Adaptive Vehicle Make Overview

Note: The views expressed are those of the author(s) and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

DARPA's AVM portfolio of programs had a goal of compressing the development timelines for new complex cyber-physical systems by at least five-fold. With AVM, DARPA pursued the development of several elements of enabling infrastructure aimed at radically transforming the systems engineering / design / verification (META), manufacturing (iFAB), and collaborative innovation (VehicleFORGE) elements of the overall "make" process for modern defense systems. Each of these infrastructure capabilities was largely generic and applicable to any cyber-physical or cyber-electro-mechanical system. This approach was highly dependent on the generation of high-fidelity component, context and manufacturing process models. In order to exercise these capabilities in the context of a relevant military system, AVM also included FANG—the Fast, Adaptable, Next-Generation Ground Vehicle—a design and development effort for a new amphibious infantry fighting vehicle (IFV) that was implemented for the purpose of demonstrating the efficacy of AVM tools and processes. In early 2014, the AVM portfolio began a transition to the Digital Manufacturing and Design Innovation Institute.

Model-Based System Design, Analysis, and Verification Technologies

The CyPhy tool suite (developed under the META program) made numerous advances in model-based system design with a focus on compressing the development lifecycle for complex defense systems. At the core of the tool suite is a generic modeling environment (GME) that can represent, transform, and integrate the numerous types of models and analysis used in the design and development of complex defense systems. This modeling environment enables the CyPhy tool suite to accomplish compositional design, trade space analysis and virtual performance testing in an extendable framework that can integrate tools ranging from robust commercial and open-source numerical simulations to cutting-edge research prototypes. CyPhy has been under evaluation since January 2013 through multiple design exercises. During these design events, teams worked to design drivetrain, suspension, propulsion, chassis, and structural elements and associated subsystems for an amphibious IFV. DARPA has built a Mobility and Chassis Automotive Test Rig (ATR), designed with the tools) in the iFAB Foundry to complete the 'end-to-end' design/develop/build/test process flow. Insights gleaned from this effort are being used to inform further CyPhy development efforts to integrate new design methodologies, consider more difficult physics domains, and validate virtual design, modeling, and simulation tools.

The expected result from a complete tool set is a revolutionary leap in the design of complex cyber-physical systems, similar in manner to the way previous government and industrial investments revolutionized the design and manufacture of integrated circuits. AVM's goal of five-fold compression in development, design, and build time – compared to state-of-the-art practice – depended on the development of specific new capabilities. The goals of the CyPhy tool suite development effort were:

- Support *design flow through levels of abstraction* with early, incremental, and continuous analysis of designs and design spaces that enable system designers to efficiently navigate the design trade-space. The tools compute physics-based simulations in order to map variations in static design features to dynamic behavioral trade-offs as measured against a set of operational requirements.
- Provide *system and subsystem verification* at different levels of abstraction. This minimizes the need to rework a design due to flaws found late in the development cycle. An important advance is a formal stochastic Probabilistic Certificate of Correctness (PCC) that incorporates

uncertainty analysis – variations in manufacturing, uncertainty in modeling, and uncertainty in the operational environment. This probabilistic analysis ensures that vehicles designed using these tools robustly meet requirements when manufactured and deployed.

- Enable *semantic integration through the lifecycle*, including ontology-indexed component model libraries; compositional design tools; design verification tools; and the generation of detailed manufacturing directives spanning machine instructions, human work instructions, and logistics flow. This semantic integration ensures a seamless and coherent design flow.

Whereas current state-of-the-art Computer Aided Design (CAD)-centric design tools encourage assembly of physical subsystem prototypes that are guided by derived system-level requirements but developed in isolation, CyPhy implements a compositional design approach that enables designers to consider the behavior of the whole system design in multiple domains throughout the design process. This ability emerges from the compositional nature of AVM component model representations of physical parts. The CyPhy tool suite is able to reason about the geometry and behavior of an assembled system by aggregating the behavioral models of each component in all of the relevant domains (i.e., physical, thermal, vibrational, electro-magnetic, etc.).

