Cloud-Based Orchestration of a Model-Based Power and Data Analysis Toolchain

Ethan Post
Jet Propulsion Laboratory,
California Institute of Technology
4800 Oak Grove Dr.
Pasadena, CA 91109
818-354-5369
Ethan.A.Post@jpl.nasa.gov

Bjorn Cole
Jet Propulsion Laboratory,
California Institute of Technology
4800 Oak Grove Dr.
Pasadena, CA 91109
818-354-0804
Bjorn.Cole@jpl.nasa.gov

Kevin Dinkel
Jet Propulsion Laboratory,
California Institute of Technology
4800 Oak Grove Dr.
Pasadena, CA 91109
818-354-7349
Kevin.Dinkel@jpl.nasa.gov

Hongman Kim
Jet Propulsion Laboratory,
California Institute of Technology
4800 Oak Grove Dr.
Pasadena, CA 91109
818-354-0608
Hongman.Kim@jpl.nasa.gov

Erich Lee
Jet Propulsion Laboratory,
California Institute of Technology
4800 Oak Grove Dr.
Pasadena, CA 91109
818-393-1696
Erich.R.Lee@jpl.nasa.gov

Bassem Nairouz
Jet Propulsion Laboratory,
California Institute of Technology
4800 Oak Grove Dr.
Pasadena, CA 91109
818-354-8386
Bassem.Nairouz@jpl.nasa.gov

***Abstract*—The proposed Europa Mission concept contains many engineering and scientific instruments that consume varying amounts of power and produce varying amounts of data throughout the mission. System-level power and data usage must be well understood and analyzed to verify design requirements. Numerous cross-disciplinary tools and analysis models are used to simulate the system-level spacecraft power and data behavior. This paper addresses the problem of orchestrating a consistent set of models, tools, and data in a unified analysis toolchain when ownership is distributed among numerous domain experts. An analysis and simulation environment was developed as a way to manage the complexity of the power and data analysis toolchain and to reduce the simulation turnaround time. A system model data repository is used as the trusted store of high-level inputs and results while other remote servers are used for archival of larger data sets and for analysis tool execution. Simulation data passes through numerous domain-specific analysis tools and end-to-end simulation execution is enabled through a web-based tool. The use of a cloud-based service facilitates coordination among distributed developers and enables scalable computation and storage needs, and ensures a consistent execution environment. Configuration management is emphasized to maintain traceability between current and historical simulation runs and their corresponding versions of models, tools and data.**

TABLE OF CONTENTS

1. INTRODUCTION 1
2. FLIGHT SYSTEM POWER AND DATA ANALYSIS 2
3. DISTRIBUTED ANALYSIS WORKFLOW 2
4. SHORTCOMINGS OF THE DISTRIBUTED ANALYSIS WORKFLOW 5
5. INTEGRATED ANALYSIS APPROACH 5
6. SYSTEM ARCHITECTURE MODEL 6
7. CHALLENGES ENCOUNTERED WITH THE INTEGRATED ANALYSIS WORKFLOW 10
8. REALIZED BENEFITS OF INTEGRATED ANALYSIS WORKFLOW 10
9. CONCLUSIONS 11
ACKNOWLEDGEMENTS 11
REFERENCES 11
BIOGRAPHY 12

1. INTRODUCTION

Europa, one of Jupiter's moons, is believed to be potentially habitable. This is, in part, due to the presence of a vast liquid water ocean believed to exist underneath an ice crust that covers its surface. Very little is known about the physical characteristics of the ice shell and ocean. The proposed Europa Mission seeks to understand the thickness of the ice crust, the extent and salinity of the ocean, the chemical composition of the crust and thin atmosphere, and the geological features and the crust, among other phenomena of Europa [1].

To accomplish this, the Europa Mission would send a spacecraft to orbit Jupiter such that it passes by Europa over multiple flybys. The notional spacecraft would be equipped with a payload of numerous scientific instruments in order to collect the necessary scientific data.

978-1-4673-7676-1/16/$31.00 ©2016 IEEE

Each scientific instrument in the payload would have unique power needs, which would vary throughout each flyby. For example, instrument behavior would vary depending on distance to Europa, local solar time, and alignment to Earth. Other spacecraft subsystems would demand power, including avionics, guidance, navigation and control, thermal control, and telecommunication. The source of this power would be the spacecraft's battery, charged by solar arrays. The solar arrays would generate power to varying degrees depending on their angle and distance to the sun, their shadowing due to the spacecraft or due to eclipse, and their efficiency, which degrades over time, in particular due to the radiation environment.

The Europa Mission concept team must size the battery and solar arrays such that they provide adequate power to the spacecraft over the course of the mission.

2. FLIGHT SYSTEM POWER AND DATA ANALYSIS

As a project moves through the design cycle it is necessary to evaluate both point design and architectural variations against requirements and resource constraints. The resources most frequently monitored during early development are mass, power, and data. Assessing these resources is a multi-variable and multi-domain problem spanning trajectory design, attitude constraints, flight system power generation and consumption, thermal control, data production, hardware design, etc. In addition to the complexities of the resource analysis is the multitude of variations in both mission and flight system design that need to be analyzed. Keeping track of inputs, outputs, analysis models and tool configurations can easily consume large amounts of a systems engineer's time.

The Europa Mission maintains a single project baseline at any point in time that provides an authoritative source of information for the flight system hardware description (e.g. Master Equipment List (MEL), Power Equipment List (PEL)) and flight system operation assumptions. This information is incorporated into various analysis tools to assess and report flight system resource metrics, including power generation capability, power and energy margins, data volume produced and returned, and data storage margin.

3. DISTRIBUTED ANALYSIS WORKFLOW

The Europa Mission flight system power and data analysis is a multidisciplinary problem, which includes trajectory planning, attitude control, 3D hardware modeling, science observation, data uplink and downlink, solar array sizing and battery sizing. The Europa Mission concept team uses a mix of off the shelf and homegrown software tools to perform the various analyses. A distributed analysis approach was used early in the project lifecycle where the various domain-specific analysis models were created independently of one another, as opposed to as part of a unified, automated analysis toolchain.

