

Institute for Software-Integrated Systems

Technical Report

TR#: **ISIS-15-102**

Title: **The META Toolchain: Accomplishments and Open Challenges**

Authors: **Janos Sztipanovits, Ted Bapty, Sandeep Neema, Xenofon Koutsoukos and Jason Scott**

This research is supported by the Defense Advanced Research Project Agency (DARPA)'s AVM META program under award #HR0011-13-C-0041.

Copyright (C) ISIS/Vanderbilt University, 2015

Table of Contents

List of Figures	3
1. AVM Vision	4
2. Status Quo.....	5
3. Key Findings and Accomplishments.....	6
3.1. Shorten Development Times for Complex Defense Systems	6
Finding 1. Need for Integration Platforms.....	6
Accomplishment 1. OpenMETA Horizontal Integration Platforms	7
Accomplishment 2. Semantic Integration	8
Finding 2. Need for Component Modeling Technology	9
Accomplishment 3. AVM Component Model and Curation Process	11
Finding 3. Need for Automation in Design Flow	11
Accomplishment 4. Design-Space Exploration Using Progressive Refinement.....	12
Accomplishment 5. Automated Analysis Using Virtual Test Benches.....	12
Accomplishment 6. Deterministic and Probabilistic Verification Tools	13
Accomplishment 7. Real-time Software Implementation Tool Suite	14
Accomplishment 8. Design Space Analyzer and Visualizer	14
3.2. Shift product value chain toward high-value design activities	15
Finding 4. Need for Product and Manufacturing Process Co-Design	15
Accomplishment 9. Design-Space Exploration with Manufacturability Test.....	16
3.3. Democratize design.....	16
Finding 5. Need for Collaboration Platforms and Access to Open- Source Tools.....	17
Accomplishment 10. VehicleForge – OpenMETA Integration	18
Accomplishment 11. Open-source Tool Configuration and Cloud-based Deployment	19
4. Opportunities and Open Challenges	21
4.1. Short Term Opportunities	21
4.2. Medium Term Challenges.....	22
5. Transitioning.....	23
6. Acknowledgment.....	23



List of Figures

Figure 1: OpenMETA Horizontal Integration Platforms.....	7
Figure 2: Notional Design Flow in META.....	11
Figure 3: Probabilistic Certificate of Correctness Details.....	14
Figure 4: VehicleForge- OpenMETA Integration.....	19



Tel (615) 343-7472 Fax (615) 343-7440
1025 16th Avenue South Nashville, TN 37212
www.isis.vanderbilt.edu



1. AVM Vision

The goal of the Adaptive Vehicle Make program of DARPA/TTO was to substantially improve the design, manufacturing, and verification of complex cyber-physical systems (CPS)¹. While the primary application domain was military amphibious ground platforms, the program called for solutions that can be equally applied in other air or ground platforms as well. The central tenet of the program was that introducing component- and model-based methods in the design of defense systems will revolutionize their make-build process similar to the transformation of VLSI design methods over two decades ago². The overall vision for the program consists of three main elements:

1. Shorten development times for complex defense systems

One important metric that can be translated into cost and affordability of capabilities is development time. By analyzing trends in defense acquisition data and comparing them with product life-cycle data in the commercial VLSI and automotive industries, the program established that a 5X decrease of average development time is achievable. This goal was translated into the following technical challenges: (a) composing designs from component model libraries, (b) raising the level of abstraction in the design of CPS, (c) enabling correct-by-construction design methods, and (d) executing rapid requirements trade-offs.

2. Shift product value chain toward high-value design activities

Adoption of component- and model-based design will lay the foundation for establishing a well-defined interface between design and manufacturing. Moving this interface toward design by incorporating manufacturing awareness into the design flows has profound impacts on the product value chain and enables new business models and new capabilities, such as (a) foundry-like manufacturing capability for defense systems, (b) rapid switch-over between designs with minimal learning curve, and “mass customization” across product variants and families.

3. Democratize design

The emergence and further maturation of open platforms and open source tools has the potential for dramatically expanding the scope of players within defense innovation processes and disrupting old business models. The program intends to seed this restructuring by (a) building on a dominantly crowd-sourced tool infrastructure (called OpenMETA) to enable open-source development of cyber-electromechanical systems, (b) using the OpenMETA tools to be developed by the program for experimenting with prize-based design challenges to involve non-traditional players in the make process, and (c) motivating a new generation of designers and manufacturing innovators by initiating student design competitions.

¹ DARPA-BAA-10-21

² Paul Eremenko: Philosophical Underpinnings of Adaptive Vehicle Make



The vision above made the META design tool component of the AVM program centrally important for achieving key program goals. At the same time, the vision has established hard technical challenges that gave opportunities for DARPA researchers not only to measure the effectiveness of the state-of-the-art in component- and model-based design in a complex design domain, but also to extend the limits both in foundations and practice.

2. Status Quo

The defense industry, and large system design and manufacturing companies in general, face immense pressures to deliver safe and complex systems at low cost. Software tools are at the heart of their engineering process covering the full spectrum of requirements, design, manufacturing and operations support. The internal tool landscapes of large aerospace and automotive companies contain ~5000 distinct tools totaling several hundreds of millions of dollars in internal investments. End-to-end software tools for these complex CPS product lines are very heterogeneous and span too many technical areas for individual tool vendors to fully cover. In addition, a significant part of the companies' design flow is supported by in-house tools that are proprietary and capture high value design IP. In many areas, such as powertrain electronics in the automotive industry, production tool suites include a combination of in-house and COTS tools in the approximate ratio of 70% and 30%, respectively³. The development and use of in-house tools is not necessarily the result of deficient COTS offerings, but, rather, it is an essential part of the innovation process that yields competitive advantage via improved product quality and productivity. The primary technology barrier that slows down this process and makes integration of in-house tools with 3rd party tools extremely expensive and error prone is the lack of modern software tool integration and deployment platforms.

Seamless integration of end-to-end tool chains for highly automated execution of design flows is a complex task, of which successful examples are rare – even after massive investment by companies. Vendors provide limited integration, primarily of their own tools, with a few cross-vendor integrations for particularly dominant tools (e.g., integration with DOORS, Word or Excel). This limitation results in design flows that consist of islands of integrated tool sub-chains, bridged by various ad-hoc, semi-automated, or manual stopgaps. These stopgaps impose a variety of costs: additional work in performing manual transformations, additional work in guarding against divergence between multiple representations, and subsequently, forgone analysis opportunities, to name just a few.

