A Common Basis for Mixed-Technology Micro-System Modeling

James P. Hanna Robert G. Hillman
Air Force Research Laboratory
Multi-Technology Integration Branch
525 Brooks Road
Rome, NY 13441-4505
hannaj@rl.af.mil hillmanr@rl.af.mil
Ph: (315) 330-3473 Fax: (315) 330-2885

ABSTRACT

An ad hoc working group, organized under the DARPA Composite CAD program, is developing a common set of VHDL-AMS (VHSIC Hardware Description Language for Analog and Mixed Signal) definitions for modeling mixed-technology systems. This common set of definitions will enable VHDL-AMS model interoperability, composition, and exchange. VHDL-AMS promises to provide similar benefits and infrastructure for mixed-signal (electronics) and mixed-technology (mechanical, fluidic, thermal, etc.) system design that VHDL did for the design of digital electronics. The working group is also developing a set of modeling guidelines and VHDL-AMS examples. This paper presents and discusses this joint effort between Industry, Academia, and the Government. The results of this effort will form the basis for standardization within the IEEE Design Automation Standards Committee. A common basis for model interoperability is essential to the successful application of VHDL-AMS to mixed-technology system modeling and design.

Keywords: VHDL-AMS, MEMS, Modeling, Simulation.

INTRODUCTION

Today mechanical microstructures, microsensors, and microactuators are being developed utilizing VLSI processing technology. This technology, referred to as MicroElectroMechanical Systems (MEMS), comprises both mechanical structures and electronics integrated in the same physical structure such as a silicon chip. This microfabrication technology enables the fabrication of integrated devices, which can perform extremely complicated functions. This miniaturization and integration promises unique new system concepts related to size, power and weight.

The Composite CAD program is performing research to develop multi-disciplinary design tools that will effectively facilitate the design, analysis, and verification of these mixed technology systems. The systems envisioned consists of combinations of elements (digital, analog, MicroElectroMechanical, RF, Optical, etc.) fabricated on a single micro-fabricated chip or minimally assembled substrate. For use within this paper, a sub-section of the overall system will be referred to as a subsystem and can contain a single domain structure/element or a collection of multiple domain structures/elements. This paper summarizes the work being pursued to establish a synergistic system design methodology by the development team.

MIXED-TECHNOLOGY MODELING

Design methodologies will vary from design group to design group, institution to institution, and application to application. Various design approaches are needed to facilitate the design of mixed technology systems. Design methodologies may include distributed design teams using distributed design tools and databases. Combinations of domain specific CAD tools are needed as well as an integrating design methodology to support system design flow and analyses. Distributed collaborative design and variable design methodologies for integrated mixed technology systems will require the use of modern software engineering practices, such as object-oriented design and software development practices and open architecture approaches to achieving "plug and play" compatibility across domain design tools.

An effective design methodology exploits design and verification tools in the process of evolving design from the initial concept to the final detailed implementation in a systematic and analytic manner. A top-down design methodology exploiting predictive simulation is a powerful combination that allows the designer to explore problems early in the development process and span the conceptual distance between block level diagram and the physical details of implementation.

The development of a design hierarchy at appropriate subsystems allows the designer to mix multiple levels of abstraction to observe and evaluate interactions between interdependent subsystems. A simulated subsystem may be represented by very detailed 3D models, other portions represented by reduced order models, while other portions of the system are represented by very high-level models such as finite state machines (see figure 1). Systems implemented hierarchically using multiple mixed levels of abstraction can be effective in performing top-down design with bottom up verification. Any design process will employ various combinations of top-down and bottom-up design techniques.