The following subsections describe the various tools and analyses used as part of the flight system power and data analysis workflow.

Activity Scheduling and State Timeline Generation

The Activity Plan Generator (APGEN) is a JPL developed tool used by mission planners and science planners to do resource-driven planning of mission activities [2]. For the proposed Europa Mission, APGEN is used to generate a schedule of all system and science activities for the entire duration of the mission. To do this, APGEN takes the projected Europa orbital trajectory, a system description of power and thermal modes, and a set of spacecraft behavioral rules. Figure 1 is a typical APGEN graphical output, showing scheduled activities and resource timelines.

The input deck to APGEN includes defined resources, such as the amount of available propellant or the battery state of charge, and a set of rules, such as producing error conditions when star trackers are occulted, or when the battery state of charge drops below a minimum threshold. Using these inputs, APGEN simulates the spacecraft's power state and thermal state throughout the mission. This mission profile provides the landscape on which a collection of APGEN scheduling algorithms is executed. These algorithms deterministically schedule spacecraft and instrument activities. These activities are simulated intervals of time in which the spacecraft is performing some action, such as slewing before taking science images, turning on the radio before contacting Earth, or firing thrusters during an orbit correction. These activities, in turn, will also consume resources, and may violate rules, so the mission is re-simulated to ensure the schedule's validity. Ultimately APGEN produces a candidate mission plan, which is a valuable input to many simulation tools downstream.

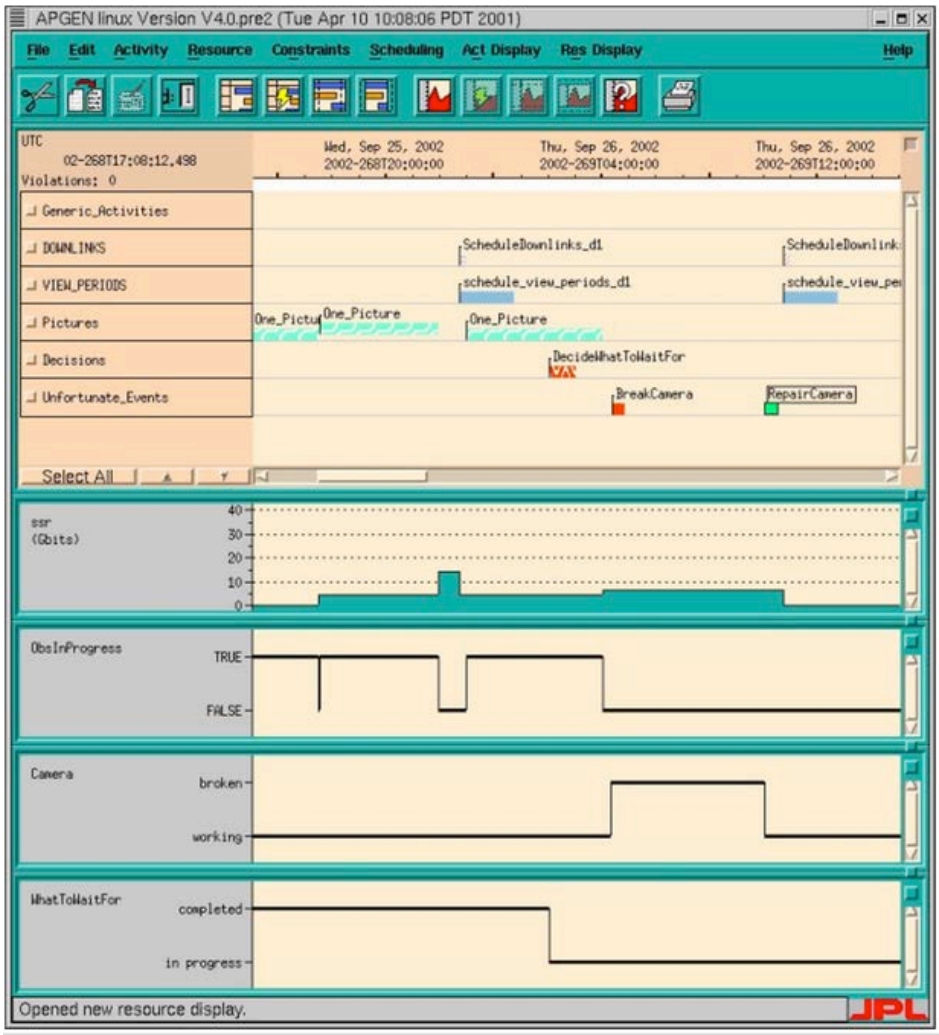

Figure 1: Typical view of mission activity schedules and resource timelines in APGEN.

APGEN only accepts two forms of inputs, both of which are in formats specific to APGEN. The first is the APGEN Adaptation File (AAF), which defines the resources, rules, and algorithms used to schedule activities in the APGEN domain specific programming language. The second is an Activity Plan File (APF), which is a set of pre-determined time ordered activities used as the starting point for planning. As output, APGEN produces a file in another tool specific format called the Time Ordered Listing (TOL), which is a file that lists all the activities scheduled in the plan, along with all the resource changes over time. Because no other tools use APGEN formats natively, transformations must be used to transfer data to and from the APGEN specific formats.

Detailed Power Analysis

The Multi-Mission Power Analysis Tool (MMPAT) is a high fidelity power modeling tool built on hardware test data and historical mission operations [3]. It is the institutional approved source for power resource analysis during design and operations. MMPAT is used to produce high-fidelity state variable timelines for parameters like battery state of charge, bus voltage, solar cell current and voltage, and solar array power. While other tools are capable of analyzing these parameters they use more simplified models. This makes MMPAT more suitable for high fidelity power modeling and less suitable for quick trade studies.