Manufacturability Feedback and Manufacturing Configuration Technologies

The manufacture of complex cyber-physical systems is generally limited to two options: large scale manufacturing that is efficient at producing one known product quickly with repeatability, or prototype assembly of a single (or few) test article(s). Factories that do one generally cannot efficiently do the other – it is currently not possible to achieve efficient large-scale manufacturing that is also flexible. The iFAB (instant Foundry, Adaptive through Bits) program attempted to accomplish both, in one manufacturing entity. The manufacturing tools developed under AVM evaluate a system design representation and automatically configure a digitally programmable manufacturing facility tailored to fabricate it. The IT infrastructure required to accomplish this is also able to provide detailed manufacturability information in the form of cost, schedule and identification of component cost drivers to the designer throughout the design process. Thus, the final verified design sent for manufacturing already has manufacturing considerations factored in. The physical instantiation of the manufacturing capability, called a foundry, includes a network of participating manufacturing facilities and equipment, the sequencing of the product flow and production steps, as well as work instructions. In essence, the manufacturing tool chain seeks to eliminate the learning curve inherent in large-scale manufacturing even for limited build numbers and thus reduce costly design changes late in the development phase.

The manufacturing tool suites center around two functions: 1) providing manufacturability feedback to the designer at a level of detail commensurate to the design; and 2) configuring a foundry of networked manufacturing capabilities tailored to the final verified design, including supply chain optimization considerations, assembly planning, and automatically generated computer-numerically-controlled (CNC) and human work instructions. The automatic generation of work instructions from designer-supplied, detailed information represents a dramatic time savings in the development of a system. In addition, the digital record of human and machine instructions brings production level documentation to prototype quantity builds.

Much like the design tools, the manufacturing tool suites require detailed formal models representing the capabilities of various manufacturing machines and processes. By mapping these models into the same semantic domain as the vehicle design, the tools can automatically constrain the design trade space such that designs that are not manufacturable in a given network of manufacturing capabilities are automatically culled from the design solution space. Though we term the network a "foundry" – principally to differentiate it from a conventional factory that, at least in the defense world, tends to be a custom facility tailored to a specific product or small set of product variants – in actuality, it is primarily an information architecture. Only the final assembly capability needs to be co-located under a single roof in anything resembling a conventional fabrication facility; the rest of the manufacturing network can be geographically distributed and extend across corporate and industrial boundaries, united only by a common model architecture and certain rules of behavior and business practices. The capabilities and cost structure of each foundry partner organization are represented by a Virtual Production Environment (VPE) which consists of a series of algorithms specific to that organization. These VPEs represent a contractual agreement between the foundry partners for cost and schedule availability, vastly reducing the quoting time on individual components. To further validate the tools developed under AVM and identify gaps for development, DARPA has built an ATR, designed with the tools and exploiting the capabilities of the iFAB Foundry.

Component, Context, and Manufacturing Model Library (C2M2L) Technologies

The trade space exploration and design analysis tools as well as the manufacturability feedback and foundry configuration tools both require a comprehensive set of component, context, and manufacturing model libraries as a foundation. The development of these libraries enabled designers to swiftly consider multiple design iterations and evaluate them in a semi-automated manner via the design (META) and manufacturing (iFAB) tool chains. Design configurations that meet the program cost, timing, and performance objectives will emerge and progress so that incrementally higher levels of analysis can be used to select the design that optimally meets the stated requirements.

The component models developed to support the AVM tool chain consist of much more than geometric CAD data; they contain data describing cyber-physical behavior, material properties, interface constructs and procurement/manufacturing capability in addition to component geometry. The component models have the ability to be composed into complex assemblies such that aggregate behaviors are defined by the component models from which they are composed. In order to facilitate the creation and exploration of broad design spaces, a component model library must contain multiple instances of each component type (*e.g.*, 15 engine variants, 12 transmission variants, 6 drive shafts) with representations across all relevant physical and cyber domains. The designer can then select appropriate components from specific part-class libraries or even have the CyPhy design tools consider multiple or all options from a part-class library to automatically form a design trade space.

