic options were hard-coded into its structure. A re-implementation of the data balance analysis as a parametric model in MagicDraw was considered, but it was later decided that the effort would be better spent first defining and documenting the mathematical relationships. Thus, the data balance model was implemented in a self-documenting Mathematica workbook that could be exported as a report. This expanded model included relationships between orbit geometry, Earth visibility and range, mission duration, and various telecom properties, and was able to present numerous options in the form of parametric plots.

*Radiated Equipment Lifetime and Margin*

A critical aspect of any mission to orbit the Jovian moon Europa is radiation shielding. Previous space missions targeting high radiation environments succeeded by using very large design margins on radiation. The target of the present study is arguably one of the highest radiation dose environments in the solar system and such conservatism cannot be implemented, as it would render the mission concept unfeasible due to the large penalty incurred in radiation shielding. Therefore, more sophisticated radiation analysis to remove some of the conservatism is a topic of great interest to this study. For example, a trade option for increasing the mission lifetime may be to reduce shielding for assemblies which have long lives, while adding shielding around items whose shorter lifetimes are mission limiting. Spot shielding, vaulting and probabilistic lifetime determination need to be studied in depth, in order to reduce the shielding mass while still preserving the spacecraft life after Europan orbit insertion. The Radiated Equipment Lifetime and Margins (RELM) model was developed in an attempt to answer "What if…" questions and scenarios. It takes three main types of input:

a) - The Radiation environment as described by the amount of energy accumulated by typical hardware items versus shielding thickness. This information comes from an analysis of the spacecraft trajectory for the whole mission lifetime and is provided as a set of tables of accumulated radiation dose (TID = Total Ionizing Dose) for the cruise/Jupiter tour and also for the Europa orbit phase.

b) - Radiation Hardness Rating of each item: this comes from component manufacturers, electronic designers and other JPL experts. For each radiation sensitive item (usually electronic components, but also glasses, lenses, etc) the radiation hardness rating (H) is the maximum dose the item can take before failure.

c) - System Architecture: This describes how subsystems are assembled. Chains of subsystems connected in series or in redundant architectures are considered.

The RELM uses Excel as an interface for input and output and Wolfram Mathematica version 8.0 for processing. The results from the calculations are the number of months each subsystem will survive in Europan orbit, with a probability of 95%. These results are plotted in excel to show graphically which subsystem is lifetime-limiting. This not only helps identify subsystems that need further shielding trades, but also help in gauging the chances of survival for extended missions. An example of the RELM output is shown graphically in Fig. (06).

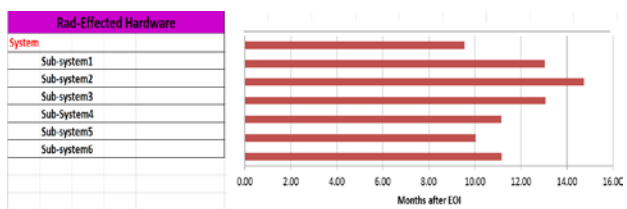


**Figure (06) - Calculated number of months of survival after Europa orbit insertion for the power distribution sub-system.**

*Science Margin*

To assess how well an architecture or point design addresses the science concerns the classical approach has been to calculate the system capability to meet a specific quantitative requirement, such as spatial resolution for example (measured in arc-seconds or meters). This classical approach does not take into account the synergistic interactions that lead to addressing a science concern. Using the classical approach it is not a straightforward matter to make quantifiably defensible trade arguments, but often hidden margins come to the rescue. On a recent astrophysics mission, a concept has been developed that gives quantification of changes in science return vs changes in technical design.[13],[14] These lessons learned have led to the development of a "Science Margin Model" (SMM) for this

mission concept. This enables the stakeholders to assess the impact of technical trades on the science return, with the objective of keeping the Science Margin at a healthy level.

For this study two of the broad science objectives, determining Europa's magnetic induction response and the amplitude/phase of tides, were cast into two SMMs; this was accomplished by working in concert with the science domain experts. A graphical example from one science objective versus two of the mission variables is shown in Fig. (07).