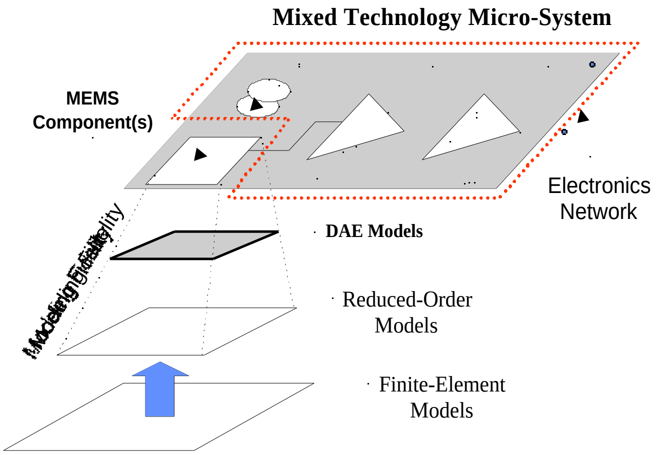

Figure 1: Model Hierarchy

Reduced Order Modeling

A reduced order model is an abstract notation for a simplified set of equations used to represent the terminal characteristics of a subsystem. Reduced order modeling is used to represent static and dynamic behavior to an acceptable level of fidelity, emphasizing terminal characteristics. This abstracted behavior may contain phenomena representative of multiple mixed energy domains. Accuracy is measured with respect to the original subsystem behavior. A reduced order model will typically require fewer components or fewer equations than the detailed subsystem model. Since the behavior is largely encapsulated at the terminal characteristics and will not necessarily represent the internal phenomena at the same level of detail as provided by 3D spatial or integral analysis, faster computation times and smaller storage requirements enable the effective simulation of larger system architectures. In addition to understanding the simplifying rules used in developing the reduced order models, the designer also needs to understand the valid ranges (and processes) for applying the model.

Reduced Order Model Generators

The Composite CAD development team has several groups performing research and developing tools to apply automation in the development of reduced order macro models. These models are derived from "X degrees of freedom" representations (3-D models and simulations) such as Boundary Element or Finite Element Methods. Most of the work in the Composite CAD program has focused on "small signal" and linear models. However, as we expand our design horizons and bring in new systems, many devices will exhibit non-linear behavior. Work is expanding to address these more difficult technical problems with reduced order modeling techniques to help develop models for highly non-linear systems with distinct discontinuities.

It is essential that the reduced order models developed be targeted for insertion into reusable libraries. Therefore the reduced order models must be appropriately parameterized and their design intent documented. Reduced order models can be a single point design or a family of parameterizable design elements. This may not hold true for reduced order models derived automatically but should definitely be considered and attempted for all reduced order models constructed. One important issue is the need for a common simulation paradigm and model framework.

Fundamental Interoperability Challenges

The Composite CAD reduced order modeling teams need to address conventions in modeling and terminology. All the reduced order models developed must have compatible interface ports so that the various architectures can be interchanged and used within a larger model. The working group will utilize VHDL-AMS to develop an initial set of common VHDL-AMS packages for MEMS technology. They are also developing a reduced order model coding style that attempts to capture and define input and output semantic requirements, capture the transfer functions or behavior utilized, annotate the valid operating ranges and constraints.

VHDL-AMS

The development and standardization of VHDL was an enabling technology that allowed the introduction of top-down design methodologies for digital electronics. VHDL allows designers to model the discrete event behavior of systems across multiple levels of abstraction. Hierarchical VHDL model-based design techniques were developed to combat design issues such as: increased complexity, time to market pressures, and life-cycle-costs of digital systems. This methodology provided the means to transfer unambiguously design information and communication about modeling methods between developers.

Many VHDL-based CAD tools (for graphical design entry, simulation, synthesis, test bench generation, etc.) are now commercially available. This infrastructure has greatly reduced the development cycle and improved the quality of digital electronics design. Recently, the VHDL language has been extended to support the integration of continuous time and discrete event simulation. This new language is known as VHDL-AMS (Analog and Mixed-Signal). VHDL-AMS will provide the same benefits, and infrastructure, for mixed-signal (electronics) and mixed-technology (mechanical, fluidic, thermal, etc.) systems design.

VHDL-AMS allows designers to model any system that can be represented by a set of Ordinary Differential Algebraic Equations (ODAEs) at any level of abstraction and supports both conservative (generalized KVL, KCL) and non-conservative (signal flow) systems. This implies that any physical system that exhibits strictly analog behavior or a mixture of analog and discrete behavior can be modeled and simulated using VHDL-AMS. VHDL-AMS is currently in the final stages of standardization within the IEEE Design Automation Standards Committee and will soon be IEEE Std, 1076.1.