The AVM program worked to instantiate a vision where the performance of a system could be validated with comprehensive modeling and simulation to result in a credible, predictably correct system design. Using the CyPhy design tools, system designs composed of models from the component model library can be virtually tested through simulation using context models that simulate the real world conditions that the system would encounter in operation. To support the Mobility & Chassis design exercise, the context model library spanned multiple environments (atmospheric, land and aquatic) with nine context categories identified for modeling, representing 33 specific environmental conditions related to IFV requirements. The context model library was significantly expanded to enable blast, ballistic impact, human factors and transportability analysis to support a second design exercise focused on the structural integrity of an amphibious IFV. The revolutionary benefit to the AVM approach is that complex systems can be conceived and virtually tested in a matter of months as opposed to years using conventional approaches.

The AVM vision also called for nearly automated assessment and feedback of design manufacturability based on the modeled capabilities of the network of manufacturing facilities within the iFAB Foundry. This Manufacturing Model Library (MML) can be thought of as yet another set of contexts to which designs must be subjected. The MML not only informs manufacturability analysis of designs, but also forms the basis for the iFAB tools that generate human and machine work instructions to manufacture these designs.

In an effort to enable transition, the C2M2L effort not only focused on model library development, but also on establishing and promulgating standards to serve as the basis for future coordinated modeling efforts across a community of interest. The existence of formalized and accepted model standards reduces the risk and up-front investment faced by new entrants who are eager to take advantage of AVM-like capabilities. For the ground vehicle domain component model library, the program established both a part-class structure that logically organizes the vast list of ground vehicle elements down to the numbered part level as well as a generalized model standard for each of the part-classes that defines all the required elements. Thus, generation of a specific commercial variant of a component model (an internal combustion engine, for example) would only require alteration of the generalized engine model standard with performance specifications, geometry and variation data unique to the identified engine.

Web-Based Collaborative Design Platform

The VehicleFORGE platform is a web-based, collaborative design environment based loosely on the concept of modern day software 'forges' (*e.g*., SourceForge, forge.mil). The purpose of VehicleFORGE within the AVM program was to provide a common platform for design event participants to team, collaborate and submit designs for testing. As such, it is also the portal for accessing design tools, manufacturability tools, component models, and generated system designs. VehicleFORGE was developed with an extensible architecture that allows plugins to be developed for specific tasks. To date, plugins exist for CAD visualization, project design trade space exploration, and scoring analysis based on virtual testing. The VehicleFORGE platform, hosted on a private cloud at Vanderbilt University, is highly scalable, fault tolerant, and flexible.

Tool Suite Elements Overview

Vanderbilt University Institute for Software Integrated Systems (VU ISIS) POC:
Janos Sztipanovits, janos.sztipanovits@vanderbilt.edu

Penn State Applied Research Laboratory (PSU ARL) POC: Mark Traband, mtt1@psu.edu

Southwest Research Institute (SwRI) POC: Dr. James Walker, james.walker@swri.org

Ricardo, Detroit Technical Center (DTC) POC: Paul Luskin, Paul.Luskin@ricardo.com

Other POCs as noted in the table below.

Tool Readiness Key:
Green : developed, Gamma tested, ready for transition
Yellow: developed, Gamma tested, needs further development
Red: developed, not Gamma tested, needs further development