*Cost Estimation and Integration with Cost Models*

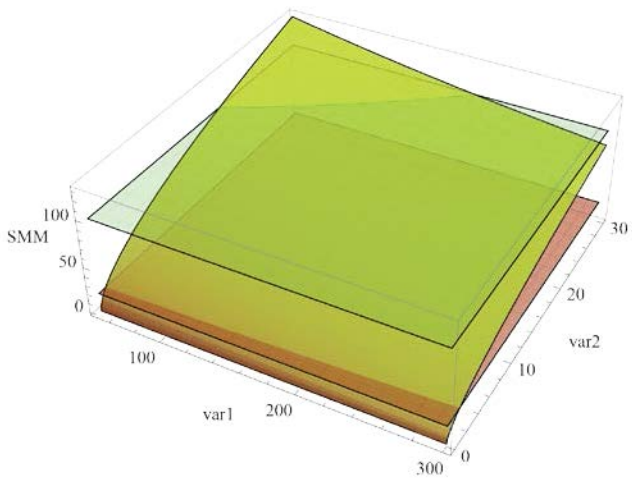


**Figure (07) — Science Margin surface, with baseline (at 100) and threshold planes shown**

There are two reasons for integrating the System Model with the cost models. The first is to have a reliable source of all the needed information that can be exported to commercial cost models for independent estimates. The second, more ambitious reason is to incorporate cost parametrics within a SysML MagicDraw model. This would allow for simultaneous optimization for a large set of parameters that would also include cost.

It is fairly straightforward to capture much of the cost information in the SysML model and report it through tables. Scripts are available that create generic tables of Value-Parameters (such as cost, mass, power) that can be buried several layers deep within the decomposition and might come from different sub-blocks. Several script blocks are in development to represent the information needed for the two commercial cost estimating packages SEER and PRICE-H[15] models. The next step, not yet completed, is to traverse the SysML model and collect the information to populate the cost model blocks. This last step is needed as a fair amount of information needs to be converted to different units for the different models. For example, if one model needs Watts as input, while another needs milli-Watts, then the math needs to be done by a script for now, as our report tables do not have the capability to convert units on the fly.

A more sophisticated aspect of cost modeling would be to integrate the cost model inside the SysML model, perhaps using parametric equations and diagrams. Two paths are planned to pursue this; one will employ the Paramagic plug-in for MagicDraw, while the other will use the Simulation Toolkit. Both methods will be used first to solve for the cost and replicate the NASA Institutional Cost Model Excel spreadsheet as a validation step. Fig. (08) shows an excerpt of a parametric diagram (with equations removed) for the cost model of a laser altimeter.

*Sizing: Parametric Web, Reaction Wheel Sizing*

Systems modeling by itself is a valuable activity as it facilitates a clear and consistent vision of interactions between design components and system-level behavior. The new SysML data structures provide a solid basis for modeling and investigating system behavior and allow more streamlined, shorter cycle updates of systems margins. Both the descriptive and analytical capabilities of a SysML model were exercised in this mission concept. The Query-View-Transformation (QVT) language has been used to feed numerical analyses for a power simulation and inertial properties from the SysML model into a mathematical engine.[12] This was found to be a powerful technique whose value will likely increase as it is better understood by the systems engineering community and streamlined workflows are developed.

The descriptive capability of SysML Parametric diagrams have been used to develop a roadmap for building and/or adapting analyses for the Europa orbiter design. The Parametric diagrams were started by a question posed by the project architect: "Why is 30 days the magic number for the orbiter mission duration?" The diagrams then followed the thread from the parameter of "science mission duration" through all of the impacts on the system – required data collection rates, required return rates and thus sizing of the telecommunications equipment, required power generation levels, enhanced radiation shielding, and so on, for a complete thread of calculation . Making this thread explicit and systemic worked against the natural inclination of disciplinary engineers to work their particular silo with assumptions handed to them by project-level engineers. It also recast the analysis into a question-centric perspective rather than a discipline-centric one.

*Automated Report Generation and Web Publishing*

The ability to separate the concern of how information about a mission and flight system is captured and the concern of how it is viewed is a key benefit of Model Based Systems Engineering (MBSE). The capability to produce and manage reports from the model is part of realizing this benefit. On the other hand, using the model-authoring tool to access information in the model is not always desirable, particularly when producing and reviewing deliverable products such as architecture descriptions and requirements documents. For this study concept the focus was to produce a report similar to the system block diagrams and mass roll-ups produced by conventional office productivity tools. The product deployment model contained the Mass Equipment List (MEL) and mass estimate information. The semantic rigor of the model facilitates analysis and generation of tables describing this information.[16] These tables are explicitly consistent with the views in the report that depict the flight system block diagrams. The table view provides a concise way to read the MEL that conventional office tools are not capable of. Additionally, generated documents have the benefit of making much more sophisticated use of Internet technology. The project now provides a website where the latest reports are accessible and configuration controlled.