Spacecraft 3D Geometry Model

A 3D computer aided design (CAD) model of the notional Europa Mission spacecraft is used for a solar array shadowing analysis. Depending on the orientation of the spacecraft with respect to the sun, the solar arrays experience varying amounts of sunlight, which affect the effective area of the solar arrays. This CAD model is generated in Siemens NX then converted into a format suitable for use in the AGI Systems Toolkit (STK) solar panel toolkit. The format conversion process is somewhat manually intensive given that a subset of the full CAD model is manually selected for conversion. This is because not all of the hardware components and geometric features are necessary for the shadowing analysis. A CAD model with a larger memory footprint requires more computation

time during the simulation, thus it is beneficial to reduce the size of the CAD model. The format conversion is accomplished using the 3D CAD conversion tool, Anark Core.

Solar Array Shadowing and Radiation Analysis

STK is a software suite used to generate several key state timelines in the flight system power analysis, including solar array shadowing throughout cruise and Jupiter tours, along with radiation induced solar array degradation. To determine the 1-MeV equivalent fluence, JPL built a custom plugin, Jupiter Environment Toolkit (JET) [3], to integrate the GIRE2 Jupiter environment model into STK [4]. Figure 2 shows the Jupiter radiation plane visualized through the JET plugin. A set of Matlab scripts is used to control and modify STK scenarios.

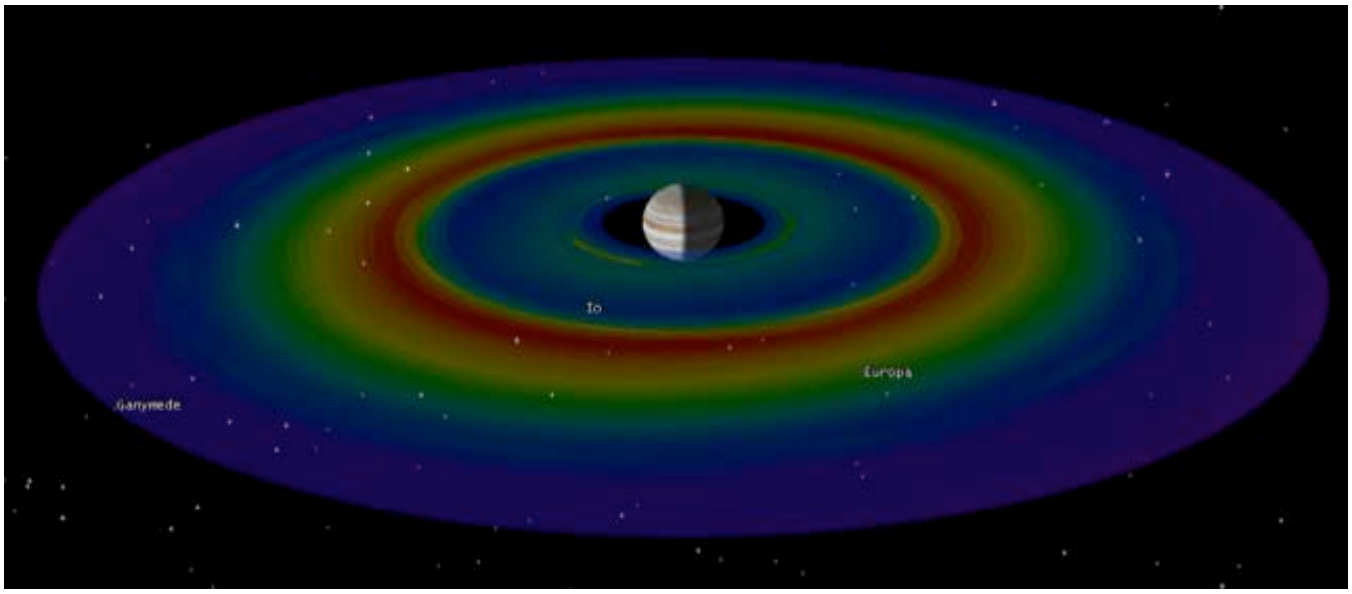

Figure 2. STK Jupiter Environment Toolkit (JET) plugin visualization of the Jupiter radiation plane.

Technical Resource Post Processing

Wolfram Mathematica is used to post-process simulation data and assess technical resources (e.g. power, data, component lifetime) over both large and small intervals of time. The integration between Mathematica and SystemModeler allows the Europa Mission team to analyze complete mission profiles for both, potential nominal and off-nominal scenarios. Off-nominal scenarios include spacecraft or payload faults in addition to "what-if" scenarios. Traditionally, during concept development, sizing or stressing scenarios are pre-defined using engineering estimates. For the Europa Mission concept, these scenarios are allowed to emerge from the simulation results.

Flight System Modelica Model

Wolfram SystemModeler is a simulation engine for Modelica models. The flight system Modelica model is a representation of the Europa Mission flight system concept as shown in Figure 3. The model is comprised of individual subsystems and instruments using pre-computed activity and event timelines from APGEN. Notional payload instruments in the model include an ice penetrating radar (ipr), thermal imager (thermi), reconnaissance camera (recon), shortwave infrared spectrometer (swirs), topographical imager (topo), neutral mass spectrometer (nms), magnetometer (mag), and Langmuir Probe (lp).

SystemModeler is able to simulate year's worth of operations in several minutes and assess flight system power and data resource metrics (e.g. minimum battery state-of-charge, bulk data storage) to evaluate both baseline and trade study options. The combined capabilities of SystemModeler and Mathematica have been demonstrated to be a suitable alternative to MMPAT for running quick, lower fidelity trade studies.

Using Mathematica as a front end to configure parameters and execute simulations enables an analytical capability and analysis traceability that would otherwise be difficult to maintain early in the project lifecycle.

