INTEGRATED MODELING AND SIMULATION FOR CYBERPHYSICAL SYSTEMS EXTENDING MULTI-DOMAIN M&S TO THE DESIGN COMMUNITY

Theodore A. Bapty
Institute for Software Integrated Systems
Vanderbilt University
1024 16th Ave S.
Nashville, TN, USA
Theodore.A.Bapty@vanderbilt.edu

Jason Scott
Institute for Software Integrated Systems
Vanderbilt University
1025 16th Ave S.
Nashville TN, USA
Jason.Scott@vanderbilt.edu

Sandeep Neema
Institute for Software Integrated Systems
Vanderbilt University
1025 16th Ave. S.
Nashville, TN 37212
Sandeep.Neema@vanderbilt.edu

Robert Owens
Institute for Software Integrated Systems
Vanderbilt University
1025 16th Ave. S.
Nashville, TN 37212
Robert.Owens@vanderbilt.edu

ABSTRACT

Modeling and Simulation is increasingly important in managing design of complex cyber-physical systems. The diversity of tools, each with unique syntax and semantics, necessitates creation of multiple independent models in order to fully analyze a system. In addition to increasing the cost and time required to fully leverage computational tools, issues with model synchronization present a hazard. This is only compounded when multiple design options must be considered. In this paper, we discuss work on the OpenMETA framework (Sztipanovits et al 2015), which uses a single integrated model, coupled with automated design composition to transform the model into the specific syntax and semantics of multiple analysis tools for computing system performance. Design space specification primitives, and exploration tools allow an entire design family to be explored, across multiple physical domains and levels of abstraction. A simple design study is reviewed to demonstrate the process.

Keywords: Model-Based, Design Space, Simulation Composition.

1 INTRODUCTION

Cyberphysical system design, especially for cost, performance, safety-critical systems is a highly complex, multi-domain task. The complexity derives from multiple sources:

- A wide variety of requirements, some of which will be hard constraints (e.g. maximum weight, output voltage, etc.), some of which are negotiable (e.g. minimize cost, maximize acceleration, etc.)
- A high degree of flexibility in design approaches: the designer can choose from a wide variety of architectures and components in fulfilling requirements.
- Significant interactions across physical domains: A mass of a vehicle impacts torque needed to achieve acceleration; increasing torque may require a heavier powertrain; more power

SpringSim-Mod4Sim 2017, April 23-26, Virginia Beach, VA, USA

consumption produces more heat; software can be tuned to produce more power, but to the detriment of reliability and emissions.

Market forces can change rapidly, resulting in the need for short design cycles and/or the ability to change the system at late stages in the design process. Clearly, a build-and-test approach is insufficient as complexity, NRE cost, and design times become more challenging. This has been recognized by the industry, and a large number of complex design processes are now fully electronic, with significant amount of modeling and simulation to support analysis.

The tools that support modeling and simulation are often loosely coupled, and stovepiped by domain and level of abstraction. One set of tools is used for conceptual design, while an entirely different set is used for the detailed design phase. The tools are further specialized across domains, for example CFD for fluids and Structural FEA for mechanics. A large aerospace company will use on the order of a thousand unique tools, which requires significant staffing to maintain the expertise needed to fully leverage the tools. Modeling and simulation is expensive, due to several factors:

- Multiple models are created and maintained, each customized for a particular purpose, a labor-intensive activity.
- The cost and time to create models can have an adverse effect on design agility. Resources may permit full evaluation of only a small number of designs, where a more comprehensive study could reveal more optimal solutions.
- Keeping the models in sync can be a challenge. Mismatched models can hide issues that do not become apparent until integration and test.
- Specialized teams focus on specific subsystems, abstractions, and domains, e.g. Thermal Systems Simulation (e.g. Modelica or Simulink), Electrical Simulation (E.g. Spice), Finite Element 3D Thermal Simulation (e.g. NASTRAN), 3D Electromagnetic Simulation of RF circuits (e.g. Ansys), 3D solid models (e.g. Creo), or 3D Structural Models (e.g. Abaqus)

Large companies, such as Dassault Systemes and MSC have been successful at supporting these core modeling and simulation needs. They consistently work to provide integrated tool environments, such as 3D Experience. These provide a significant advantage in linking models together, automating workflows across disparate tools to achieve systematic, cross-disciplinary analysis. They provide an environment for managing models, specifying common system parameters and their linkage to multiple models, and how results flow from one model analysis to another. The tools assist the designer in executing models and managing results of an analysis.

Thus, a comprehensive modeling and simulation approach is often limited to highly integrated, complex systems integrators. Even for companies with the personnel resources, the modeling and simulation activities are costly, and can limit development schedules. The value of Modeling and Simulation is clear: Complex systems would not be possible without these tools. Our research involves how we can bring the power of modeling and simulation to a wider range of users and uses. This necessitates a significant reduction in the expertise and manpower needed to create, execute, and comprehend M&S results in the context of design of smaller and more diverse systems. See (Sztipanovits et al, 2015).