## 9. INFRASTRUCTURE, TOOLING, CONFIGURATION MANAGEMENT

First we discuss the interplay between the two main tools used for capturing the architecture and design of each of the Europa Habitability Mission (EHM) elements. The Architecture Framework Tool was described in Sec. 4 and captures most of the descriptive type of information. Because the tool is completely web based, it lends itself to be more fluid and editable by a larger team. On the other hand, the SysML model is both more graphical and more rigorous. A good example of the distinction between the two tools is as follows. The description of how we plan to take science data once EHM is in orbit around Europa is in the Architectural Framework Tool. The mass and power calculations, and the On/Off states of the instruments during that time are captured in the SysML Model. Thus, the

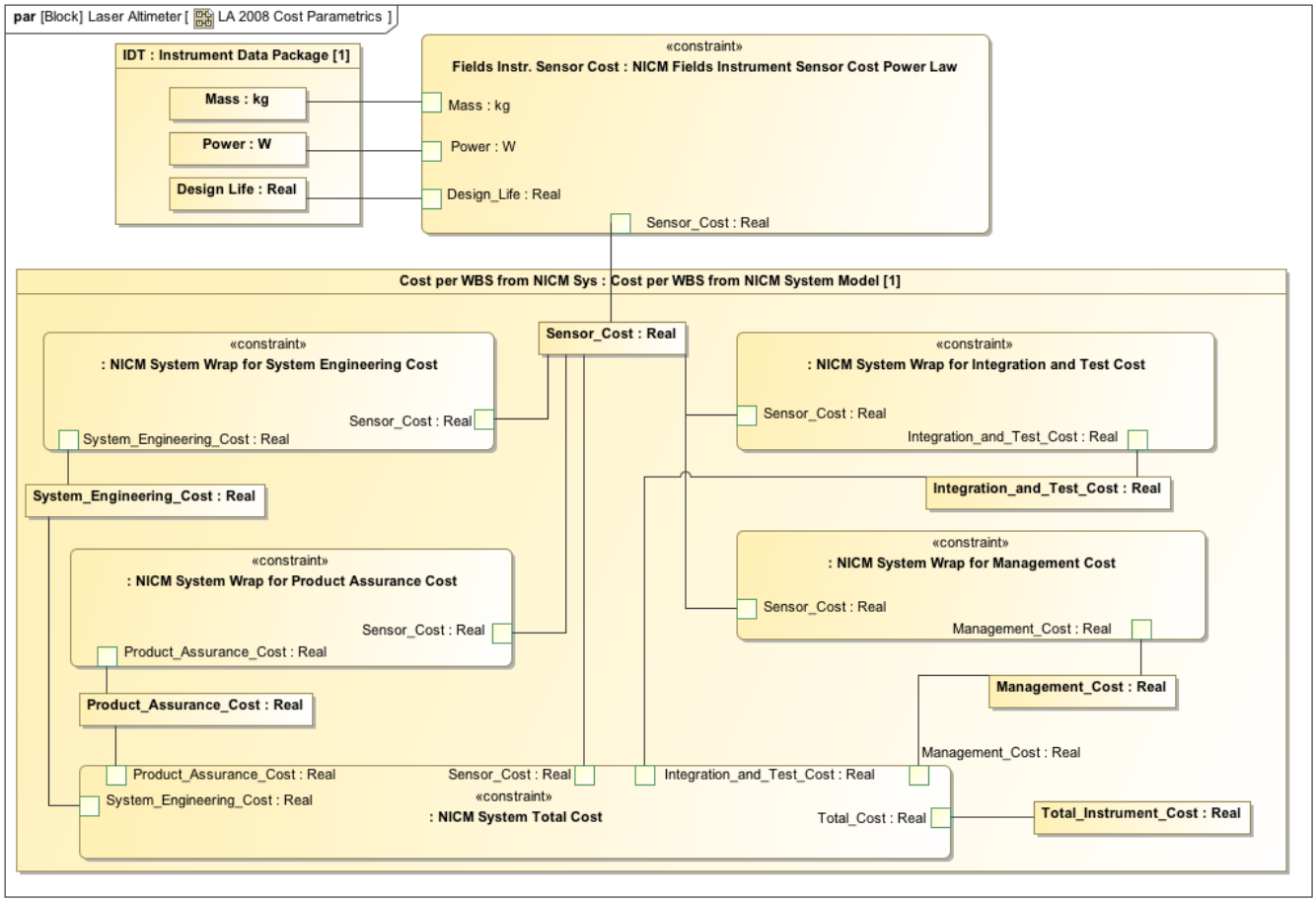


**Figure (08) - Parametric Diagram for the Cost model of the Laser Altimeter (equations not shown). Equations are associated with each of the <<constraint>> blocks in the parametric diagram. These are written as equalities, not assignments. So by setting flags as to which parameter to derive, it is just as easy to solve for the maximum mass given a cost, as it is to solve for a cost given a current mass. The latter is what is typically done in current cost modeling.**

SysML modeling team references the AFT information as guidance on what and how to model the system. There is sufficient structure in AFT to make this mapping fairly straightforward and rigorous, even though the AFT content is less formal.