Truly transformational impact requires an approach for synthesizing an end-to-end integrated tool chain from a heterogeneous collection of COTS, open source, and proprietary tools. The ideal solution would support tools from multiple vendors, and allow companies themselves to include the most closely guarded of proprietary tools. Such a truly integrated toolset would yield significant improvements in productivity and decreases in design time, by eliminating the

³ Boeing, GM, Microsoft Research, MetaMorph Inc, Vanderbilt: “Comprehensive Transitioning Model for AVM Tools” joint response to DARPA RFI DARPA-SN-14-04, 2013



unnecessary work associated with the existing integration mechanisms and shortening the learning curves associated with diverse, un-integrated tool suites.

3. Key Findings and Accomplishments

The fundamental barriers in the META project for developing the OpenMETA design flow and its supporting tool suite have been (a) heterogeneity and semantic ambiguity of models and tools that span the design and manufacturing space, (b) the lack of integration technology for models and tools, available only in isolated stove pipes, and (c) the lack of availability of delivery platforms that break down the cost and complexity of using an integrated tool suite. This section summarizes key findings and accomplishments of the META design tool development effort of the AVM program.

3.1. Shorten Development Times for Complex Defense Systems

The most significant source of long development times for complex CPS has been the limited predictability of system properties “as manufactured” during the design process. A typical characteristic of the current systems engineering practice is that limited predictability forces the development process to iterate over lengthy design→build→test→redesign cycles until important requirements are achieved. There are three fundamental contributors to radically shortening systems development time: (a) changing the level and scope of *abstractions* in the design flow (Accomplishment 4) and incorporating new technologies for *correct-by-construction design* (Accomplishment 6), (b) *reusing design knowledge* from component model libraries, and (c) introducing *automation* in the design flow for executing rapid requirements evaluation and design trade-offs. The META project addressed all three of these technology components, with capabilities included in the OpenMETA tool suite.

Finding 1. Need for Integration Platforms

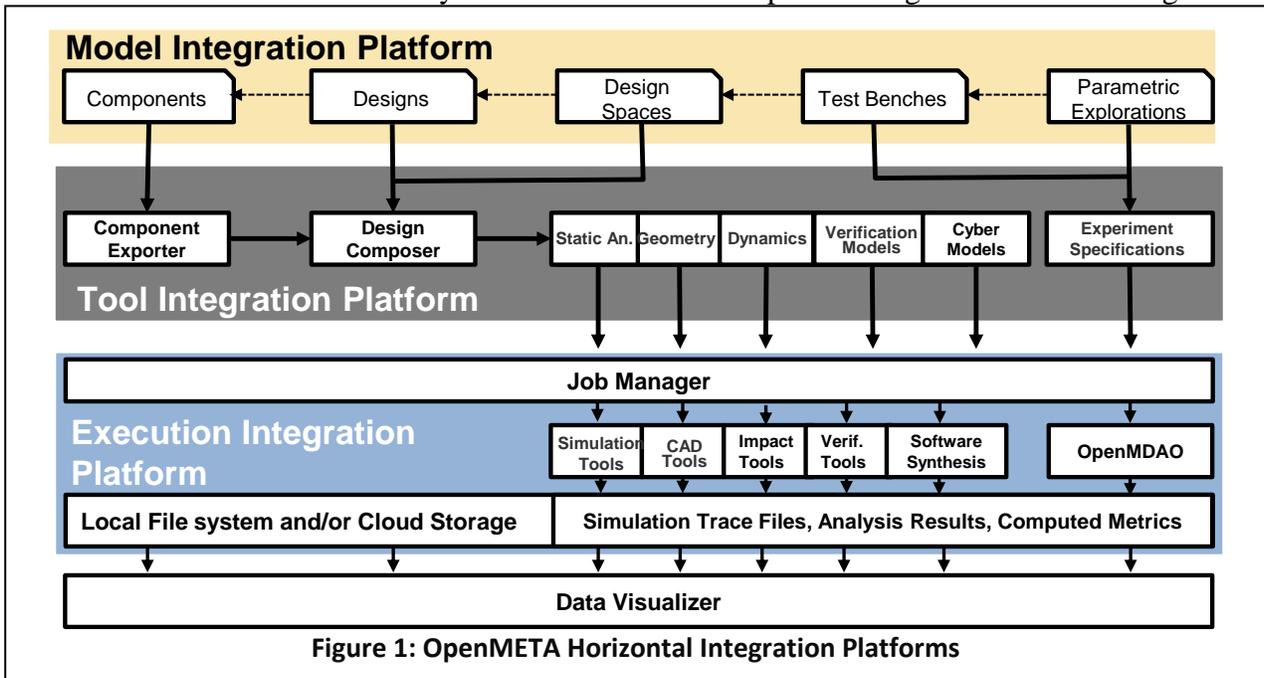
While model-based design has a proven track record and strong acceptance in many focused areas of engineering (such as VLSI design, control system design), the heterogeneity of CPS technologies and application domains, combined with the need for achieving correct-by-construction design, create new technical barriers for its wider use. The most widely used strategy to deal with heterogeneity in the design process is *separation of concerns*. Its goal is to decrease design complexity by decomposing the overall design problem according to physical phenomena (electrical, mechanical, thermal, structural, etc...), level of abstraction (static, lumped parameter dynamics, distributed parameter dynamics, etc...) or engineering discipline (performance, systems engineering, software engineering, manufacturing, etc...). Negative consequences of this design strategy are quite significant, both in terms of weakening the opportunity for correct-by-construction design, as well as missing out on potential cross-domain optimizations in CPS design flows. The chief reason is that discipline-oriented design flows usually overlook modeling interactions/interdependences among the various design views. The separation approach could work if the design concerns were orthogonal, but in tightly coupled CPS, this is rarely the case. The price of the simplification is decreased predictability of properties of the implemented CPS and costly re-design cycles.



We believe that the single most important change necessary to achieve correct-by-construction design is the introduction and systematic use of *cross-domain modeling*. However, creating design tool chains that cover all potentially relevant CPS modeling abstractions and satisfy the needs of all application domains, is unrealistic. In addition, tool chains that are highly configurable to specific application domains are not available. Consequently, our objective was to develop horizontal integration platforms that allow the rapid construction of domain-specific, end-to-end tool suites for CPS design.