This paper will describe a prototype Cyber-Physical modeling environment, named OpenMETA, which has the explicit goal of modeling systems in a single, high-level modeling language, such that the model is executable across multiple domains and multiple abstractions. The models are independent of the analysis tool, and the environment performs automated composition to the target tools.

2 THE OPENMETA FRAMEWORK

The OpenMETA Cyber-Physical modeling framework is an approach to analysis-based systems design, specifically designed to provide a high level of abstraction from the analysis tool syntax and semantics. Its aim is to allow designers, systems engineers, and analysts to use a single, comprehensive model for the target, while supporting all necessary analyses to design systems which meet requirements. The framework supports simulation of the system across multiple domains and abstractions. The underlying objectives are threefold: 1) Reduce the effort (and cost) necessary to design a system; 2) Encourage cross domain analysis of full systems, including cross-disciplinary interactions; and 3) Reduce the barrier to entry, in terms of the level of expertise needed to leverage the tools, and minimize the cost of design software. To accomplish these objectives, we follow a comprehensive modeling approach, where:

- Systems are created using multi-domain components, which encapsulate multiple physical and behavioral models of the component, including solid models (i.e. CAD) and ODE-based models (i.e. Modelica), while spanning multiple abstractions, from simple analytical/lumped to complex, spatial, time dependent models. Systems are composed by connecting these components in hierarchical system models.
- Design flexibility is captured at any and all levels of the design hierarchy, using alternatives for component and subsystem structural choices. In addition to structural options, parameterization is supported.
- Requirements are also modeled, as executable specifications. These models are able to be evaluated automatically, using industry-standard tools.

2.1 Modeling Approach and Semantics

At the core of the OpenMETA process is a comprehensive modeling language, termed CyPhyML, for Cyber Physical Modeling Language. The modeling language has two competing demands: simplicity and depth.

- For simplicity, the details of the target tool syntax and semantics are largely hidden from the user. Rather than forming a union of all target tool languages, CyPhy provides a higher level representation, with semantic mappings defined per target.
- With respect to depth, the CyPhy Language has formally defined semantics for both system and mapping to tools. (see Figure 1a) The CyPhy language semantics are precisely specified per the Microsoft Research tool FORMULA (Simko *et al.*, 2013), as described in Figure 1. FORMULA describes both the semantic meaning of the model, and the mapping to the target tool semantics. The importance of a strong foundation cannot be understated given the requirement to map the models to many diverse semantics, and the precision required to achieve predictable translations. This establishes the meaning for the models, and the expectations of how the models will be interpreted in the target simulation tools.

Components effectively "wrap" the simulation artifacts, and provide a common interface for component-component interaction. (Figure 1b). The interfaces, termed Connectors in CyPhy, encapsulate all interactions, including acausal power transfer (mechanical, electrical, fluid, thermal), causal signals, and 3D structural connection points. A Parametric interface allows for flexible component models to represent physical artifacts that can be varied during manufacture or runtime. Modeling and simulation artifacts include representations of the component in specific tools, including dynamic simulation languages (Modelica, Simulink, Spice), Software/Cyber (SystemC, C), and physical (ECAD, Mechanical CAD, RF). A more detailed explanation of modeling semantics can be found in (Simko *et al.*, 2013) and (Lattman 2012). The tangible impacts are illustrated in Figure 2:

- Components form reusable design artifacts, encompassing cross domain, executable models.
- System design options, e.g. for a flexible product line, are captured in a single model as Design spaces, and can be readily elaborated and explored, via automated analysis tools.
- Design models adaptable, where new concepts and technologies can be inserted into a model and fully evaluated, computationally.

- Design cycles decrease, due to automation and pervasive, cross-disciplinary modeling and simulation-based validation, using a range of trusted, high performance, existing modeling and simulation tools.

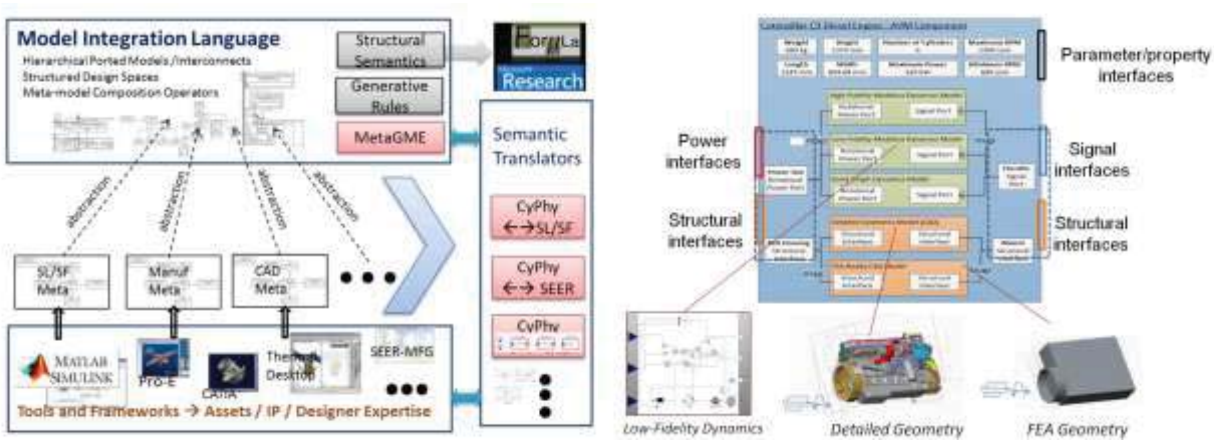

Figure 1: a) Semantic Backplane and b) example component model.