As mentioned before, the SysML model was built using MagicDraw, with modules stored on a MagicDraw Teamwork server, which handles configuration management (CM), including version control, authentication and locking and unlocking of partial or entire sections of the model. We implemented loose control on the SysML and AFT models, enabled by the small team of frequently interacting people. We let Teamwork track changes in the model, and used the DocWeb report generation tagging capability to mark 'baseline' versions of the model as it developed. The IMCE Concept of Operations exploration of Configuration Management issues was useful as we stood up a lightweight and agile system for our study.

The EHM model is broken into 7 major modules: one for each of the 3 mission elements, an instrument library, a prototypes library, an element library and an architectural library. The latter 3 form the basic building blocks for each of the components, which are then used in the mission elements and instrument library. The architectural module defines the same concepts inside of the SysML tool as those present in the Architectural Framework Tool. Further modules are used for secondary functions, such as document models that produce automated documents, report models for costing, etc.

Several scripts were also developed both in Jython and QVT, which help with automation. The most notable of these is a script to propagate Value-Properties along generalized elements. This allows the generalization of hardware components that have value properties for mass and power created using scripts. Another set of scripts performs the mass roll-ups. These scripts take Value-Properties of the leaf components and sum them to the next layer up (roll them up). This is done recursively, until it reaches the total mass for the mission element.

The remaining sets of scripts are used to produce documents. Smaller script modules access the main model and query it for diagrams and tables. By having a server that automates these scripts, nightly documents are generated from the model and hence remain current. The web-based server allows the scripts to run immediately on demand or at various automated times. It generates both PDF and HTML versions of each document. Each version is kept in the server, so that different versions can be compared later and changes highlighted. Finally, the server has the capability to tag particular versions for subsequent access, such as those used for various reviews.

This infrastructure has greatly increased the speed with which new models can be created. The re-use of the leaf components, value parameters on stereotypes and the generalizations have accelerated work by at least a factor of 3. We have now reached an important threshold, where we spend more time discussing the particulars of the design, than the infrastructure or implementation methods. As we add functionality, we will continue to require building up the infrastructure in order to meet our next level of expectations from the model.

## 10. LESSONS LEARNED

The key lessons learned can be summarized as:

- Investment is crucial
- Unity of leadership is essential
- Early efforts draw from a limited pool of talent
- CM can start modestly
- Team organization matters
- Everyone needs training, but not to the same level
- "Just Do It"
- Keep the focus on engineering products
- Let questions drive coverage and depth
- First description then analysis
- Separate the model from the analysis
- Real examples are powerful

*Investment is Crucial*

When EHM embarked on its scaled-back modeling efforts in May 2011, it had the benefit of several years' worth of investments in foundational infrastructure by IMCE. It also had the benefit of almost a years' worth of application investment by the JEO Pre-Project.

*Unity of Leadership is Essential*

In the first infusions, management support for the effort must to be clear and consistent. Management must be willing to pay the startup costs and to give time for the effort to pay dividends. In addition, the engineering leadership must be reasonably unified in their willingness to work together to figure out how to do this. Both of these things were true on JEO and were essential to finding success even after the transition to the smaller, leaner EHM phase.

*Early Efforts Draw on a Limited Pool of Talent*

Similarly to the "Unity of Leadership" above, the first infusions will not have the benefit of an engineering pool with ubiquitous modeling skills. On JEO we found that the best way to get started on the right path was simply to hire as many of the existing cadre of skilled MBSE practitioners as we could afford. At that time there were no more than ten or so truly expert system level modelers versed in SysML, and we hired nearly all of them.

*Configuration Management Can Start Modestly*

Initial exploration of Flight Project Model CM was done in the IMCE Concept of Operations. This knowledge was

useful as we stood up a lightweight CM system for the Europa Studies.

Model organization: modules and packages were structured with collaboration in mind, and we emphasized single owner packages in topically-defined modules

In early formulation, we have found agility to be more important than tight access control. The full team has read access, and the core modeling team plus key systems engineering leads have write access to the models. We have assigned individuals responsibility for individual packages, and put everyone else on the honor system to not write into this model without coordinating with the "owner"

Versioning. The Teamwork server tracks changes to model elements. DocWeb reports capture snapshots of the full model and resource reports, and reviewed and baselined versions are tagged as such in DocWeb

Quality Control: Scripts are beginning to perform some rudimentary model validation. A hand calculation is used before report release as final correctness check because the scripts are not completely bug-free yet, and because in general a hand check is good engineering practice.

We plan to evolve and tighten the controls as necessary as the effort moves into later phases.