Accomplishment 1. OpenMETA Horizontal Integration Platforms

OpenMETA complements the traditional, vertically-structured and isolated model-based tool suites with horizontal integration platforms for models, tools, and executions. The horizontal integration platforms allow combining the separation of concerns strategy with cross-domain modeling whenever domains are not orthogonal and cross-domain interdependences cannot be neglected. Our focus in the project was not restricted to OpenMETA as configured for a specific ground platform design, but was extended to the OpenMETA Integration Platforms for models, tools and executions such that they can be used both for experimenting with different design



flows and for creating highly domain specific design tool chains.

These integration platforms are the following:

1. *Model integration platform*, supported by generic OpenMETA tools for creating and using semantically rigorous model integration languages, metaprogrammable modeling tools, metamodel repositories and the OpenMETA Semantic Backplane including formal specification of the model integration language ChyPhyML and all model transformations (Accomplishment 2).

2. *Tool integration platform*, with generic tools for the precise specification, verification and generation of model transformations – a widely used technology in the OpenMETA tool chain. The tool integration platform also includes the specification of design flows (Experiment Specifications) composed from predefined design threads and vignettes.
3. *Execution integration platform*, to provide an affordable, web-based delivery platform of integrated design tools, enabling their cloud-based deployment through a software-as-a-service delivery model. The platform includes job manager for distributing computation intensive requirements evaluation and probabilistic certification tasks across a highly scalable set of computational resources.

The OpenMETA design tool chain has been delivered as a fully configured integrated tool suite with model libraries and has been tested in DARPA’s Fast Adaptable Next-Generation Ground Vehicle (FANG1) Mobility/Drivetrain Design Challenge (2013) and Chassis and Survivability Design Challenge (2014, Gamma Test). During the FANG1 challenge, the tool suite was stress tested in a national design competition for the power train of an amphibious vehicle that included over 1000 participants in more than 200 design teams in 2013.

Accomplishment 2. Semantic Integration

In META, as in all approaches to model-based design, modeling languages and their underlying semantics play a fundamental role in achieving compositionality. Heterogeneity of the multi-physics, multi-abstraction and multi-fidelity design space, and the need for rapidly evolving/updating design flows, require the use of a rich set of modeling languages influenced/determined by both existing and emerging model-based design, verification and simulation technologies and tools. Consequently, the language suite and related infrastructure cannot be static; it will continuously evolve. To address both heterogeneity and evolvability simultaneously, we have departed from the most frequently-used approach to address heterogeneity: the development or adoption of a single, very broad and necessarily hugely complex language standard designed for covering all relevant views of multi-physics and cyber domains. Instead, we placed emphasis on the development of a model integration language – CyPhyML – with constructs limited to modeling the interactions among different modeling views, yet easily expandable.

In a naïve approach, model and tool integration is considered to be an interoperability issue between multiple models that can be managed with appropriate syntactic standards and conversions. In complex design problems these approaches inevitably fail due to the rapid loss of control over the semantic integrity of set of diverse models involved in real design flows. The “cost” of introducing a dynamic model integration language is that mathematically precise formal semantics for model integration had to be developed under OpenMETA.

The OpenMETA Semantic Backplane is at the center of our semantic integration concept. The key idea is to define the semantics of the CyPhyML model integration language using formal metamodeling, and to use a tool-supported formal framework for updating the CyPhyML metamodels and verifying its overall consistency and completeness as the modeling languages



are evolving. The selected tool for formal metamodeling is FORMULA⁴ from Microsoft Research. FORMULA's algebraic data types (ADTs) and constraint logic programming (CLP) based semantics are effective at mathematically defining modeling domains, transformations across domains, as well as constraints over domains and transformations. At the conclusion of the project, the META Semantic Backplane includes the formal specification of CyPhyML, the semantic interfaces to all constituent modeling languages, and all model transformations used in the tool integration framework. (The size of the specifications is 19,696 lines out of which 11,560 are auto-generated and 8,136 are manually written.)

The Semantic Backplane is a pioneering approach to construct complex component- and model-based design tool chains. It is an essential tool for those who design and evolve domain specific tool chains and responsible for the overall integrity of the model and tool configurations used in the design process. Its importance was proven in the following use cases:

1. As in all areas of engineering, mathematical modeling helped designing and evolving modeling languages, composition semantics and model transformations. It was invaluable in finding and correcting inconsistencies, identifying incompleteness problems, and fixing errors in the semantic foundations of the tool chain.
2. The Formula-based executable specifications were used for generating reference traces and served as abstract prototypes for constraint checkers and transformations used throughout the tool chain.
3. The CyPhyML Reference Manual was auto-generated from the formal specifications.

While most of the activities in the use cases above are manual at this point, creating tighter link between the specification and the production tools and increased automation such as auto-generation of transformation code from formal specification is feasible.

Finding 2. Need for Component Modeling Technology

The appeal of component-based design is the potentially massive productivity increase due to the reuse of the design knowledge that is captured by the component models. Their significance was recognized early in the AVM program and was noted as the main contributor to achieving the 5X decrease in design time⁵. In a component- and model-based design flow, system models are composed of component models guided by architecture specifications. To achieve correct-by-construction design, the system models must be heterogeneous multi-physics, multi-abstraction and multi-fidelity models that also capture cross-domain interactions. Accordingly, the component models, in order to be useful, must satisfy the following generic requirements:

1. Elaborating and adopting established, mathematically-sound principles for compositionality. Composition frameworks are significantly different for physical dynamics, structure and computing, and must be precisely defined and integrated.

⁴ <http://research.microsoft.com/formula>

⁵ O. L. de Weck, "Feasibility of a 5× speedup in system development due to meta design," in *32nd ASME Computers and Information in Engineering Conference*, Aug. 2012, pp. 1105–1110

2. Inclusion of a suite of domain models (e.g., structural, multi-physics lumped parameter dynamics, distributed parameter dynamics, manufacturability), on an established number of fidelity levels with explicitly represented cross-domain interactions.
3. Precisely defined component interfaces required for heterogeneous composition. The interfaces must be decoupled from the modeling languages used for capturing domain models. This decoupling ensures independence from the modeling tools selected by the component model developers.
4. Established bounds for composability expressed in terms of operating regimes where the component model remains valid.
5. Established and documented component model validity, since the use of non-validated component models makes model-based analysis meaningless even under the most rigorously defined composition mechanisms.