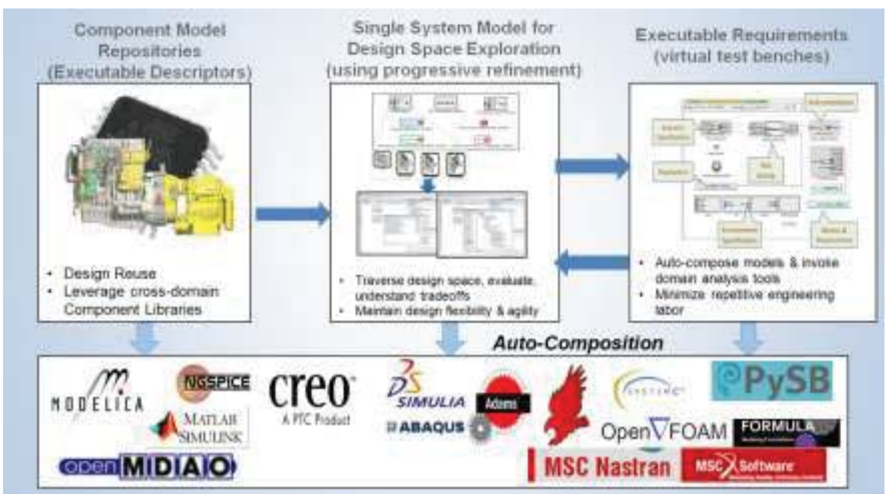

Figure 2: OpenMETA Component, Design Space, and Test Bench.

2.2 Example Design Study

Consider the conceptual design of a spacecraft. In its simplified mission cycle, it will spend the majority of time in an energy collection mode, where the solar cells are pointed at the sun. During this time, any data collection and processing is done. Periodically, the craft will rotate 90 degrees to point the antenna at a receiving station on earth and transmit data. During this time, the system will operate from stored power.

The designer must select a set of components, subsystem sizes, control parameters, and physical placements in order to meet a set of system requirements, including:

- Physical dynamics: orienting the system for energy collection and data transmission at the programmed time, repetitively,
- Providing consistent power for the system across all operational cycles, via storage and Photo Voltaic (PV) arrays,
- Maintaining thermal limits, ensuring that the cooling system is sufficient to avoid overheating of any component at any point in the cycle.

To evaluate the system, and all design options, we begin by creating a (conceptual level) system design space model, shown in Figure 3.

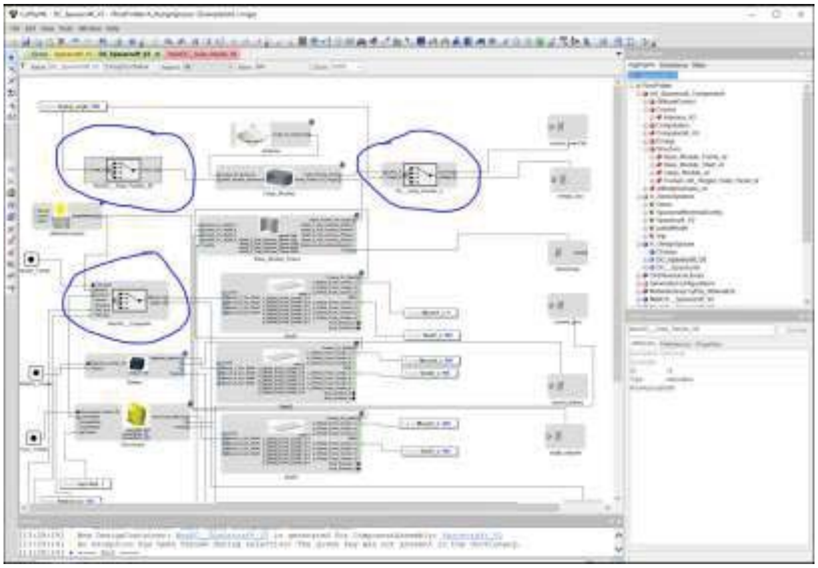

Figure 3: Design Space for Spacecraft System.

The system contains a range of components to cover key functions including structural (cargo bay, body, mounting shelf), energy management (battery, PV arrays), force generation (control-motion gyroscope), and control (computer, antenna). To specify the options for some of these components, the model contains alternatives for PV arrays (2,3, 4 panel) and computer(7 alternatives).