*Team Organization Matters*

It is important to consider carefully the use cases for how modeling is done. Before JEO, most of the MBSE pilots that had been tried at JPL were small scale and often grass-roots. Systems Engineers on these small teams tended to become the primary modelers and therefore tended to be the primary custodian of the single source of authoritative information as well as being the most expert SysML users. The IMCE Concept of Operations looked at what a large scale flight

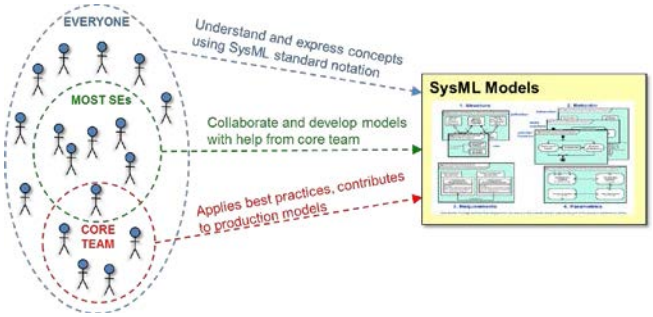


**Figure (09) — JEO modeling team organization.**

project would need to do differently. JEO, as a fledgling project and as the first full, top-down infusion of MBSE at JPL, had the benefit of that work as we worked out what the model usage patterns should be. The pattern we have found that works well is a three-tiered one involving a small set of core modelers within a larger set of modeling-savvy systems engineers, within a larger set of all project personnel. While we have found that descriptive modeling can be done by almost anyone with the basic training, the additional rigor and consistency needed for quantitative analysis requires us to designate a smaller team of people who are modeling experts and who can apply best practice to the official configuration managed project models. Presently we have a core modeling team of a half dozen or so, within a larger team of 20 or so engineers. See figure 09.

The experienced systems engineers provided guidance to keep the modeling focused on providing useful information, as well as mentoring of the core modelers who tended to be more junior. Frequent (daily) interactions were key to getting useful products: we were pathfinding so we had to stay very closely in touch.

As important as it is to have a core modeling team, it is just as important to avoid fencing them off from the rest of the project. If the models are to be useful to the project, the project must understand and interact with the modeling team regularly, and likewise the modelers must be engaged in the larger engineering effort. Modelers who are also capable systems engineers will naturally employ their modeling skills to deliver their engineering products in a model based way.

## *Everyone Needs Training, but to Different Levels*

In the usage model shown in Fig. (09) it is clear that all three groups need to receive training commensurate with their level of interaction with the models. Different levels of modeling familiarity are required, thus resulting in different levels of training. Working with IMCE, we have constructed a set of classes that addresses all three user-type groups. The classes are sequential, each one building on the one before it. We start with the basics: Architecting, AFT, MBSE and SysML familiarization for everyone. The second class is the advanced SysML and Tool-Specific training, for the engineers who will be working with the models (MagicDraw and AFT). Then finally the third class, really a continuing series of special topic sessions, is for those engineers who actually construct, maintain, and analyze the official project models.

*Just Do It*

We've found that the best way to figure out how to apply MBSE is to do it for real: make the commitment to adopt MBSE as the way to produce (at least some subset of) the project products, and then figure out how to accomplish this. This is in contrast to the suggestion sometimes made by skeptics, that a "safer" or "more gradual" approach would be to conduct a "shadow" or "parallel" pilot that allows side-by-side comparison of benefits and drawbacks, including cost.

The problems with this suggestion are significant. First, it can be a challenge to obtain adequate resources to do the job right once, let alone twice. This was especially true of our team as it de-evolved from a pre-project to a concept study team. Conducting a shadow pilot would have taken critical

resources away from an already meager supply. Second, a useful comparison requires some set of metrics. Such metrics have not been developed for a comparison between model-based and traditional systems engineering methods, and we believe doing so would be a significant and costly task in itself. Third, we have found that the pressure to deliver real engineering products has forced us to confront and solve many problems we would likely not have seen in a shadow pilot. Fourth, we think it's asking the wrong question. We believe the move to MBSE is not a question of "whether" but a question of "when" and "how", something a shadow pilot does not help to answer. Finally, we think the answer to the question "Does capturing our designs in an expressive and rigorous language via an integrated, durable, analyzable model give us better engineering products?" is self-evident. Of course it does. And "does that help avoid risk and cost downstream?" Of course it will. "Measure twice; cut once" as the adage goes.

So how does a project control infusion cost and risk without this comparative knowledge? Do it by carefully scoping the infusion: start small, but always start on a real product. These ideas are further discussed below.

*Keep the Focus on Engineering Products*

Keeping focused on real engineering deliverables is important to avoid the pitfall of delivering a modeling solution everyone thinks is finished but which doesn't provide the required engineering answers. After all, the engineering deliverables are the whole point of the exercise. Our early attempts at "rolling up the mass" for the mass margin report showed this in stark relief: getting the numbers to add up, make sense, and be reliable turned out to require significantly more modeling and scripting than expected.