These requirements are widely accepted in all engineering design processes where component-based approaches are used. A common misconception in physical system modeling is that useful models need to be hand-crafted for specific phenomena. One explanation for this is the frequent use of modeling approaches that do not support generic compositionality. The AVM Component Model (Accomplishment 3) places strong emphasis on compositional semantics (Accomplishment 2) that resolve this problem. A harder problem is that automated composition from multi-phenomenon component models can easily produce very complex, high-order models, if incorrectly used. The solution is to support multiple phenomena, abstractions, and fidelities, and to adapt the selected level of abstraction, level of component fidelity and the suite of physical phenomena to the examined system property. While we applied this approach in META (Accomplishment 4) there are still open challenges (Section 4.1) to be addressed in the future. A well-known method for controlling design complexity is to adjust the level of granularity for components, and use more abstract models for larger, more complex components such as engines, transmissions or microprocessors. There are excellent examples for existing and emerging component libraries, both in crowdsourced or COTS form: DOE's EnergyPlus⁶ is an open-source model and simulation library for building energy analysis; the Modelica Standard Library (MSL)⁷ is a crowdsourced, multi-physics lumped parameter dynamics library developed and maintained by the OpenModelica Consortium; Modelon's Vehicle Dynamics Library⁸ is a COTS component library on the top of the Modelica Standard Library; and many others. We believe that domain specific model libraries will continue emerging both in open-source and COTS form and will become one of the engines in the progress of component and model-based design.

⁶ http://apps1.eere.energy.gov/buildings/energyplus/energyplus_addons.cfm

⁷ <http://www.modelica.org>

⁸ <http://www.modelon.com/products/modelica-libraries/vehicle-dynamics-library/>

Accomplishment 3. AVM Component Model and Curation Process

The META project developed a standard AVM Component Model that provides a framework for integrating multi-domain and multi-language structural, behavioral and manufacturing models into a Component, and provides the compositional interfaces for the OpenMETA tools. In constructing an AVM Component Model from domain models, (such as from Modelica models representing lumped parameter dynamics) the interfaces, connectors, and parameters must be extracted from the domain models, and mapped to the interface abstractions used in the AVM Component Model. This process can be time-consuming and error-prone. In order to improve productivity, the META program has developed a full tool suite for importing domain models (such as Modelica dynamic models), integrating them with standard AVM Component Model Interfaces, automatically checking compliance with the standard, and automatically checking model properties, such as restrictions on the types of domain models, well-formedness rules, executability, and others. Based on our direct experience, the automated model curation process resulted in orders-of-magnitude reduction in required user effort for building AVM Component Model libraries.

Finding 3. Need for Automation in Design Flow

As shown in Figure 2, CPS design in META is divided into the following main phases:

1. Architecture design of a combinatorial design space, with rapid exploration using static, finite-domain constraints and architecture evaluation.
2. Integrated multi-physics/cyber design, exploring design choices and optimizing parameters using quantitative, lumped parameter hybrid dynamic models, and incorporating both

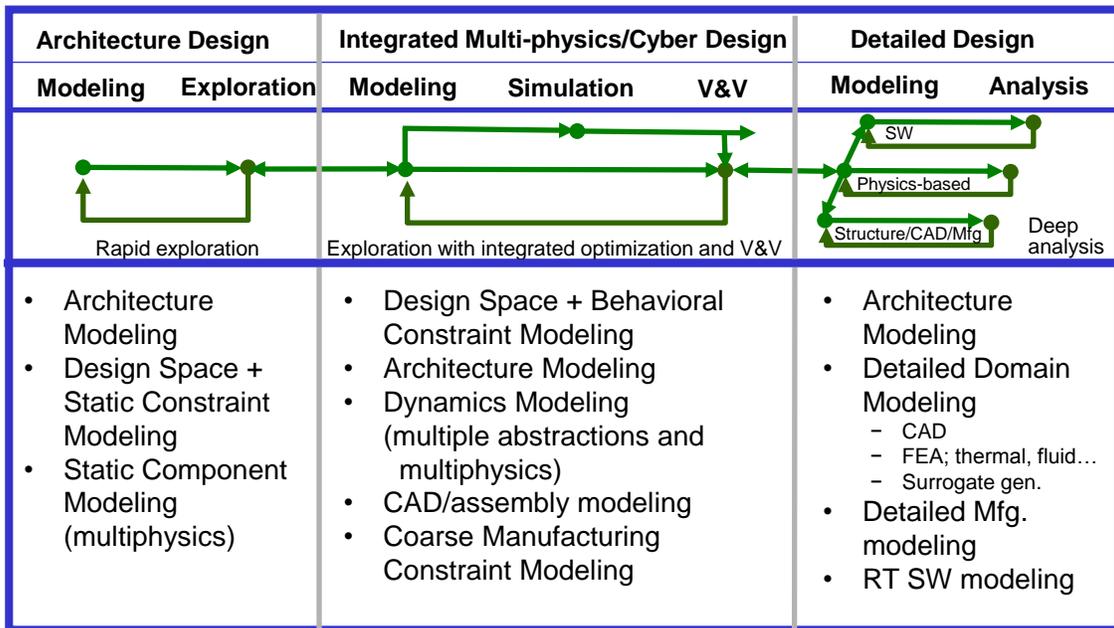


Figure 2: Notional Design Flow in META

deterministic and probabilistic approaches.

3. Detailed design including geometric/structural design space exploration using deep analysis with physics-based, nonlinear PDE analysis of thermal, mechanical and mobility properties.

The design space exploration phases require the composition of system models using model libraries, the analysis of the models against design requirements and the performance of a multi-objective optimization process combined with probabilistic and deterministic verification methods (Accomplishment 6). The META design flow must manage heterogeneity in multiple dimensions, such as physical phenomena, levels of abstraction used in modeling physical and computational structures and processes, and engineering disciplines involved in CPS design. If we combine this challenge with the need for exploring large design spaces, it is clear that without full automation of the exploration process, the overall META vision would not be achievable.