To help the designer visualize the physical meaning of the topological model, the elaboration of a model can be done interactively, via a tool named META-Link, as shown in Figure 4. A single design point configuration is shown in the figure, using the component's connectors to form mechanical constraints, illustrating the composition of the components and connections into a three dimensional representation of the models. The currently supported CAD tool is PTC Creo, however the component connection semantics are common to all parametric, constraint-based CAD package. Issues with automated CAD assembly are discussed in (Wren et al, 2012)

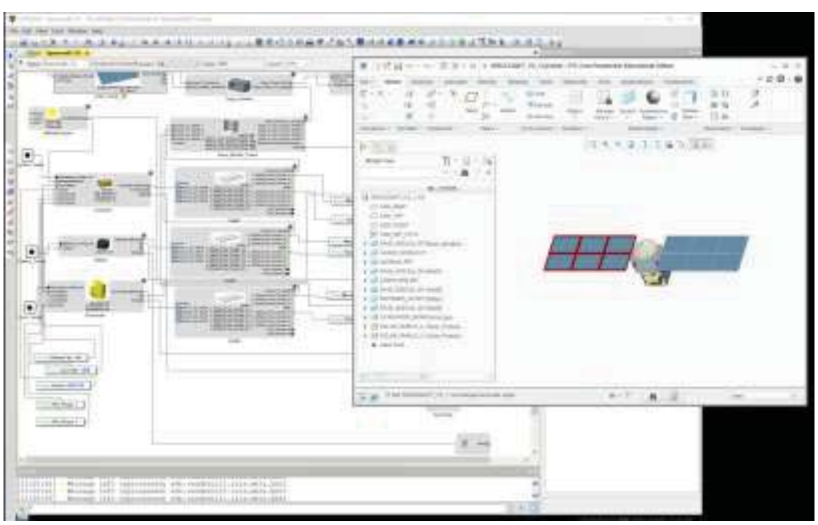

Figure 4: META-Link, Topology and Solid Model Views.

In this example, the component models capture behavioral dynamics and 3D structure. The Gyroscope is shown in Figure 5 below.

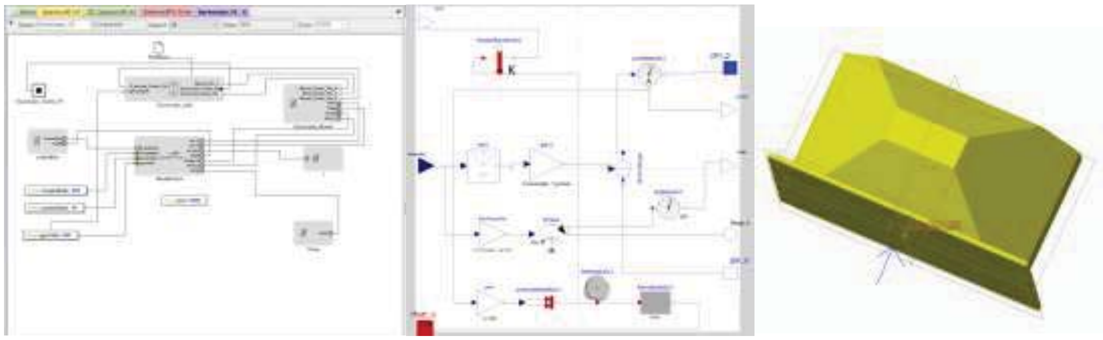

Figure 5: Gyroscope Model, with Behavior and Solid Model Artifacts.

Elaborating the model choices is highly automated using the design space exploration tool. This tool can elaborate the entire combinatorial space, or a focus on a subset of component/architecture choices, based on designer-specified constraints. In this study, the initial design space contains 27 alternatives, later reduced to 3 by applying constraints.

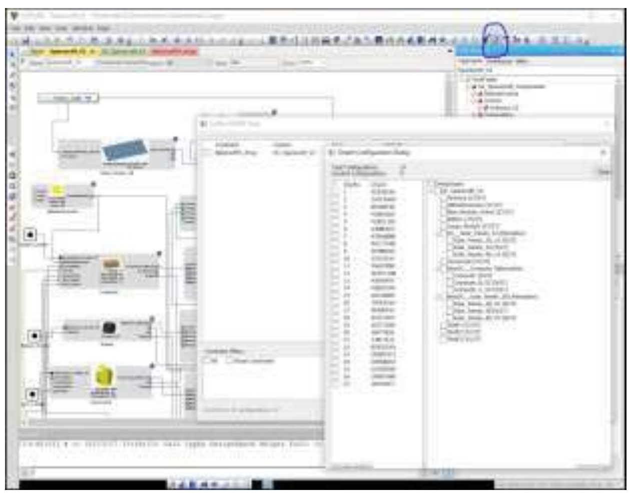

Figure 6: Evaluation of The Design Space.