Although the VHDL-AMS standard provides extremely powerful support for mixed-signal and mixed-technology system design, it does not include a set of common definitions for any particular modeling application (electrical, mechanical, fluidic, etc.). Common definitions are needed if libraries of elemental component models (essential to enable synthesis) are to be developed and exchanged. A common set of definitions will provide the basis for VHDL-AMS model interoperability, composition, and exchange.

Equally important are guidelines that define common methods for addressing various issues surrounding VHDL-AMS model development. Specific modeling approaches may have significant impact on accuracy, efficiency, and convergence. The intent in providing modeling guidelines is to promote a common set of approaches which address differing requirements, support automation, and discourage the proliferation of VHDL-AMS dialects.

Common Definitions

VHDL-AMS is a strongly typed language. Strong typing simplifies model development by providing early error checking during model development. Strong typing, however, requires models to be connected together only by ports that have common types (natures). Model reuse and the availability of MEMS Intellectual Property (IP) rely on the existence of common definitions and naming conventions.

The VHDL-AMS standard supports the aggregation of common definitions within a language construct called a package. Models may reference these common packages to make use of these common definitions. Common definitions may consist of type and subtype definitions, nature and subnature definitions, constant and alias definitions, and subprogram definitions.

The VHDL-AMS *nature* construct defines a template for *terminal* definitions. Natures define the *across* and *through* aspects of conserved energy connections. Terminals provide connection points where the conservation of energy laws are enforced. These laws state that the sum of all across quantities around a closed path must equal zero, and that the sum of all through quantities at a node must equal zero when the system is in equilibrium. Common nature definitions assure that VHDL-AMS models can be shared and composed into larger, more complex system applications.

Common packages have been developed to support the modeling of any combination of the following: electrical systems, mechanical systems, fluidic systems, thermal systems, and radiant systems. A package common to all modeling domains has also been developed. The energy systems package contains types, constants, and scaling factors that are commonly used across energy domains. Figure 2 illustrates nature declarations for the electrical and mechanical domains, as well as a VHDL-AMS example of a piezoresistive pressure sensor.

```
-- from package ELECTRICAL_SYSTEMS
--
subtype VOLTAGE is REAL tolerance
          "DEFAULT_VOLTAGE";
subtype CURRENT is REAL tolerance
          "DEFAULT_CURRENT";
nature ELECTRICAL is VOLTAGE across CURRENT
        through ELECTRICAL_REF reference;

-- from package MECHANICAL_SYSTEMS
--
subtype DISPLACEMENT is REAL tolerance
         "DEFAULT_TRANSLATION";
subtype FORCE is REAL tolerance "DEFAULT_FORCE";
nature TRANSLATIONAL is DISPLACEMENT across FORCE
         through TRANSLATIONAL_REF reference;

library IEEE;
use IEEE.ELECTRICAL_SYSTEMS.all;
use IEEE.MECHANICAL_SYSTEMS.all;
entity MEMS_PIEZORES_TRANSDUCER is
  generic (constant R_NOM        : RESISTANCE;
        constant YOUNG_MOD     : REAL;
        constant POISSON_RATIO : REAL;
        constant APPLIED_AREA  : AREA);
  port    (terminal NODE1, NODE2 : ELECTRICAL;
           quantity F : in FORCE);
end entity MEMS_PIEZORES_TRANSDUCER;

architecture BEHAVIORAL of
MEMS_PIEZORES_TRANSDUCER is
  quantity V across I through NODE1 to NODE2;
  quantity R, R_DELTA, E_STRAIN, K, STRESS : REAL;

begin
   -- define branch constitutive equations
   -- define conservative energy network relations
   V == R * I;

   -- define adjoint energy network relations
   R == R_NOM * (1.0 + R_DELTA);
   R_DELTA == K * E_STRAIN;
   E_STRAIN == STRESS/YOUNG_MOD;
   K == 1.0 + 2.0*POISSON_RATIO;
   STRESS == F/APPLIED_AREA;
end architecture BEHAVIORAL;
```

Figure 2: Piezoresistive Pressure sensor.