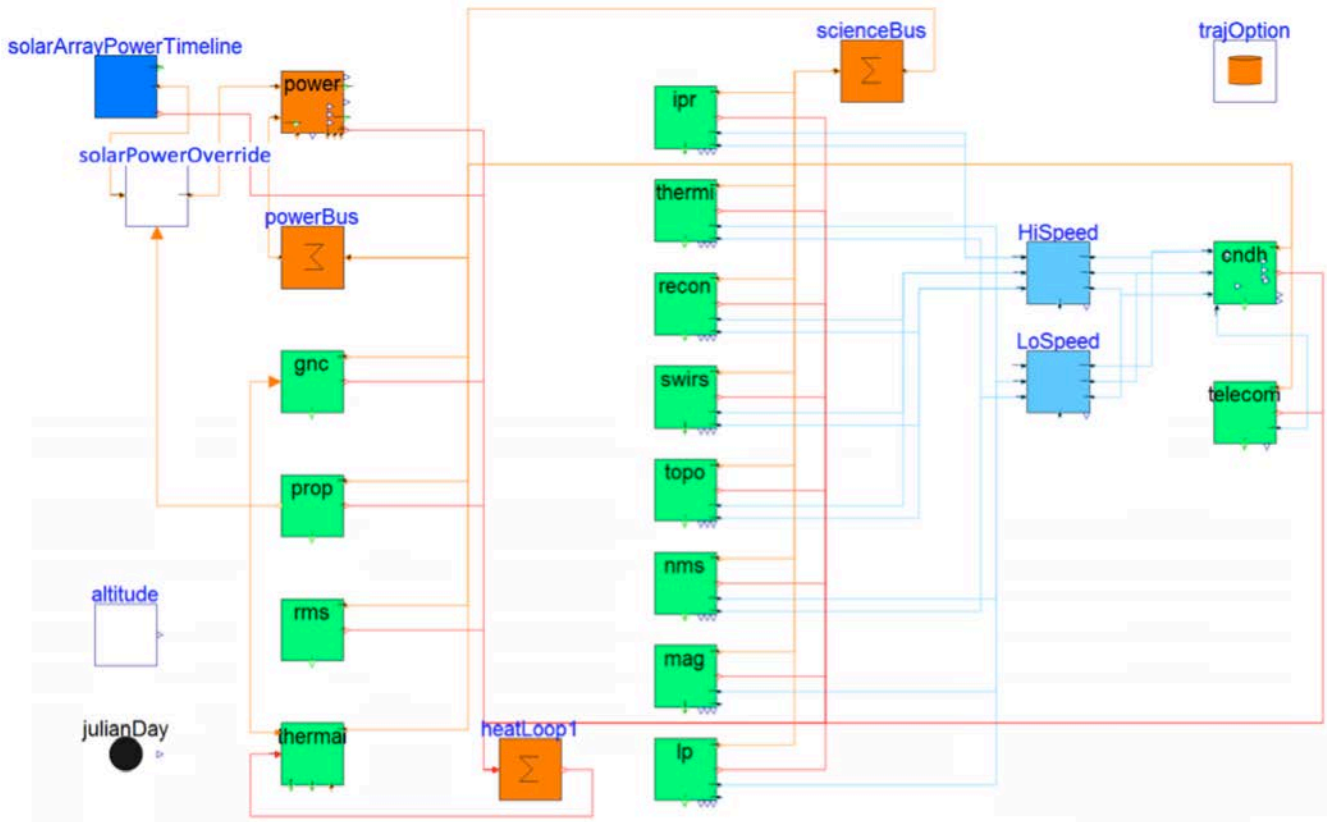

Figure 3. SystemModeler representation of the conceptual Europa Mission flight system Modelica model.

4. Shortcomings of the Distributed Analysis Workflow

Due to the specialized nature of these analyses and tools, the analysis work is distributed among numerous specialists. While this distributed analysis approach allows for numerous specialists to create and run detailed analysis models, this approach leads to some challenging technical interfaces between numerous people.

Isolated Analysis Models

Analysis models are often developed in isolation, with the intention of standalone, local execution. Inputs are limited to the model's use case(s), such that all information needed from other analyses (e.g. other models) are assumed and hardcoded. The advantage is to enable standalone execution, but the disadvantage is to limit the model's connectivity to other models. An integrated model-based systems engineering (MBSE) environment requires that models exchange data, which requires automated connectivity.

Model Fidelity

The fidelity level may be dissimilar across different analysis models, resulting in less accurate computations. Not only this, but fidelity usually correlates with execution time. Therefore, care has to be taken with respect to computational load balancing and order of execution.

Data Consistency

Data consistency across all models is essential. Inputs common to multiple analysis models should originate from a single source. Typically, this single source of information is non-existent, and needs to be built from scratch.

5. Integrated Analysis Approach

In response to the shortcomings described in the previous section the Europa Mission team developed a more integrated approach to flight system power and data modeling and analysis.

Approach

The integrated workflow development involved much interaction with the subject matter experts for the various analyses. This was necessary to understand what analytical models, software tools, and platforms were being used. Moreover, it was necessary to learn how the models and tools worked, and what data and software interfaces existed between them.

For each analysis model it was necessary to learn the assumptions, the set of input and output variables, and the various data formats used. In order to expose the model

inputs and outputs to a unified analysis coordination tool there was generally some amount of code refactoring necessary. The desire was to refactor code minimally so as to make use of existing code as much as possible and to maintain the capability to perform standalone analyses when needed. The analysis models were refactored as necessary and wrapped in Phoenix ModelCenter components. These components were tested with a range of input datasets and their results were compared with those of the underlying model when run standalone to ensure the results were as expected. This contributed to the validation of the wrapped analysis components.

Motivation and Perceived Benefits

Integrating distributed, non-connected models into a single integrated model has several benefits. It is a prerequisite for execution automation, which enables conducting end-to-end trade studies across the whole system. Automation allows for large-scale design of experiments (DOE) techniques, which may be used as part of flight system design optimization. Responses to engineering change requests can be produced more readily because a change in one of the main upstream inputs of an analysis is more easily propagated downstream using the integrated model. Automation reduces the time it takes for this change to affect parts of the system. More frequent MEL and PEL versions can be produced based on design changes, ensuring a better consistency across all elements of the system.