To automate evaluation of a system, OpenMETA uses Test Bench Models, and workflows of these in parametric studies. A Test Bench is a formal specification for how to evaluate a set of requirements, as system metrics. An example of a testbench is shown Figure 7, which specifies composition of the system models into an executable Modelica model (simulated using OpenModelica) (Lattmaan et al, 2014), with a set of externally controllable parameters (inertia, PID controller parameters, battery size, gyroscope size), along with a specification for a set of metrics to be computed as a result of the simulation (Max/Average pointing error, min/max battery voltage, min/max temperatures, etc.) . Note that when the system under test is a Design Space, implying that the analysis campaign should be executed across all of the valid configurations in the design space.

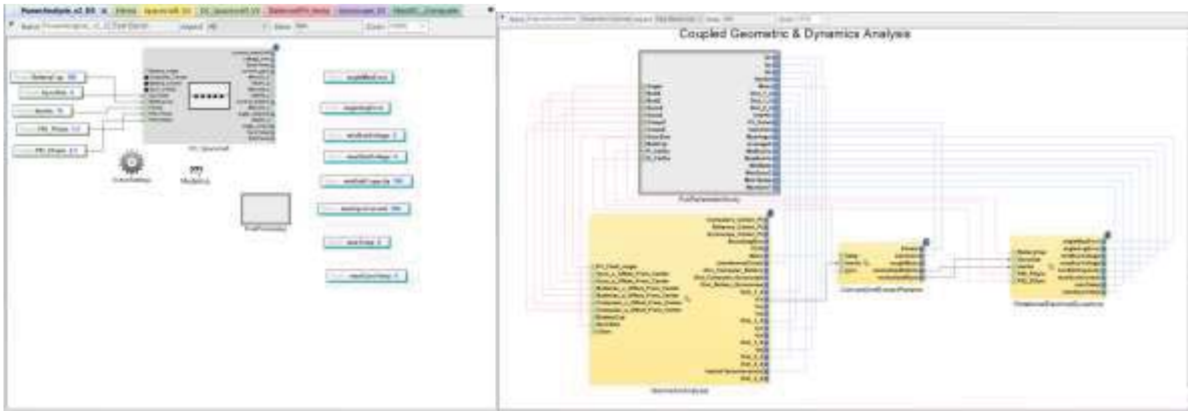

Figure 7: a) Test Bench Model and b) Parametric Exploration Model.

Test Benches are typically used inside a Parametric Exploration Tool model, as a workflow. An example workflow is shown below. This executes a Multi-Disciplinary Analysis and Optimization (MDAO) across a set of Design Variable ranges, while collecting a set of Objective variables. This is a common mode of analysis, supported by tools such as iSight and ModelCenter. We use the open source OpenMDAO as a target execution framework.

Bypassing many details, the models are virtually executed with a single operation of the tools, automatically simulated across the entire parametric and discrete design space, and presented for analysis. For this analysis, over 6000 unique CAD models were created, analyzed for interference, inertia, mass, and bounding box, most of which are dependent on component sizing (battery, gyroscope, and computer), along with the specific placement of these components within the spacecraft body. The computed geometric parameters drive another 6000 dynamic simulations, where the system behavior was simulated over 3000 seconds of system operation, computing voltages, currents, torques, position, velocity, and temperatures of the system. For each simulation, the Test Bench metrics were automatically computed, as specified in the OpenMETA models, and logged into a results database. Each of the model evaluations generates a full time history, with all internal variables computed over time, from which metrics are computed Typically, these are not stored, except when a deeper system inspection is needed, or when validating models.

The results are explored using the OpenMETA visualizer. This tool leverages the open source package R/Shiny to produce a customizable set of data exploration functions. A typical progression through the exploration of the system's simulation result data space is shown in Figures 8 and 9.

Broad exploration of entire data space, via cross parameter plots, colorized by design variables and system metrics, shown in Figure 8. These are filtered by mapping system requirements into numerical constraints (i.e. minimum bus voltage, maximum temperature). These constraints effectively filter out non-compliant design points.

The visualizer supports the ability to zoom in and select design points for further review or design elaboration, along with methods of comparing a wide variety of design points, based on composite metrics (Figure 9). Design studies can also consider probabilistic evaluation across component parameter ranges, to assess the likelihood of the system meeting design metrics, given component variation, dubbed Probabilistic Certificate of Correctness.

Figure 8: Visualizing Results Database, with Requirement-Driven Filters.

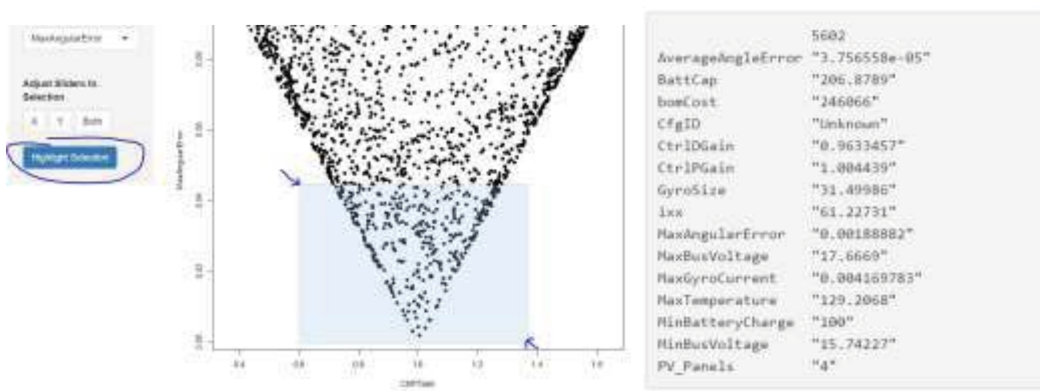

Figure 9: a) Data Exploration, and b) Selecting a Design Point.