Accomplishment 4. Design-Space Exploration Using Progressive Refinement

Automated exploration of a heterogeneous CPS design space is not only semantically complex, but is also computationally expensive. Quality of the resulting design depends on the size of the explored space, which is determined by the number of architectural variants, the number of parameters, and the parametric ranges. However, executing the exploration process with the highest fidelity detailed design models is computationally prohibitive.

One of the key enablers for automating the design space exploration process is the automated adjustment of the level of abstraction of the composed system models starting with static models and combinatorial exploration of very large design spaces, progressing to lumped parameter dynamic models using different levels of fidelity, and finally performing first principle-based deep analysis for only a few candidate designs.

There are several exploration strategies that can be built in the OpenMETA design flow. We currently implemented the *progressive refinement strategy* that starts with a seed design (a single design point), around which the design space can be carefully defined by designers using architectural alternatives and parametrization.

The automated design space exploration process was stress tested in the FANG 1 Powertrain Challenge. During the 3 month competition period, design teams submitted for remote evaluation 51,424 candidate models and each received evaluation scores against a set of system requirements.

Accomplishment 5. Automated Analysis Using Virtual Test Benches

Another enabler for automating the design space exploration process is the fully automated evaluation of points in the design space against the full set of system requirements. The key OpenMETA innovation for this is the introduction of Virtual Test Benches that are the executable versions of the requirements. Each Test Bench is linked to the specification of a design space and used for evaluating the system performance metrics associated with requirements across all generated design point samples. While executing Test Benches during the exploration process, the design space continually evolves to include only those designs that



satisfy all requirements. Test Benches are also modeled using a modeling language that defines analysis tool setup, parameters, context models, metrics, and post processing scripts.

Accomplishment 6. Deterministic and Probabilistic Verification Tools

While OpenMETA is primarily a design automation infrastructure and open integration framework utilizing a large number of open-source and COTS analysis and verification tools, the META project also included important development efforts performed by Modelon, SRI, Oregon State University and PARC researchers to add specific model libraries and integratable, formal verification technologies to the design flow.

Formal verification is a technique for checking correctness of a system design that is complementary to simulation. Deterministic formal verification techniques work symbolically, rather than on concrete numbers, and hence, are able to reason about all possible behaviors of the system design in all possible environments. They provide the highest level of guarantee about correctness, and they can find errors that exhaustive simulation can miss. They are especially important for safety-critical systems. Two important contributions to deterministic formal verifications that have been fully integrated in the META design flow are the following:

1. **Modelon Inc.** has developed a fully equation-based, symbolic version of the FANG Modelica component libraries that the formal verification tools are able process. The symbolic version of the FANG model library has been integrated with the AVM component models as a distinct fidelity level and can be selected during composition.
2. **Stanford Research Institute (SRI)** has developed a relational abstraction-based verification tool for verifying safety properties of cyber-physical systems using hybrid dynamics. The SRI tool was integrated into the META design flow using a virtual test bench. If a violation of a desired property is found, the SRI tool produces a counter-example that can be simulated, which helps the designer view the scenario in which the violation occurs. The relational abstraction tool improved scalability of current model checking tools over 10X.

Similar to deterministic verifications, the key consideration for probabilistic methods is the characterization of the ability of the designed system to meet the specified performance requirements. While deterministic methods seek to offer a yes/no answer to verification questions, probabilistic methods provide a probabilistic certificate of correctness (PCC) using methods of uncertainty quantification (UQ). The META project used probabilistic techniques developed by our teammates for two purposes.

3. **Oregon State University (OSU)** developed test benches for PCC calculation, both for lumped parameter dynamic models and finite element models. Five categories of uncertainty quantification methods were incorporated in the test benches. This allows the selection of a technique which matches the dominant modeling abstraction used in the different design phases. In addition, the tool suite includes methods for global sensitivity analysis by utilizing an algorithm developed by the **MIT** team.

4. **Palo Alto Research Center (PARC)** developed a simulation-based uncertainty quantification method for evaluating system performance requirements under degraded conditions. The Fault-Augmented Model Extension (FAME) approach models degradation of components under different operational scenarios using probability distributions of damage parameters and runs the simulation test benches against reliability requirements using Monte Carlo analysis.

Accomplishment 7. Real-time Software Implementation Tool Suite

In the CPS context, the cyber subsystems and components - including software and computing/communication platforms – are considered to be an implementation technology for dynamic behavior that is integrated with physical dynamics. An essential element of the META approach is that computationally-implemented behaviors are specified using hybrid, lumped parameter dynamics. Therefore, cyber components can be co-designed with physical components and can be integral part of the design-space exploration process. The OpenMETA includes Vanderbilt’s Embedded System Modeling (ESMOL) and software generator tool suite, which is integrated into the overall META design flow.

Accomplishment 8. Design Space Analyzer and Visualizer

During design space exploration, the execution of test benches results in a massive amount of data. The **Aerospace Systems Design Laboratory of Georgia Tech** developed a systems engineering visual analytics tool that converts predefined data sets into a collection of interactive



Figure 3: Probabilistic Certificate of Correctness Details

analytical visualizations for the purpose of enabling or enhancing a user’s capacity for cognitive reasoning based on perceptual principles. The two primary benefits are:

- An increased conceptual understanding of the data being visualized.
- And an increased transparency as to how one should react to the information embedded in the data, including the use of surrogate functions that show a consolidated system behaviors over a domain of design parameters.

Being web-based enables the tool to be easily embedded in, distributed or integrated with other web-based services. Figure 3 shows one visualization method, the result of the Probabilistic Certificate of Correctness calculations generated by OSU’s PCC tool.

3.2. Shift product value chain toward high-value design activities

An important expectation for the META program was the development and utilization of a new interface between product design and product manufacturing processes. The fundamental enabler for creating this new interface is that both META and iFAB (the digital manufacturing foundry component of the AVM program) are model-based design environments—one for the product domain (vehicles in the case of AVM) and the other for fabrication facilities—and they are both predicated on the existence of a rich set of component models, context (environment) models, and manufacturing process models⁹. DARPA’s vision of defining the interface between design and manufacturing is to enable the separation of “fables” design processes from foundries that are able to accept formal META design representations (Technical Data Packages) and automatically configure a digitally programmable manufacturing facility. Untangling the hidden dependences between design and manufacturing by establishing precise, well-defined interfaces was the key in the radical restructuring in the VLSI industry over two decades ago.