Tool Name	Developer	Readiness	Description
CyPhy/GME	VU ISIS	**Green**	The CyPhy editor is a desktop tool for graphically capturing components, designs, and design spaces; it also works with requirement specifications and system evaluation test-benches. This is the primary graphical user interface (GUI) through which models and design spaces are constructed, analyzed, assessed, and managed. This Windows desktop tool builds upon the generic modeling environment (GME) that has had over 15 years of development. It employs several interpreters integrated through the use of a common semantic backplane.
CyPhy2Modelica	VU ISIS	**Green**	Composition of Dynamic System Simulations for component models for dynamic behavior written in Modelica.
CyPhy2Cyber	VU ISIS	**Red**	Composition of Software-Based Controllers from state machines and signal flow description written in Simulink/Stateflow to a physical/software co-simulation.
CyPhy2CAD	VU ISIS	**Green**	3D Computer Aided Design Composition.
CyPhy2FEA	VU ISIS	**Yellow**	Composition of 3D analyses using Finite Element Analysis (FEA) techniques.
WebGME	VU ISIS	**Red**	An open-source web-based GME platform providing high-performance and simultaneous access to the model database and META design environment to multiple online users.
PDE Analysis	VU ISIS	**Red**	Integration of existing Partial Differential Equation analysis tools for thermal analysis, computational fluid dynamics, etc., of complex systems.
Design Space Exploration Tool (DESERT)	VU ISIS	**Green**	A powerful constraint evaluation engine to explore vast design spaces. The tool provides a listing of all of the possible combinations of components and rules can be created by the user to filter these results by design symmetry, complexity, logical assembly, and much more. This allows for the rapid visualization and narrowing of the design space based on the completeness of the component library and desired filter inputs. After static constraints are applied that narrow the design space, the remaining design combinations are simulated or analyzed against system test benches through the job manager, with the results displayed in multiple formats.
Parametric design Exploration Tool (PET)	VU ISIS	**Green**	Allows for the exploration of a design space with respect to adjustable component parameters for a fixed design architecture. Complements DESERT with parametric variation.

META Link	VU ISIS	**Yellow**	A powerful design tool that enables synchronous operation of a CAD application with the META modeling environment (CyPhy) in a manner that automatically and instantaneously translates edits in one to the other.
Job Manager	VU ISIS	**Green**	An infrastructure for managing local and remote execution of analyses.
Hull Design for manufacturing Assist Tool (HuDAT)	PSU ARL	**Green**	Enables the annotation of weld information for standard and ballistic welds and automatically creates solid model geometry for the production of weldments from a simple faceted solid shape where it is exported into the META-articulated design representation for the purpose of inserting detail into a system design to prepare it for manufacturing.
TDP Editor/MAAT	PSU ARL	**Green**	Assists a designer in rapidly creating detailed design annotations for tolerances, surface preparation and assembly information in a native CAD environment to a level sufficient for a Technical Data Package for manufacturing.
Liaison Graph Generation	PSU ARL	**Green**	Automated generation of liaison graphs depicting the connection of all constituent components of a fully-articulated system design to allow reasoning about parts and groups of parts used in the assessment of optimized assembly sequences.
Subassembly Creation	PSU ARL	**Yellow**	Automated division of fully-articulated system designs into manufacturable subsections.
Manufacturing Assessment Tools			
Assembly	PSU ARL	**Green**	Determines if a complex, fully-articulated system design can be manufactured within a given foundry and evaluates assembly types for mechanical joins, welding and wire harnesses.
Casting	PSU ARL	**Red**	Determines if a component can be cast and automatically generates necessary risers and gates for casting process.
Machinability	PSU ARL	**Green**	Assesses if a component can be machined (drilled, turned, milled, etc.).
Tolerancing	PSU ARL	**Yellow**	Automatically assigns basic feature-based tolerances to components that lack them.
Foundry Configuration	PSU ARL	**Green**	Generates all possible foundry configurations and schedules using Manufacturing Assessment Tools and the Manufacturing Model Library; displays Pareto front of desirable configurations utilizing user-specified metrics to allow for comparison and selection.
Robustness Analysis	PSU ARL	**Green**	Monte Carlo simulation of foundry configurations to determine sensitivity of metrics to small perturbations in machine or component availability.