*Answering Questions Should Drive Coverage and Depth*

A common misconception of MBSE is that in order for the model to be useful it must describe everything and it must describe it to a fine level of detail. This misconception needs to be corrected for an infusion to succeed, because otherwise resources will run out long before the job is complete. A key principle we have followed is to model only as far as we need to answer the question at hand. Assuming this is done on an infrastructure of common languages and tools (see "Investment" above), then the model can grow over time, as necessary, and each new model element adds synergistically to the body of work.

In the early formulation phase in which we find ourselves there is a curious duality. On the one hand, the key work in early formulation centers around conceptual thinking. The spacecraft we propose are mere sketches, and a critical function of models is to describe the design space generously, in which the concept can evolve and take shape. On the other hand, we must always show that our concepts are feasible, and one of the ways we do this is build and analyze a 'point design' which we analyze for technical resources, performance, and cost. The models that we build to address these two disparate viewpoints must of necessity be partly conceptual and partly realizational. This should not mislead one into thinking that the space between them must be entirely filled in: it has proven very workable to have some parts of our model be treated as strictly conceptual (e.g., for the parameter web roadmap), and other parts be treated as realizational (e.g., for the mass margin analysis).

*First Description, Then Analysis*

Another common misconception is that models are not really useful until they can be subjected to quantitative analysis. This is simply not the case. Capture and description are powerful and far-reaching first steps. Just describing something in a formal modeling language like SysML immediately improves communications and understanding. The benefits of this would be difficult to overstate.

Additionally, this part of modeling is relatively easy, especially compared to analysis. For the mass margin report, even our modest ambitions were a bit of a stretch the first time because of all the capabilities that needed to be created along the way. But if the first mission element took longer than expected to analyze for mass margin, the second and third ones showed the power of developing reusable methods: they each took a fraction of the time of the first. For the first mission element we captured and analyzed in the model, the EHM Orbiter, the working model plus mass report took approximately two work-months to complete. The subsequent two concepts we modeled (Flyby and Lander) each took about one half work-month each. And,a subsequent change of Flyby design was accomplished in a fraction of *that* time.

So, our advice is to first focus on description, and then implement analyses. A large part of the benefit accrues as soon as people start using the descriptive models, and this gives time and support to allow the more difficult work of analysis to be done.

*Separate the Model from the Analysis*

Looking at Excel, the traditional tool for systems mass analysis, is instructional for understanding why we need better systems modeling tools. Two troublesome characteristics worth mentioning in this context are: as it is commonly used, the model and the analysis are inextricably intertwined; and by the nature of the tool, the model is forced into a form which facilitates the analysis. Together these make the model less intuitive, and drive major duplications of model information (the equipment list is manually duplicated for every different analysis, such as mass margin, power margin, and fault modes effects and criticality analyses).

It is clear that the more the model can be a self-contained internally self-consistent and intuitive description of the concept, the more informative it will be. And, the more the analysis can be separated from the model, the more reusable

it will be. For our mass analysis we have achieved a high degree of separation of the model from the analysis, and as a result we are able to run exactly the same mass analysis script on all three of our mission option models.

The corollary to this is "keep the model aligned with the concept rather than with the analysis". We initially found ourselves adopting modeling patterns which made the analysis scripts easier (drifting back into the Excel mindset). But we soon discovered that in order to further expand and refine these analyses, we would be forced to model in more and more non-intuitive ways. Therefore we discovered, and adopted, the principle that the model should be kept intuitive and aligned with the concept. We are convinced that the extra work required to make the analysis tools work is well worth it in the long run.

*Real Examples are Powerful*

Trying to describe to stakeholders and potential collaborators what MBSE looks and feels like has proven to be rather difficult and not very effective. We have found that many people 'get it' for the first time only when they see an actual example.

The EHM Orbiter Mass model and Margin Report were immediately recognized as higher fidelity work than reports generated by traditional methods. Since parametric cost estimates are based heavily on mass, this is a crucial parameter to estimate accurately. This was also the feature that helped the 'light go on' for several skeptical but open-minded stakeholders.

Finally, projects are where the 'just do it' happens, working on actual products - that's where the applications are really worked out, and that is what feeds back into IMCE for others to use. These first examples discover useful patterns which can be fed back into IMCE for capture, stewardship, and provision to the next users.

## 11. FUTURE WORK

The Europa Study will deliver a Final Report to our sponsor in 2012 and then await further direction. Therefore the focus of the forward work will be on the review and refinement of our engineering concepts and creating the report. Part of the refinement and review is the making of internal and independent cost estimates of these concepts. We had originally planned to construct specialized model output reports to enable this, but to our pleasant surprise we have found that our cost engineers are using the SysML models themselves as the source of the information they need. The one new capability we do plan to complete is the automated generation of as much of the Final Report as possible. The Architecture Framework Tool is being enhanced to allow creation of flexibly structured and formatted reports, so that the AFT can be the single source of truth for the full narrative of the architecture description.