End-to-end workflow integration allows for better traceability of the upstream analysis model inputs to the downstream analysis model outputs (e.g. how the choice of trajectory affects the number of battery recharge cycles). Hence, a better understanding of the system behavior can be achieved. Wide scope "what-if" scenarios are possible only with an integrated model using automation.

Automation reduces many of the manual and repetitive aspects of running an end-to-end simulation. Much time can be saved if a single analyst can run a simulation based on a new set of design inputs, rather than multiple analysts coordinating their schedules, emailing results back and forth, and verifying the consistency of analysis assumptions. Moreover, many errors can be avoided if the number of manual data exchanges and model updates can be reduced.

The existence of an integrated model, with less simulation time preparation overhead, encourages more users to run more case studies, more scenarios, and more thoroughly explore the design space. Improved designs can be reached, with emphasis on technical, programmatic and financial design parameters.

6. System Architecture Model and Model Connectivity

The proposed Europa Mission has been using an MBSE approach to develop its mission design. A Systems Modeling Language (SysML) based system model is used to capture the system architecture and behavior, and it is used as the authoritative source of the current state of the mission design. The Europa Mission system architecture model describes structural decomposition of the flight system as well as component design parameters and behavioral descriptions, including state machines that describe the spacecraft's power behavior, as depicted in Figure 4.

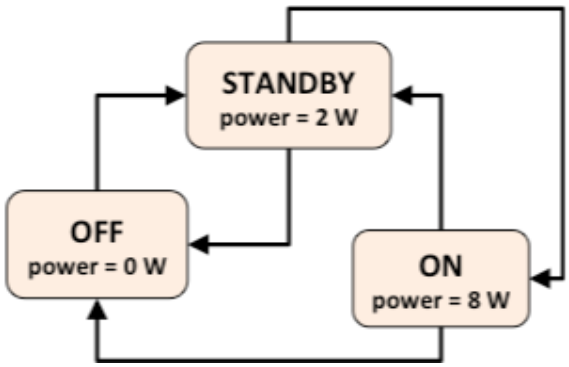

Figure 4: Notional representation of an instrument's power consumption state machine.

The system model is largely descriptive and provides limited analytical capabilities. Therefore, there is a strong need to connect the system architecture model with the analytical models discussed earlier. For example, analytical models need to use data available in the system model as inputs, and the system model needs to be updated based on the results from analytical models. This section describes how the system architecture models are created and managed for the Europa Mission concept.

Connectivity with Web-Based Model Repository

To support different types of users who need to interact with the system model, the NoMagic MagicDraw SysML authoring tool is used together with a web-based system model repository. Figure 5 summarizes the tool environment for the Europa Mission system model.

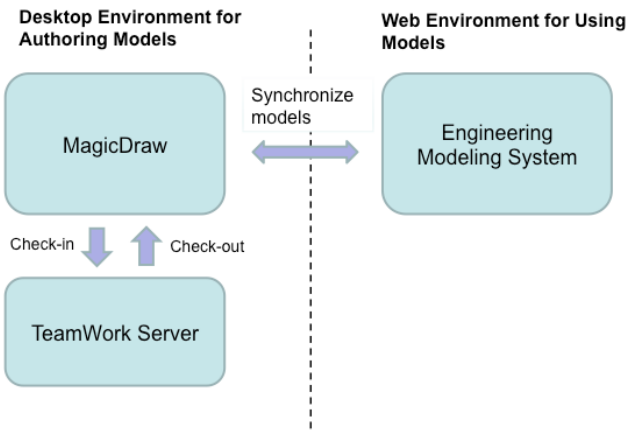

Figure 5. Tool environment for defining the system model and for viewing reports derived from the system model.

A dedicated team of SysML modelers uses MagicDraw to define the structure, behavior, requirements, and analytical relationships of the system. SysML models are collaboratively developed and version controlled using NoMagic's Teamwork server.

To support broader users in the project, SysML models are published to the Engineering Modeling System (EMS) server, a web-based model repository, developed by JPL. EMS presents system information in webpages using Views and Viewpoints defined in the model [5]. For example, a View may present information in tables and paragraphs. This allows users to consume system information in a format that they are familiar with. Users can edit documents, edit values of system model elements, and post comments through the web interface.

Since system model elements can be edited either in MagicDraw or in EMS, the system model must be synchronized between both. A MagicDraw plug-in was developed to synchronize models. Using the plug-in, changes can be pushed from MagicDraw to EMS or vice versa.

System Model Interface to Analysis Model Inputs

The flight system analysis toolchain development began with a set of point-to-point data transformations, writing scripts to take data from one tool's format and convert it into another. This solution is not scalable, and can quickly become unmanageable with a large amount of tools, all changing their interfaces regularly.

A better solution has been to transform data from a tool into an intermediate representation. Exports from any tool are produced in this intermediate representation, and imports into any tool are transformed from this intermediate representation. This method decouples the tools, allowing one tool to change without affecting another, while simplifying the transformation software, encouraging code reuse. This method has been employed successfully on the Europa Mission in the form of a simplified JavaScript Object Notation (JSON) representation for transformations from the MagicDraw system model to external tools.

Data is transferred from MagicDraw to APGEN using a model transformation. This transformation is accomplished through a series of simplified intermediate representations [6]. The steps are summarized below, and visualized in Figure 6:

1) SysML patterns describing the Europa Mission flight system (Metamodel A) are matched and extracted into a simplified restricted model representation.

2) This restricted model representation is transformed into a set of JSON files representing the data objects and connections between them.

3) These JSON data objects and relationships are parsed by a transformation script, which auto codes the necessary Adaptation Files for APGEN to ingest (Metamodel B).

Figure 6. Complex model transformations on the Europa Mission concept are conducted by first mapping a source model to a restricted intermediary JSON format. From this restricted JSON a new model can be built up in a different format.

Certain system variables in the APGEN Adaptation Files are exposed for modification via a wrapper script. This wrapper script allows a user to quickly modify parameters such as the vehicle battery capacity or the cable loss factor in order to run quick trade studies or parameter sweeps within APGEN. These sweeps can be made independently of the source data that come from the system model.