Machine Instruction Generation	PSU ARL	**Yellow**	Automatically generates G-code and setup sequences for parts (currently only for 3-axis orthogonal features) to manufacture parts on the machines specified by the foundry configuration.
Human Instruction Generation	PSU ARL	**Green**	Creates detailed work instructions for each portion of and assembling the system including lists of required tools and equipment for each step drawn from available facility equipment.
iFAB Product Lifecycle Management (PLM)	PSU ARL	**Yellow**	Purpose-built ,open source PLM tool to track build status and identify impacts on build schedule.
Blast and Ballistics Analysis Tools	Southwest Research Institute (SwRI)	**Green**	A suite of tools, scripts, and simulation code to assess design performance against various user defined/selected blast and ballistic threats. The ballistic analysis tool chain contains three tiers of analysis depending on the fidelity desired from quick scaling algorithms and physics-based codes that run in minutes, to complete Hydrocode simulations that reason about 3D part geometry over the course of days. The blast analysis tool chain contains five tiers of analysis with tiers 3-5 leveraging LS-DYNA commercial software. The blast analysis tools contains pre- and post-processor scripts including an auto-meshing capability that greatly reduce the analysts time setting up and interpreting results from the analysis tools. Fidelity of analysis varies between tiers from rigid-body motion effects on the design in Tier 1 through complete anthropomorphic test dummy injury number analysis in Tier 5.
Component Model Library	Ricardo, Detroit Technical Center (DTC)	**Yellow**	This library of more than 1,000 components, multi-domain sub-components, and domain-specific extensions of the Modelica Standard Library has been populated to represent several component classes across the powertrain & mobility; material stock, structures and parametric hulls; and miscellaneous hardware system contents for an amphibious infantry fighting vehicle. It should be noted that many of the components developed contain ITAR-protected data and will not be publicly distributed.

Probabilistic Certificate of Correctness (PCC)	Oregon State University Christopher Hoyle chris.hoyle@ oregonstate.edu	Yellow	The use of probabilistic verification techniques to calculate the uncertainty bounds resulting from model uncertainty of the system's components as they impact performance predictions. It can be used for uncertainties ranging from manufacturing tolerances and differences to variations in the operational environment. It employs six different algorithms, with Monte Carlo being the most robust but also least efficient. The result is uncertainty quantification expressed as a probability density function for the various dimensions of performance. This enables a system designer to choose designs that are robust and have a high likelihood of successful performance given real-world variations.
Qualitative Reasoning (QR)	Xerox PARC Johann DeKleer dekleer@ parc.com	Red	QRM performs automated qualitative reasoning about the physics of a system at a high level of abstraction. In essence, it performs a set of parallel qualitative simulations of a system that results in a state graph called an envisionment. As envisioning is done at a high level of abstraction, it simulates a large portion of the design space, encompassing wide ranges of concrete design parameter choices in the same abstraction. After the envisionment process, QRM can partition the parameter ranges into those that definitely meet design requirements, those that definitely fail design requirements, and those that might -in other words, be uncertain. The mapping from the abstracted qualitative system to concrete system instantiations is done through automated algebraic analysis of the differential algebraic equations governing the system (specified in either Modelica or QRM's Qualitative Modeling Language). QRM can also provide the designer feedback in the form of parameter inequalities differentiating system instantiations that have only good states versus those that have bad states. Instantiations that only have good states are verifiably correct. In mathematical terms, the process is a formal method that does explicit-state model checking at an abstract level through the qualitative calculus, followed by a refinement step that maps this to the concrete level.

Fault Augmented Model Extensions (FAME)	Xerox PARC Bhaskar Saha Bhaskar.Saha@ parc.com	**Red**	Tools to support reasoning about and understanding system reliability. Over long periods of operational use, components become degraded with wear and fatigue. First performance is affected, and eventually the system fails unless maintenance is done. FAME automatically generates component models of degraded dynamic behavior from the Modelica descriptions of normal component model dynamic behavior. It then uses first-principle physics analysis to plot the parameters of degradation over time based on profiles of operational use.
HybridSal and Relational Abstraction (RA)	SRI Ashish Tiwari tiwari@ csl.sri.com	**Red**	For a wide range of properties that can be specified by a designer, HybridSal will either verify that the design of the embedded software control and physical system is correct, or generate a counter-example of a scenario where the property is violated. In its current instantiation integrated into CyPhy, software controllers are represented as Simulink/Stateflow designs (StateCharts with PID controllers), and physical systems are represented in Modelica as sets of differential equations. In mathematical terms, this tool is a formal method model checker for cyber-physical systems, in which embedded software controls a physical system operating in a range of environments. Relational Abstraction makes this one of the most scalable hybrid system model checkers capable of doing design verification.