Aside from this, the nature of the remaining modeling work on our team will be more consolidation than expansion. We will focus on solidifying the gains we have made by: refining and documenting the patterns we have found useful; documenting and packaging the analyses we have constructed; and describing the future developments we would undertake should the resources become available.

Because of our institutional partner IMCE, we are confident that the good work done so far on the Europa Study will be captured and made available for use by all future model-based efforts.

## 12. CONCLUSION

Our early progress on JEO/EHM clearly shows both the feasibility and the benefits of adopting MBSE. We have shown that the current state of the art in early formulation can be extended using architecting frameworks, SysML, and symbolic math tools, to produce better formulation products and to begin to bridge the information divide between early formulation and project start.

Using the institutional investment emplaced by IMCE, only a modest additional project investment in MBSE was required to realize significant benefits. Multiple mission options are now captured and described in SysML models, including physical decomposition, interface block diagrams, delivery responsibility, mass and power estimates. This information is reported regularly via web-accessible reports and is the recognized single source of truth for this information. Analyses of mass and power margin, while still performed in Excel, use the SysML model as the source of their data. Other analyses are now performed via well-documented and mathematically rigorous methods: data balance, radiation lifetime, and the first pieces of science margin are calculated via Mathematica notebooks.

Our use of system modeling has contributed directly to the recognized high quality of our mission concept, including: increased stability of the concept description, increased accuracy of the key technical resource estimates such as mass and power, and increased agility in the face of changing sponsor needs and priorities.

If these studies result in an approved mission, then that team will have much more useful information to draw on than the traditional study would have provided: our use of MBSE will pass to the project team a much more durable, rich and extensible body of information from which to start.

Finally, through institutional stewardship, our progress will serve as a positive example and provide a powerful springboard to the next projects adopting MBSE.

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

## 13. REFERENCES


[1] - "Europa Jupiter System Mission (EJSM): Exploring the emergence of habitable worlds around gas giants, JPL D-67959, Task Order NMO711062, 15 November 2010. Available at http://www.lpi.usra.edu/opag/JJSDTNASAReport.pdf

[2] - "An Operations Concept for Integrated Model-Centric Engineering at JPL", T. Bayer, L. Cooney, C. Delp, C. Dutenhoffer, R. Gostelow, M. Ingham, J. S. Jenkins, B. Smith, 2010, IEEEAC Paper #1120.

[3] - "Update – Concept of Operations for Integrated Model-Centric Engineering at JPL", T. Bayer, M. Bennett, C. Delp, D. Dvorak, J. S. Jenkins, S. Mandutianu, 2011, IEEEAC paper#1122.

[4] - see for example the presentation, http://www.omg.org/news/meetings/tc/mn/special-events/ecl/Rouquette.pdf.

[5] - D. E. Knuth, "Literate Programming." The Computer Journal, NO27(2) ( May): 97-111. (1984)

[6] - http://www.omgsysml.org/

[7] - http://www.iso-architecture.org/ieee-1471/

[8] - http://www.uml.org/

[9] - http://www.w3.org/TR/owl-ref/

[10] - S. Friedenthal, A. Moore and R. Steiner, in "*A Practical Guide to SysML - The Systems Modeling Language*", 2nd Edition, Chapter 16, Morgan Kaufmann Pub., Burlington, MA, 2008.

[11] - https://opfm.jpl.nasa.gov/library/, Jupiter Europa Orbiter Mission Study 2008: Final Report, Table 4.2-1, page 4.2-2.

[12] - B. Cole and S. H. Cheung, "Getting a Cohesive Answer from a Common Start: Scalable Multidisciplinary Analysis through Transformation of a System Model", paper xx, Big Sky Conference, 2011.

[13] - R. Duren, Aerospace Conference, 2006 IEEE

[14] - M. Moshir, et al., Proc. SPIE 7734, 2010

[15] - Parametric Cost Estimating Handbook http://cost.jsc.nasa.gov/pcehhtml/pceh.htm, chapter 4.