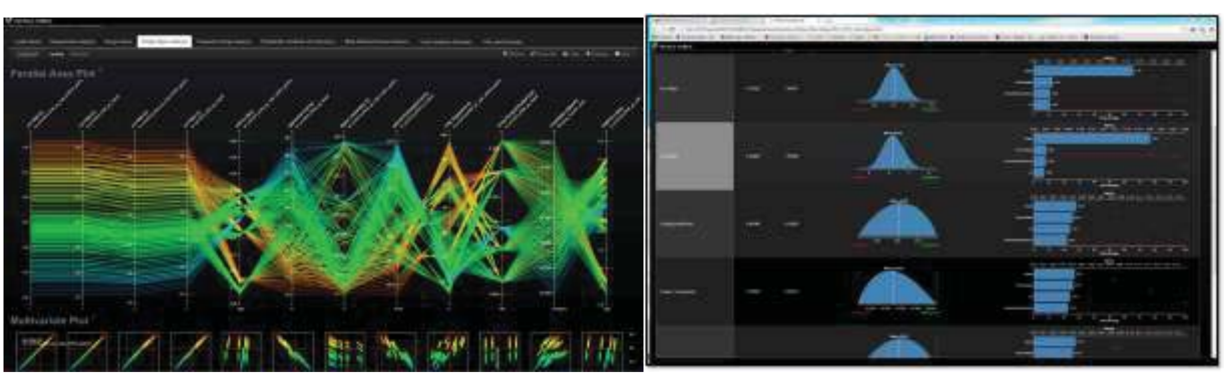

Figure 10: a) Simultaneous Visualization of Designs and b) Probabilistic Certificate of Correctness.

In addition, we can perform a ‘dive-down’ operation within the visualizer. Using the PET refinement tab, new design variable ranges can be specified, along with subsets of the discrete configurations, and sampling methods, Figure 11. This facility allows higher level systems engineers to benefit from a wide range of simulation models and domain analyses, all within the context of the data exploration/results visualization tools.

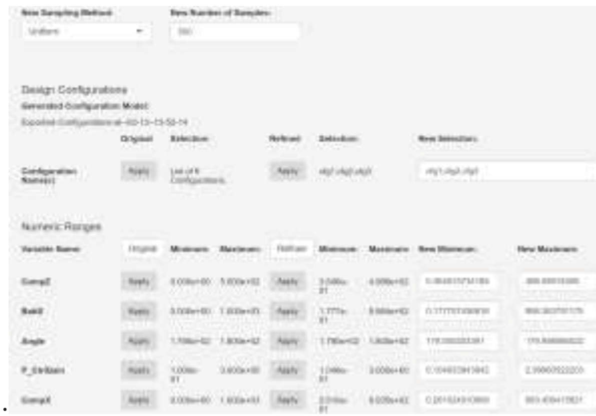

Figure 11: Requesting Simulations from the Visualizer.

Drilling down to a deeper level of abstraction, we evaluate and optimize the design of a physical geometry component. The tools manage the execution of finite element analyses (FEA), in this case MSC NASTRAN. For the design study, a test bench and PET was created for the assessment of the battery load on the mounting shelf, based on launch acceleration loads. The test bench describes the loads and constraints on the sub-assembly (i.e. shelf and battery) and the PET iterates over dimensions (e.g. shelf height) of the shelf and submits an FEA for each dimensional variation. Figures showing the test bench and PET follow:

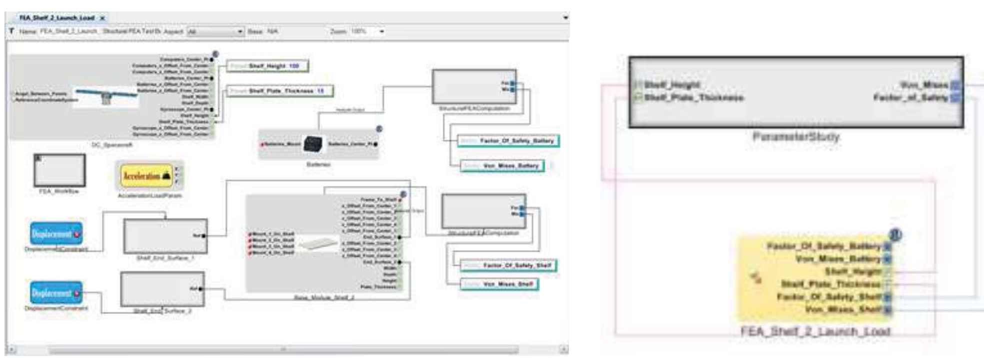

Figure 12: a) Test Bench for Finite Element Analysis and b) FEA Design Study.