APGEN was wrapped in a Phoenix Analysis Server script wrapper in order to easily connect APGEN results to downstream simulations. Script wrappers allow custom tools, such as APGEN, to be run remotely from a local ModelCenter instance.

System Model Interface to Modelica Model

The SysML-based system model contains representations of the flight system components' power-related behavior in the form of state machines. These state machines include mappings of component power states to power values. A MagicDraw plugin is used to auto generate Modelica files through a model transformation based on these mappings. These Modelica files then feed into the SystemModeler analysis.

Time-Oriented Data Archival

The Timeline Management Service (TMS) is a web service used to store time-oriented data timelines [7]. On the proposed Europa Mission, these data timelines originate as outputs from the analytical tools described previously and can be analyzed at a later time or used as inputs for downstream analyses. These timelines can be stored using a variety of time formats, including Julian and Coordinated Universal Time (UTC). Uploading, downloading and time format transformations are achieved through a Representational State Transfer (REST) interface. Requests to TMS are made through scripts, which are wrapped in Analysis Server script wrappers, which are called by ModelCenter.

Cloud-Based Model Execution

Integrated models in ModelCenter represent engineering analysis workflows that can be run in an automated fashion. While the models are suitable to engineers who create, maintain and use them on a daily basis, there is a need to make the engineering workflows available to a broader user base on the project. For example, once an engineering workflow is well defined, it can be used by casual users to perform quick "what-if" studies. This paradigm of dual use cases is similar to that of the SysML-based system model, where MagicDraw is used as an authoring tool by SysML experts and a web-based model repository is used by a wider user community.

ModelCenter Cloud (MCC) is a web-based service that executes analysis workflow models using distributed computing resources. Workflow models created in ModelCenter Desktop were published to MCC so they could be executed through a web interface. Figure 7 shows the process to publish a workflow model to MCC. First, individual analysis components are packaged and uploaded to MCC in its model repository. Analysis components are put under version control in MCC. Second, the workflow model in ModelCenter is updated to use specific versions of the components in MCC. Third, the updated workflow model is packaged and uploaded to MCC.

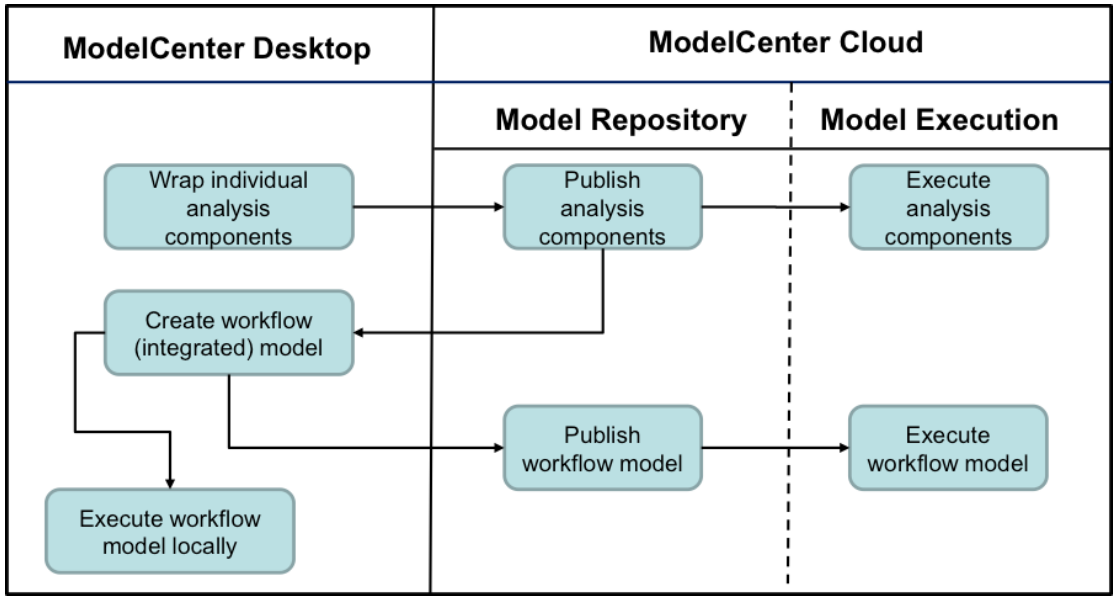

Figure 7. Engineering analysis workflows are defined in ModelCenter Desktop and then published to ModelCenter Cloud, which enables repeatable analysis workflows through a web interface.

The workflow model is also version controlled in MCC. Therefore, it is possible to keep track of the complete state of a workflow model including its analysis components.

During the packaging and publishing process, it is possible to select a subset of variables of a model that will be exposed in MCC. This mechanism is used to abstract out a complex workflow so that end users of MCC can work with a simplified interface to the workflow model. Users can execute single instances of workflows or execute DOE workflows that perform analyses on multiple input sets.

When a workflow model is executed, MCC will find an appropriate computing node that meets the computational requirements for the model (e.g., computer platform and required application software) and forward the job execution to the node. The distributed computing and load balancing of MCC allows users to run different workflow models at the same time. Since published models are executed through a server, it is possible to provide repeatable engineering workflows to many users. MCC's repository area can be used to store result files produced from analysis workflows.

System Model Interface to Analysis Model Outputs

It is necessary that the integrated model outputs be displayed consistently and easily to the user. A ModelCenter component was developed that accepts the integrated model's inputs and outputs, and passes them to an EMS web report. Figure 8 depicts the complete round trip of data where input parameters originate from EMS, get processed through analysis models, and the resulting outputs get published back to EMS. While most of the analysis components have been demonstrated in the cloud-based framework, the STK and Matlab components are currently in development.