[16] - http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=05747541


## 14. BIOGRAPHIES

***Todd J. Bayer*** *is a Principal Engineer in the Systems and Software Division at the Jet Propulsion Laboratory. He is currently the Flight System Engineer for the Europa Habitability Mission Studies. Most recently he was the Assistant Manager for Flight Projects of JPL's Systems and Software Division. He received his B.S. in Physics in 1984 from the Massachusetts Institute of Technology. He started his career as a project officer in the US Air Force at Space Division in El Segundo, California. Following his military service, he joined the staff of JPL in 1989. He has participated in the development and operations of several missions including Mars Observer, Cassini, Deep Space 1, and Mars Reconnaissance Orbiter, for which he was the Flight System Engineer for development and Chief Engineer during flight operations. During a leave of absence from JPL, he worked as a systems engineer on the European next generation weather satellite at EUMETSAT in Darmstadt, Germany.*

***Brian Cooke*** *received a B.S. in Engineering Science and Mechanics from Virginia Tech in 1995. He has been with JPL for more than 15 years and is the recipient of three NASA Exceptional Achievement Medals. He is currently the Project System Engineer for the Europa Habitability Mission helping to develop and plan NASA's exploration of this intriguing Jovian moon. He has previously served as the Kepler Project System Engineer, Dawn Project V&V Engineer and the GALEX Instrument I&T Manager.*

***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. His most recent body of work concerns the infusion of systems modeling as a data structure into multidisciplinary analysis and architectural characterization. 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.*

***Frank Dekens*** *is a Systems Engineer in the Instruments and Science Data Systems division at Jet Propulsion Laboratory. He is a member of the project systems engineering team on the Europa Habitability Mission. Prior to this, he*

*was the Instrument Systems Engineer for the Space Interferometry Mission. Frank received his BS and Ph.D. in physics, both form the University of California, Irvine.*

***Christopher Delp*** *is the Systems Architect for Ops Revitalization task in MGSS. He is also a member of of the Flight Software Systems Engineering and Architectures Group at the Jet Propulsion Laboratory. His interests include software and systems architecture, applications of model-based systems engineering , and real-time embedded software engineering. He earned his M.S. and B.S. degrees from the U of A in Systems Engineering.*

***Ivair Gontijo*** *is a member of the project systems engineering teams on the Europa Habitability Mission and the GRACE-FO projects. Prior to this he lead the design and manufacturing of the RF packaging for the radar that will control the final descent of the NASA Mars Science Laboratory on Mars. His area of technical expertise is optoelectronics and RF devices. In the 90s he worked on III-V semiconductor materials, semiconductor lasers and optical waveguides. For the past 15 years he led process Engineering and Packaging groups in the fiber optics industry, developing transmission systems at data rates up to 40GB/s. He has a BSc in Physics and an MSc in Optics from UFMG in Brazil and a PhD in Electrical Engineering from Glasgow University, UK.*

***Kari Lewis*** *has been a flight system engineer at JPL since 1996. She joined JPL after graduating with her BS in Aerospace Engineering from the University of Texas at Austin. She also received her MBA from the Anderson School of Business at the University of California, Los Angeles in 2004. Ms. Lewis has worked several missions in her career, including Deep Space 2, Mars Reconnaissance Orbiter, Mars Science Laboratory, and recently JEO. Her current position is project system engineer for the JPL contributions to the Jason-3 mission.*

***Mehrdad Moshir*** *is a Systems Engineer in the Systems Engineering Section at the Jet Propulsion Laboratory. He is currently the Flight Systems Engineer for the Surface Water and Ocean Topography (SWOT) Mission. He was previously the performance modeling manager for the Kepler and SIM/SIM-Lite missions and incorporated many Model-Based concepts into this work and evolved the approaches for developing Science Margin Model. He holds an A.B. in physics from UC Berkeley as well as M.A. and Ph.D. in physics from Princeton University. He began constructing particle detectors and electronics for high energy physics experiments at Brookhaven National Laboratory while in graduate school and over time his interests migrated towards systems engineering of space missions.*

***Robert Rasmussen*** *is an Engineering Fellow in the Systems and Software Division at JPL, which he joined in 1975 with a Ph.D. in Electrical Engineering from Iowa State University. He has since supported several flight projects in both technical and management roles. These have included G&C Technical Group Supervisor during Galileo development, and Subsystem Cognizant Engineer for the Cassini Attitude and Articulation Control Subsystem. In addition, he has led various research efforts in fault-tolerant parallel computing, spacecraft automation and autonomy, and model-based software architecture. Bob has been both Chief Technologist and Chief Engineer for JPL Technical Divisions. He is presently the Architect for Europa mission studies, where he is helping to introduce and refine improved architecting and model-centric engineering methods.*

***David Wagner*** *is a software system engineer and architect in the Flight Software Applications and Data Management group at JPL and was a principal developer of the Mission Data System in 2000-2006. Since then he has continued to apply MDS technology and State Analysis in several applications. He is currently a member of the project system engineering team on the Europa Habitability Mission formulation project. He has a BS in Aerospace Engineering from the University of Cincinnati, and MS in Aerospace Engineering from the University of Southern California.*