Finding 4. Need for Product and Manufacturing Process Co-Design

The OpenMETA toolchain generates digital blueprints for design candidates in successively increasing detail, as the design progresses from a conceptual design to a highly detailed design. These META design artifacts flow to the iFAB Foundry tool chain in a standardized Technical Data Package (TDP). The goal of the iFAB Foundry is to automatically configure a digitally programmable manufacturing facility for the selection of manufacturing equipment, the sequencing of product flow and planning of assembly process steps, and the automated generation of machine and human instruction sets needed. In the other direction, the iFAB tool chain provides feedback to META informing the designer about the manufacturability of the design, known as Manufacturability Feedback Analysis (MFA).

While investigating the specification of interactions between META and iFAB, it has become obvious that the active use of varying levels of modeling abstractions (commonly used in META) are a much less utilized approach in the design of manufacturing processes of physical components. This is not the consequence of overwhelming detail-richness of

⁹ DARPA-BAA-12-14

manufactured physical components (model-based methods are widely and successfully used in software synthesis – an even more detail-rich implementation technology), but rather the reflection of the dominant engineering approach. This “impedance mismatch” in abstraction levels created a significant gap that challenged both the META and iFAB teams.

Accomplishment 9. Design-Space Exploration with Manufacturability Test

Establishing a robust interface between META and iFAB was an important goal for narrowing the design space with manufacturability considerations early in the architecture exploration phase of the design flow. The META and iFAB teams defined a new interface for providing Manufacturability Feedback Analysis for META by the iFAB tool suite that included results of the following analysis:

1. **Cost/Schedule:** Total estimated cost of manufactured product and schedule required. Includes cost of procuring components and the end-to-end manufacturing of the design.
2. **Conceptual Manufacturing Analysis:** Give quick feedback to the designer about the validity of the design with respect to manufacturability (e.g. completeness of design)
3. **Detailed Manufacturing Analysis:** Highly detailed feedback as to the manufacturability of both individual META components (such as machinability), as well as manufacturability of the entire design, considering availability of manufacturing equipment, dependencies encountered in sequencing of product flow, etc.
4. **Reliability, Availability, Maintainability, & Durability (RAMD) Analysis**

Results of these analyses have been utilized in the META exploration process to guide the design toward manufacturable systems.

A unique integration challenge between the META and iFAB processes was created by the difference between the tool integration frameworks. The key iFAB analysis tools, Manufacturing Analysis Augmentation Tool (MAAT) and Hull Design for Manufacturing Assist Tool (HuDAT), were implemented as plug-ins to the Creo CAD authoring tool. These tools provide a set of functions to help the manufacturing process designer move from a concept level design to a fully detailed design in the CAD domain, using structural abstractions. On the other hand, the META tools use the architectural abstractions of the CyPhyML model integration language, where CAD is only one of the design domains. Our integration solution for the META design process with the iFAB analysis was MetaLink, which provided real-time interaction between the CyPhyML design and component editor and the Creo CAD tool. The manufacturing information for components as provided by HuDAT and MAAT is captured as an iFAB-specified data structure that is incorporated into the CyPhy Component definition.

3.3. Democratize design

The powerful third element of DARPA’s AVM vision was the inclusion of small companies and research teams in defense innovation by making the technology infrastructure for advanced CPS design affordable and available. This approach and the technical framework are closely



related to Eric von Hippel's observation of a trend toward democratization of innovation¹⁰. There are two economic drivers for this trend. The first is the ongoing dramatic expansion of CPS application areas in defense systems combined with the rapid increase of new platforms and related ecosystems (e.g. modular UAV platforms, mobile devices and Transformative Applications, cloud platforms and many others). The emerging new platforms – although mostly appearing in IT domains for now – show examples for new business models that provide the technology infrastructure for small users virtually freely and gain benefits via increased component sales or other indirect returns. The second economic driver is the current lack of affordable, integrated engineering tool chains supporting CPS development. In the engineering domains, the \$10K-80K/seat tools are quite common and their integration into end-to-end tool chains is either not solved or prohibitively expensive. Appearance of low-cost (even no-cost) open-source alternatives to a CPS design automation infrastructure would drastically expand competition, widen available design expertise, and improve even the expert user base for sophisticated COTS tools integrated with open source and proprietary, in-house tools.

The situation with CPS design tools today has some similarity to the conditions in information technology in the early nineties, when the GNU project and Microsoft's Visual Studio integration platform pushed down the cost of software productivity tools by roughly two orders of magnitude. This change made the wide scale participation of small software houses and individuals in commercial software design and production feasible, which was a large contributing factor to the IT revolution of the nineties. The impact is further strengthened with DARPA's decision on making the AVM tools available for the NSF CPS research community via the CPS Virtual Organization (CPS-VO.org) portal.

Finding 5. Need for Collaboration Platforms and Access to Open- Source Tools

The accepted model of open source software development is peer production by collaboration, with the end-products – source code, "blueprints", and documentation; available at no cost to the public. Key enablers for the tremendous success and economic impact of open-source software are the collaboration platform technologies emerging from the tradition of software forges, such as SourceForge, GoogleCode, and GitHub. These platforms combine project hosting, web-based collaboration, and centralized version control system repositories. The open source software communities were transformed by services that support collaboration within teams by coordinating the work of geographically dispersed developers, and between the teams and their user communities, by providing web-based tools including documentation wikis, issue tracking systems and discussion forums and shared code repositories. Open- and crowd-sourced tools have a long tradition due to their long development and maturation time, and significant public investment in their creation. Today, open source repositories include 666,998 projects, 674,380 source control repositories, 30,879, 289,910 lines of code produced by 3,627,589 contributors worldwide¹¹.

¹⁰ Eric von Hippel: Democratizing Innovation, The MIT Press, 2005.

¹¹ <https://www.openhub.net/>

The open source movement is now spreading across different fields. While open-source frameworks, platforms and tools were isolated and highly specialized in the past, appearance of new, domain-specific open platforms (such as ROS¹², Eclipse¹³, Open Source Ecology¹⁴) and the related viable business models extended from software industry to robotics and to physical equipment design and manufacturing are becoming more popular and form well established part of the innovation infrastructure. The “Maker” community, and its tools (such as Arduino/Sketch and MIT’s Scratch) further demonstrate the trend.