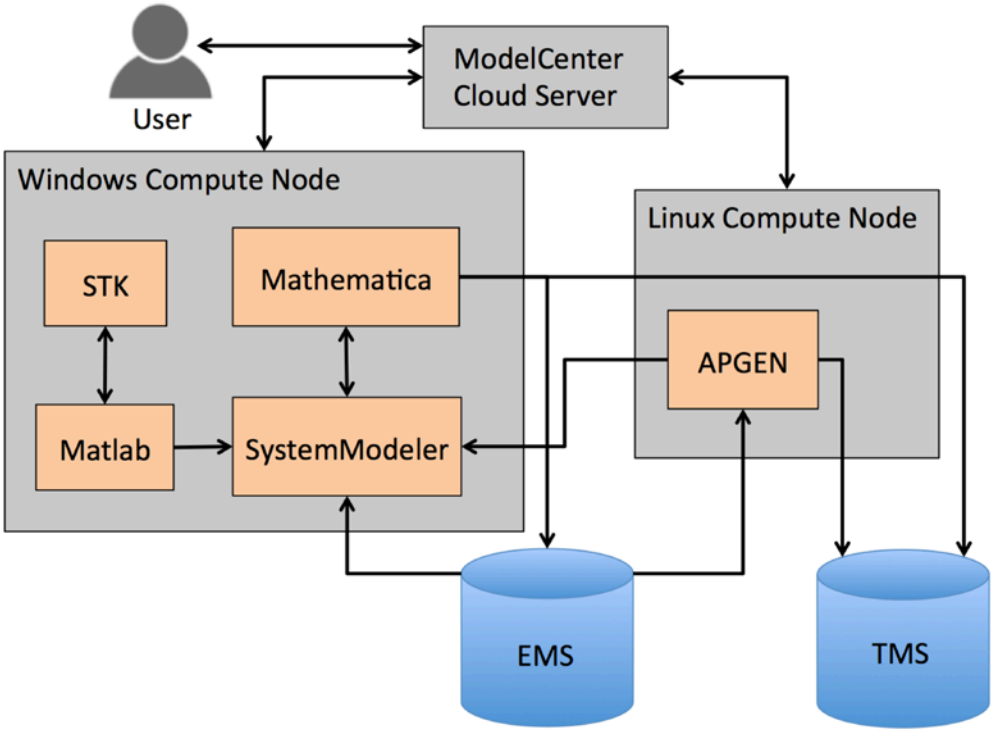

Figure 8. Users interact with the integrated cloud-based analysis toolchain through the ModelCenter Cloud server, which coordinates the execution of analyses on compute nodes. Engineering inputs are pulled from EMS and outputs are pushed to EMS and TMS.

7. Challenges Encountered with the Integrated Analysis Workflow

Several challenges were encountered throughout the analysis toolchain integration process.

Data Traceability

One of the common challenges across all analysis components was with exposing input and output variables. Since the analysis models were developed originally in a distributed manner, where typically a single subject matter expert developed and ran the simulations, there was not a strong need to organize the code such that input and output variables were grouped together conveniently. Instead, many input values were hard-coded throughout multiple files. Moreover, the variables were named differently across the different analysis models, which added confusion. In addition, tracing a given input set to a resulting output set, and associating these with a given integrated model version, is complex. This complexity is partly due to the large number of input, output, and intermediate analysis parameters.

Multiple Platforms

Another challenge was coordinating multiple operating systems. Some of the analysis models ran on Linux applications (APGEN, TMS) and some ran on Windows applications (STK, Mathematica, SystemModeler, MagicDraw, ModelCenter). Having analysis models on different platforms complicated the transfer of files and data. By wrapping the Linux-based components with Analysis Server script wrappers, ModelCenter could pass information between analysis components regardless of where the components resided. Mapping shared drives was a quick fix for transferring files between machines throughout development.

Authentication

When analysis models are run in a distributed manner, one at a time, it is fine for a user to provide their credentials manually each time, for example through a command line prompt or a log in screen. However, when analysis models are run in an automated end-to-end manner, it is impractical for the user to re-enter their credentials partway through the simulation. TMS, EMS, and Teamwork each require user authentication, which makes management of credentials, including security tokens, a complex problem.

Analysis Data Interfaces

Interfaces between models are far from standardized. Some models require direct inputs, while others require data files in different formats. In many cases, analysis model developers will resort to building interface models (or modules) to act as translators between analysis models, adding to the architectural complexity.

Configuration Management

Configuration management (CM), including versioning, is another issue that has to be tracked if the model is part of an integrated model. When analytical models are developed independently of one another, they often have different CM approaches, which add complexity to an integrated analysis model.

The combination of multiple analysis tools, many models, data repositories, as well as large number of interface parameters and files, increases the integrated model's architecture complexity. Practitioners should take extra care in designing the integrated models so as to decrease complexity. Automation of some parts of the integrated model building process may be helpful in some cases.

Hence, building an MBSE environment utilizing an integrated model is not a straightforward task. Many challenges have to be overcome in the process.

8. Realized Benefits of Integrated Analysis Workflow

Several notable benefits have been realized since pursuing the integrated workflow approach.

Improved Interfaces

The MCC front end provides a simple and uniform interface to different workflows. Multiple analysis jobs can be queued, by multiple users, both for multiple workflows as well as for DOE trades for a single workflow. The process of wrapping analysis models has enforced clear exposure of input and output parameters within the analysis code, which makes the parameter interfaces easier to understand and is good design practice in general.

Configuration Management

The Wrapped models and workflows are individually versioned through MCC which makes it easy to rerun old analyses for traceability. Input and output parameter sets are also saved for each run, which improves traceability.

Streamlined Reporting

The ability to publish results automatically to a central web-based viewer enables quick reporting, accessible to the Europa Mission team, presented in context with information from across the project.

Broader Exposure of Analysis Models

New users are able to use the model more readily through the cloud-based interface. It is no longer necessary for a new user to install all the analysis software, obtain licenses, and gather all the necessary models on their own machine in order to run an analysis. This expands the model usage in design and analysis to a wider community within the project.