The following figures show the FEA mesh with loads/constraints and the resulting Von Mises stress contour plot for one dimensional variation:

Figure 13: a) FEA Model Load Conditions and b) Analysis Results.

The above FEA PET was run over 50 various shelf heights and the results are presented in the following diagram. The goal is to select the minimum shelf-height (and thus the minimum weight shelf) with an acceptable factor-of-safety.

The results, in Figure 14, show that a shelf thickness of 15 mm and height of 67 mm is sufficient to support the mass during launch, allowing the designer to minimize total system weight.

This design study shows a fully automated analysis of a complex system, using a single model, across multiple levels of abstraction. Geometric analysis was coupled with ODE-based dynamics models incorporating electrical, thermal, and mechanical analysis. These properties were evaluated over a broad design space, with a refinement into a small parametric range. Finally, the tools supported optimization of one of the components for lightweighting using detailed finite element analysis. Aside from execution time, engineering time and effort is minimized.

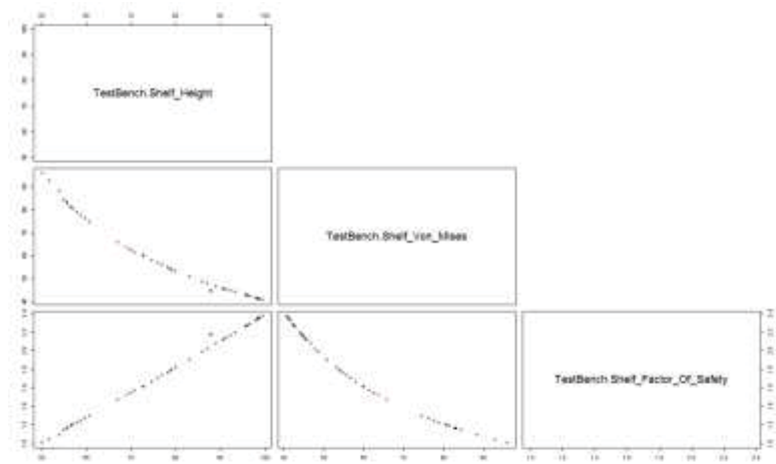

Figure 14: FEA Design Study Results.

This design study exercised a small fraction of the tools integrated into OpenMETA. The table below elaborates many of the tools used across different application areas.

Table 15: Tools used across different application areas.

Technology	**Description**	**Developer**	**License**
META-Link	Synchronization between CAD & Model	VU	Open Source
OpenModelica	Lumped Parameter Dynamic Sim. (ODE)	OpenModelica.org	Open Source
Dymola	Lumped Parameter Dynamic Sim. (ODE)	Dassault	Proprietary
Simulink	Controller Modeling & Synthesis + ODE	Mathworks	Proprietary
CREO	CAD Models, Product Definition	PTC	Proprietary
Abaqus	FEA: Structural, Thermal, Fatigue, Modal	Dassault	Proprietary
NASTRAN	FEA: Structural, Thermal, Fatigue, Modal	MSC	Proprietary
PATRAN	Model Preprocessing, Meshing, Composites	MSC	Proprietary
Calculix	FEA: Structural, Thermal,	Calculix.de	Open Source
ANSYS	FEA: Structural, Thermal, Fatigue, Modal	ANSYS Inc	Proprietary
OpenFOAM	CFD Analysis	OpenFOAM	Open Source

OpenCascade	Survivability Analysis, Ballistics	Open Cascade	Open Source
LS-DYNA	Survivability Analysis, Blast	LSTC	Proprietary
NRMM	NATO Reference Mobility Model	DOD	NATO
AF DATCOM	Aerodynamics, Lift, Drag, Forces	DOD	DOD
Manufacturability	iFAB Manufacturability Tools	Penn State	Open Source
FoV, FoF	Field of View/Fire for Military Vehicles	Ricardo	Open Source
Ergonomics	Egress for Crew	Ricardo	Open Source
SPICE	Electrical Circuit Analysis	NGSPICE	Open
SystemC	Digital & Software Timing Simulation	Accelera	Open Source
EagleCAD	Electronics CAD, Schematics &PCB Layout	Autocad	Proprietary
AutoPlace	Constraint-Based Chip Placement	Metamorph	Open Source
EBOM	Electronic BOM and Source/Pricing	Metamorph	Open Source
OpenEMS	Electromagnetic Systems Analysis	OpenEMS	Open Source

3 CONCLUSIONS

We do not create new simulation/analysis tools, but make them easier/cheaper to use. Maintenance of a single model is an important contribution to minimizing effort, as is the time saved via automatic composition of analysis tool models. The tools have been applied by several design teams for a variety of system types and design abstractions:

- Electronics: Design of low-power, wearable IoT devices and of option modules for Google's Project Ara modular phone, addressing analog and digital simulation, synthesis of detailed schematics and printed circuit boards, and RF analysis of antenna/assembly performance. (Google Project Ara)
- Defense: design of an amphibious assault vehicle, addressing drivetrain performance, structural loads, terrain mobility, survivability, and human factors. (Myers et al, 2013)
- Aerospace: Missile system conceptual design studies, and design processes for aircraft, addressing aerodynamics and system performance (Larkin et al, 2016)
- Commercial vehicles: Passenger cars, including performance and human-in-the-loop testing. (Eisele et al, 2016)
- Modular Robotics: Rapid design and assessment of robotics systems from a flexible set of robotic components.
- Systems analysis and cybersecurity, modeling physical and software behaviors.
- Systems Biology, adapting the concepts to simulating cell pathways.

The tools have proven to be remarkably versatile, due to the ability to evolve the language, and the commonality of the core infrastructure. Extensions of the language have allowed new types of connection semantics and component domain artifacts. Supporting a new analyses requires effort, primarily in creation of composition tools, but the core functions involved with targeting a new tool share common graph-processing operations. While the tools continue to be extended, they have proven to be useful in their current form. The tools and documentation are available on github https://github.com/metamorph-inc/meta-core and www.isis.vanderbilt.edu/openmeta.

REFERENCES

Simko G, Levendovszky T, Neema S, et al. Foundation for Model Integration: Semantic Backplane. ASME. International Design Engineering Technical Conferences and Com-puters and Information in Engineering Conference, Volume 2: 32nd Computers and In-formation in Engineering Conference, Parts A and B ():1077-1086. doi:10.1115/DETC2012-70534

Simko, Gabor, et al. "A framework for unambiguous and extensible specification of DSMLs for cyber-physical systems." Engineering of Computer Based Systems (ECBS), 2013 20th IEEE International Conference and Workshops on the. IEEE, 2013.

Zsolt Lattmann, Adrian Pop, Johan de Kleer, Peter Fritzson, Bill Janssen, Sandeep Neema, Ted Bapty, Xenofon Koutsoukos, Matthew Klenk, Daniel Bobrow, Bhaskar Saha, Tolga Kurtoglu, Verification and Design Exploration through Meta Tool Inte-gration with OpenModelica, Proceedings of the 10th International Modelica Confer-ence; March 10-12; 2014; Lund; Sweden, Volume , Issue 96, 2014-03-10, Pages 353-362, ISSN 1650-3740

Sztipanovits, Janos, et al. "Design tool chain for cyber-physical systems: lessons learned." Design Automation Conference (DAC), 2015 52nd ACM/EDAC/IEEE. IEEE, 2015

Lattmann, Zsolt, et al. "Component-based modeling of dynamic systems using heterogeneous composition." Proceedings of the 6th International Workshop on Multi-Paradigm Modeling. 2012.

Wrenn, Ryan, et al. "Towards automated exploration and assembly of vehicle design models." ASME 2012 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. American Society of Mechanical Engineers, 2012.

Lattmann, Zsolt, et al. "Towards an analysis-driven rapid design process for cyber-physical systems." Rapid System Prototyping (RSP), 2015 International Symposium on. IEEE, 2015

Larkin, Dale, et al. "Ontology-Driven Metamodel Validation in Cyber-Physical Systems." AIAA Modeling and Simulation Technologies Conference. 2016.

Myers, Michael R., et al. "Collaborative Design in DARPA's FANG I Challenge." ASME 2013 International Mechanical Engineering Congress and Exposition. American Society of Mechanical Engineers, 2013.

Eisele, Scott, et al. "ADAS Virtual Prototyping with the OpenMETA Toolchain." SAE International Journal of Passenger Cars-Electronic and Electrical Systems 9.2016-01-0002 (2016): 22-29.

AUTHOR BIOGRAPHIES

THEODORE A. BAPTY is a Research Associate Professor at the Institute for Software Integrated Systems, Vanderbilt University. He holds a PhD in Electrical and Computer Engineering from Vanderbilt University. His research interests lie in Model-Based Systems, Simulation-Based Design Tools, primarily for complex cyberphysical systems.

JASON SCOTT is a Research Program Manager at Vanderbilt University. He holds a Ph.D. in Electrical and Computer Engineering from Vanderbilt University. His research interests include Model-Based Engineering and Formal Verification, and applications in electronics design and IoT.

SANDEEP NEEMA is a research associate professor at Vanderbilt University, currently on assignment at DARPA. His research interests include Cyber Physical Systems, model-based language semantics, and big Data in Science and Engineering.

ROBERT L. OWENS is a System Architect at the Institute for Software Integrated Systems, Vanderbilt University. He holds a M.S. Computer Science from Florida Institute of Technology and a B.S. Mechanical Engineering from Tennessee Technological University. His research interests include the application of modeling and automation to consider more design options, shorten design cycles, and improve overall quality.