There are two essential infrastructure pieces needed for democratizing design for complex CPS. The first is the adoption of the collaboration platform concept that facilitates the formation of geographically dispersed designer teams, providing easy access to shared computation resources and design repositories. The second is availability of open source (or integratable COTS) tools that can be deployed in a software-as-a-service (SaaS) distribution model.

The first need is addressed by the VehicleForge project of DARPA’s AVM program executed by another research team at the Institute for Software Integrated Systems at Vanderbilt. While VehicleForge was not part of the META project, the Vanderbilt teams worked closely together and integrated OpenMETA with the VehicleForge collaboration platform – a key technical prerequisite for the FANG1 design competition.

The second need led us to explore the availability, span, and quality of open source tools essential for model-based CPS design. Many of the open source tools, such as NASA’s OpenMDAO¹⁵, Sandia National Lab’s DAKOTA¹⁶ or OpenModelica are directly relevant to the goals of META and of high quality. We have found that 70-80% of the OpenMETA tool suite functionalities can be covered using high quality open source tools.

Accomplishment 10. VehicleForge – OpenMETA Integration

The VehicleForge – OpenMETA integration concept centers on the needs of the vehicle designers, whose primary analysis and manufacturing workflows were supported by the OpenMETA tool chain and the analysis services of the iFAB Foundry. Designers used VehicleForge services for team formation and administration, team collaboration, searching for and access to components in the Component Exchange using a discovery interface, creating and maintaining design repositories, accessing and downloading OpenMETA and iFAB tools and to perform analysis workloads to a remote job execution service running in the VehicleForge Cloud¹⁷. Component developers and curators used the OpenMETA tools for creating AVM

¹² <http://www.ros.org/>

¹³ <https://eclipse.org/>

¹⁴ <http://opensourceecology.org/>

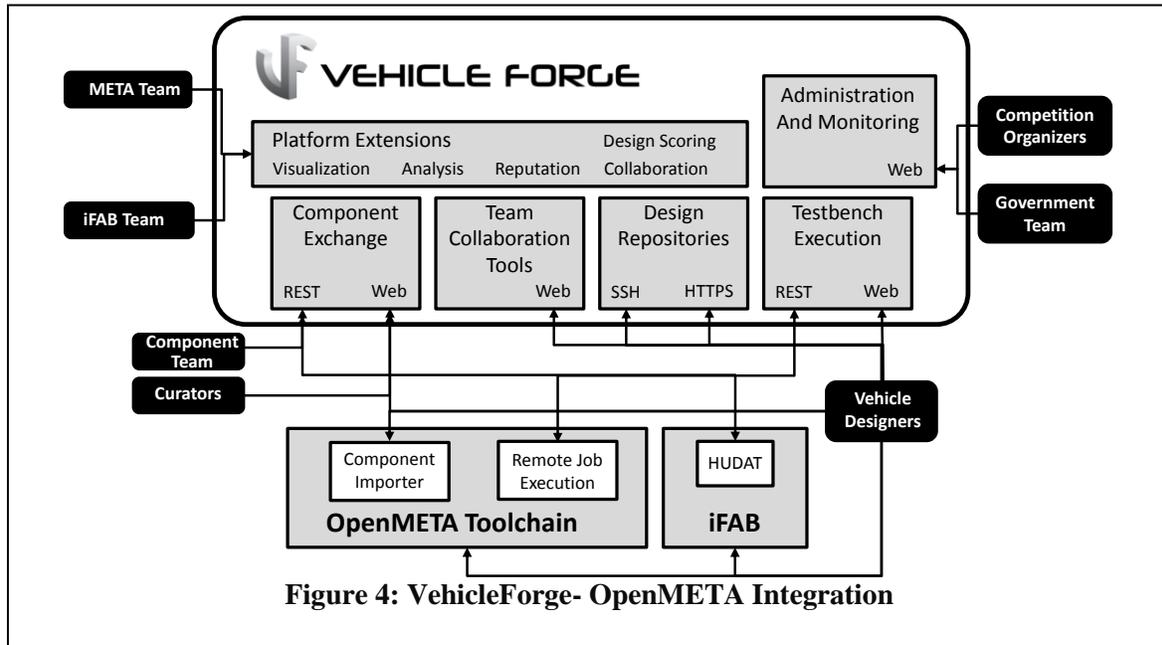
¹⁵ <http://openmdao.org/>

¹⁶ <https://dakota.sandia.gov/>

¹⁷ Larry Howard: “Final Report – vehicleforge.mil” Submitted to DARPA under Control No. HR0011-11-C-0098

components and added them to the VehicleForge Component Exchange. Competitions were facilitated by the administration and monitoring services.

While deeper integration of OpenMETA and VehicleForge based on web services is still in the future plans, we believe that OpenMETA and VehicleForge provided strong evidence that the integration of collaboration platform technologies with advanced tool suites for CPS design is a viable approach for opening up the field for a broad design community.



Accomplishment 11. Open-source Tool Configuration and Cloud-based Deployment

An important and far-reaching requirement in the development of the OpenMETA tool suite was the use of open-source tools and frameworks. The OpenMETA integration platforms enable the substitution of open-source tools with COTS alternatives (such as OpenModelica can be replaced with Dymola®, or OpenMDAO with Model Center®).

We estimate that in the current OpenMETA configuration the overall statistics are the following:

1. The OpenMETA integration platforms are implemented by ~1.5M lines of code developed under the META project. These integration platforms enable the utilization of (in the current tool configuration) ~29 open source and ~8 commercial tools. We estimate that this represents a 2 orders of magnitude larger source code base than the OpenMETA platforms.
2. We moved towards “democratizing design” in the following sense:
 - a. In many cases, the integrated tools provide functionalities that allow the use of open source and commercial alternatives to solve the same problem interchangeably.

- b. The level of expertise needed to access the tools required is greatly reduced, due to the automated composition of executable tool jobs via a highly simplified Test Bench model.
- c. Source code of the open source tools are available and can be changed/customized if needed.
- d. The source code base of the integrated open source tools are comparable (same order of magnitude or one order of magnitude larger) then the commercial tools.
- e. The cloud-based deployment strategy of OpenMETA enables low cost/low effort access to the OpenMETA design tools by prospective users.