9. CONCLUSIONS

The existence of a distributed, loosely connected, collection of models prohibits MBSE. To enable MBSE, individual models should be connected into an integrated model, together with common inputs shared across all the models, and results saved to a common repository. However, connecting these models into an integrated model imposes a plethora of challenges to any MBSE environment. Such challenges drastically limit the benefits of MBSE if not addressed.

Since engineering tools are designed with their specific domains in mind, they rarely have uniformity in their interfaces. There are usually only a few experts for each individual tool, who are rarely experts in other discipline tools. Because systems engineering involves exploring the design space while taking all disciplines into account, it is always necessary to utilize many tools in order to solve problems. When trying to combine toolsets into a push button analysis, many issues can arise.

Transferring data from the format of one tool to the format of another is a major challenge. Each tool will often require a different subset of data or a different structuring of that data. This makes data transformations between tools nontrivial. Products like Phoenix ModelCenter can help ease the data transfer problem, but they are not an out of the box solution to the simulation toolchains needed for the Europa Mission concept.

Even after developing complicated transformations between tools, they can quickly become out of date. While developing the Europa Mission concept, the simulation tools often need to change in order to meet the analysis needs of the project. These upgrades can often change the data interface to or from the upgraded tool, breaking any data transformation utilities. Versioning multiple different analysis tools and the transformation tools between them can quickly become a configuration management headache.

The transition to a cloud-based integrated analysis toolchain has yielded numerous benefits. Improvements to data interfaces, configuration management, results reporting, and user accessibility have been significant. As more components become fully integrated into the toolchain, it is expected that flight system power and data analysis will realize further process improvements and that the interactions between flight system parameters will be more deeply understood.

ACKNOWLEDGEMENTS

This work was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. Copyright 2016. All rights reserved.

REFERENCES

[1] Europa Study Team, "Europa Study 2012 Report", National Aeronautics and Space Administration, May 1 2012.

[2] P. Maldague, S Wissler, M. Lenda, D. Finnerty, "APGEN scheduling: 15 years of Experience in Planning Automation," in *Proceedings of SpaceOps Conferences*. Pasadena, CA: AIAA, 2014.

[3] E. Sturm, "The Jupiter Environment Tool" *AGI Virtual Users' Conference*: California Institute of Technology, 2011.

[4] R. Evans, D. Brinza, "Grid2: A Program for Rapid Estimation of the Jovian Radiation Environment", National Aeronautics and Space Administration, January 2014.

[5] C. Delp, D. Lam, E. Fosse, C. Lee, "Model Based Document and Report Generation for Systems Engineering," in *Proceedings of Aerospace Conference*. Big Sky, MT: IEEE 2013.

[6] B. Cole, K. Dinkel, "Multidisciplinary Model Transformation through Simplified Intermediate Representations," in *Proceedings of Aerospace Conference*. Big Sky, MT: IEEE 2016.

[7] W. Reinholtz, "Timeline as Unifying Concept for Spaceraft Operations," in *Proceedings of SpaceOps 2012 Conference*. Stockholm, Sweden: AIAA 2012.

BIOGRAPHY

Ethan Post *received a B.S. in mechanical engineering from the Massachusetts Institute of Technology in 2006 and an M.S. in mechanical engineering from the University of California, Los Angeles in 2008. He is a systems engineer and has been with JPL for 7 years. He has supported hardware design, delivery, integration and testing for various flight projects, including Mars Science Laboratory, Orion EFT-1 and the Low Density Supersonic Decelerator supersonic flight demonstration test. His other interests include model-based systems engineering infrastructure development.*

Bjorn Cole *is a systems engineer at the Jet Propulsion Laboratory in the Systems Modeling, Analysis, and Architectures group. He is currently the lead of the Project Systems Engineering Analysis team on the Europa project, which is charged with deploying MBSE analysis capability. He has previously supported a series of MBSE research efforts and was a member of both the A-Team and Team X formulation groups at JPL. He has a Ph.D. and M.S. in aerospace engineering from the Georgia Institute of Technology, and a B.S. in aeronautics and astronautics from the University of Washington.*

Kevin Dinkel *is a flight software engineer at the Jet Propulsion Laboratory, focusing on small-scale flight systems. Prior to JPL, he worked at the Southwest Research Institute (SwRI) as a cognizant software engineer for 3 high-altitude balloon payloads and for the Laboratory for Atmospheric and Space Physics (LASP) as a mission planning software engineer. Kevin graduated with a Masters in Aerospace Engineering from the University of Colorado at Boulder in 2014.*

Hongman Kim *is a systems engineer at Jet Propulsion Laboratory. He is currently working on model integration and resource analysis for NASA's Europa project, and development of engineering environment for JPL's Innovation Foundry. Before joining JPL, he was a project manager at Phoenix Integration, Inc., where he led development of model-based systems engineering (MBSE) technology. He also worked on a number of government funded projects including points cloud visualization, distributed computing, and design optimization. He holds a PhD degree in Aerospace Engineering from Virginia Tech.*

Erich Lee *received a M.S. in Space Systems from Florida Institute of Technology in 2012 and a B.S. in Aerospace Engineering from California State Polytechnic University, Pomona in 2009. He has been with JPL for almost 10 years and is currently the Resource Systems Engineer on the Europa Mission with a focus on technical resource management and integrated Flight System analysis. Prior to working on the Europa project, he completed the JPL Phaeton Early Career Hire (ECH) program and worked on the Mars 2018 and TRaiNED project.*

Bassem Nairouz *is an Aerospace Systems Engineer in the Project Systems Engineering and Formulation Section (312) at JPL. He received his M.Sc. and the Ph.D. degrees in Aerospace Systems Engineering from the Georgia Institute of Technology, in 2006 and 2013 respectively. He participated in several MBSE initiatives, ranging from the project formulation phase to flight projects. He focuses on data analysis and modeling. He is a member of the JPL Europa Project Systems Engineering and Analysis Team. His research interests include Model Based Systems Engineering, Aerospace Systems Engineering, System Architecture, Data Analysis, Data Architecture, Simulation, SysML, Modeling, and Integrated Computational Models.*