Tel (615) 343-7472 Fax (615) 343-7440
1025 16th Avenue South Nashville, TN 37212
www.isis.vanderbilt.edu



4. Opportunities and Open Challenges

The META project of the AVM program addressed very hard problems, which are at the epicenter of component- and model-based design: (a) composing designs from reusable component model libraries, (b) extending the limits of correct-by-construction design, (c) raising the level of abstraction in design of CPS, (d) executing rapid requirements trade-offs, (e) restructuring the interface between design and manufacturing for CPS, and (f) creating an open framework for reusing open-source tool assets. Additionally, the large span of domains and tools was a challenge in itself. The project gave META developers unique opportunity not only to understand the limits of the current state-of-the-art in the context of a real-life DoD challenge problem, but also pushed the limits in several areas.

We believe that DARPA's AVM META program also provided opportunity for the developers and the research community in general to better understand the open problems and their impact on the broad applicability of model-based design technologies. Based on this experience, we summarize below the open challenges and opportunities that serve as a basis for defining short term opportunities and medium term challenges.

4.1. Short Term Opportunities

We believe that the following steps would significantly accelerate rapid transitioning of the AVM META results:

1. Creating seeds for validated, multi-fidelity model libraries targeted to different application domains.

While the AVM program initiated the construction of the FANG model library, the results are too limited for making a wider impact. A graduated approach based on designing core parts of domain-specific component model libraries and making those accessible through model exchanges (e.g. by leveraging AVM VehicleForge) could tremendously accelerate progress and impact.

2. Creating repositories of seed designs for accelerated design space construction.
It was our experience that seed designs (canonical system architectures in aerospace, ground vehicle, and other domains) are tremendously helpful for constructing design spaces and accumulating knowledge. With OpenMETA, this captured knowledge is immediately executable and able to be queried. These seed designs serve similar roles that design patterns perform in software engineering, a large impact productivity improvement method in information technology.
3. Benchmarks, test beds and repositories for CPS design tools.
A critical factor in the development of high quality tools for CPS is the availability of examples, benchmarks, and automated test benches that can be used to assess tool performance. Open availability of benchmarks and automated test benches that capture salient aspects of CPS design from an industrial perspective would be a significant step toward better utilizing the results of public investment.
4. Documented, carefully designed experiments with META design flows.
Compelling evidence on the effectiveness of model-based design flows in selected



application domains requires disciplined experiment design methods and extensive documentation. While the FANG1 Power Train challenge and the Hull Design challenge have started this process, significantly more data is needed to demonstrate and measure the radical impact of this new technology.

4.2. Medium Term Challenges

1. Product and manufacturing process co-design
Merging isolated product and manufacturing process design phases into an integrated co-design process promises the largest benefits and truly revolutionary advantages. This will be particularly important with the increased use of composites in manufacturing, in which the interdependence of product models and manufacturing process models is important and not well understood. The META design space exploration strategy and infrastructure can be extended to address this problem after bridging the “abstraction gap” between the current design and manufacturing sides.
2. Goal directed model composition
Automated system-level composition plays a fundamental role in design space exploration. Test benches use the composed models for running simulation or verification based checks if requirements are satisfied. Scalability is becoming a bottleneck if system-level models are always composed from the highest fidelity component models. An important opportunity for addressing the scalability challenge is to make the composition process adaptive to the property computed by the test bench. At a higher level, functional & physical design techniques can be merged using design space exploration to match goals from functional system decomposition to predicted performance of a library of trusted physical design spaces.
3. Extension of design objectives
The current META design flow is focusing primarily on performance, manufacturability, and some aspects of safety and reliability. The OpenMETA integration platforms are generic enough to enable the extension of design objectives to include security, resilience and other essential objectives.
4. Configurable design environments
The META horizontal integration platforms have emerged as “side products” of the OpenMETA tool chain development effort. The primary end users during the OpenMETA development were vehicle designers. Consequently, the implemented automations and user interfaces serve designers. However, the emergence of the model, tool and execution integration platforms – the core contributions of OpenMETA – creates opportunity for automation and improved user interfaces for another category of users, whose goal is to integrate domain specific integrated CPS design tool chains. While the solution is a significant undertaking it could have a long term impact on the future of component- and model-based design.



5. Transitioning

The META transitioning process has started in 2013 with founding Metamorph Inc., a Vanderbilt spinoff company. Metamorph is currently developing an open design automation tool suite based on the OpenMETA integration platforms directed to electronic design, which is expected to be an important element of Google's ARA smart phone platform.

Vanderbilt is conducting a range of early pilots and experiments with companies such as GM, Oshkosh (as part of the DMDII transitioning), GreenDynamics, Raytheon, and others. There are strong initial interests at government agencies as well, including NIST, DOT and NSF. Although the transitioning efforts just started, we believe from the early indications that there is a strong growing momentum both in the public and private sector.

6. Acknowledgment

DARPA's AVM program provided exceptional opportunity for testing the current state-of-the-art and pushing the limits of component- and model-based design in a real-life DoD challenge problem. The comprehensive vision of the program, the very hard challenges and the demanding application domain, highly motivated the participating teams of researchers from Vanderbilt, Georgia Tech, MIT, Oregon State, SRI, PARC, and Modelon.

On behalf of the research teams we would like to express our gratitude to DARPA and the Tactical Technology Office for making this project possible. We are grateful for the vision and leadership of Mr. Paul Eremenko who conceived and led the program through the crucial first two years. We also would like to express our appreciation to Dr. Nathan Wiedenman for steering the program through the critical phase of FANG design challenge, and Dr. Kevin Massey for guiding the program to its conclusion.

The program was tremendously helped by the leadership of Dr. Mike Lowry, Chief Engineer, and Dr. Kirstie Bellman, Chief Scientist in the early years, as well as the SETA team at the DARPA Program Office.

We would also like to recognize the importance of the guidance we received from the Senior Strategy Panel including Prof. Joseph Sifakis (VERIMAG), Alberto Sangiovanni Vincentelli (UC Berkeley), Daniel Frey (MIT), Olivier De Weck (MIT), Karl Hedrick (UC Berkeley) and Janos Sztipanovits (Vanderbilt) during the crucial early period of the program.