The results of this work will be presented to the IEEE Design Automation Standards Committee (DASC) later this year. This is the path that must be followed to establish an

IEEE study group under the DASC to establish an IEEE Project Authorization Request (PAR) and bring these packages through the standardization process.

Guidelines & Examples

Guidelines are intended to define standard methods for dealing with various issues surrounding model development. Different modeling approaches may have very different requirements and implications as to model accuracy, efficiency and convergence. The intent in providing modeling guidelines is to promote a common set of approaches which address these differing requirements, while at the same supporting automation and discouraging the use and support of VHDL-AMS dialects. Issues that are being addressed by modeling guidelines include:

- Approaches to modeling discontinuities
- Modeling piecewise behavior
- Creation of component libraries (what additional information need be present to allow consistency checks by automated tools)
- What are the basic attributes that need to be defined in all reduced order models (e.g., required inputs, available outputs, transfer functions / behavior, valid operating ranges, constraints on operation,)?
- How accurately must the reduced order models reflect the expected physical behavior of the component? Will information on assumptions and reduced order be contained in reduced order model documentation?
- In what different forms can those attributes be expressed (e.g., how can behavior be represented: ODEs, PDEs, truth tables, systems of state equations, graphs, or other modeling methods - algorithmic, graphical, textual)? What technique for model representation is each research team pursuing?
- Can we develop base models for general classes of components? There are several different ways to pump fluids in a MicroFlume system - EO, EP, diaphragms, bubble pumps, etc. Different approaches to building MEMS-enabled gyroscopes are also appearing. Is there some common way of representing the required information for the generic component that would allow designers to investigate the different architecture types in the system design?

A set of examples is being developed to illustrate the use of VHDL-AMS for modeling mixed-technology microsystems. This model set consists of two classes of models: 1) models that illuminate the use of various VHDL-AMS language constructs for MEMS modeling and 2) models that demonstrate the application of the modeling guidelines to real-world MEMS modeling problems.

The modeling guidelines and examples are currently being developed by the working group, and they are expected to be distributed to the general Composite CAD community in the second quarter of 1999.

CONCLUSIONS

VHDL-AMS provides a new paradigm for modeling and simulation of complex mixed-technology microsystems. Interoperability standards and guidelines are essential to the successful application of VHDL-AMS to mixed-technology microsystem design. The application of the VHDL-AMS standard to MEMS system design will enable the emergence of a robust MEMS CAD tool base that will drive down the time-to-market and overall development costs for integrated microelectromechanical systems. Integrated, robust design environments and methodologies will bring the promise of greater infusion of mixed technology systems into a broad array of commercial and military applications.

REFERENCES

[1] R. Hillman, H. Dussalt, A. Dewey, “Composite CAD Reduced Order Modeling”, Workshop on Canonical and Hierarchical Problems for Mixed Technology Computer Aided Design Tools, April 23, 1998.

[2] A. Dewey, et al., “VHDL-AMS Modeling Considerations and Styles for Composite Systems”, Tech. Rep. Version 1.0, Defense Advanced Research Projects Agency (DARPA) 1998.

[3] A. Dewey, et al., “Intellectual Property (IP) Component Definition and Modeling for Microelectromechanical Systems (MEMS), Submitted to IEEE Design and Test of Computers, Spring 1999.

[4] B. Romanowicz, et al., “VHDL-1076.1 Modeling Examples for Microsystem Simulation”, 2nd Workshop on Libraries, Component Modeling and Quality Assurance, accompanying Computer Hardware Description Languages Symposium'97, Toledo, Spain, April 1997.

[5] B. Romanowicz, “Methodology for the Modeling and Simulation of Microsystems”, PhD, July 1997, Institute of Microsystems, Swiss Federal Institute of Technology of Lausanne.

[6] Lynn D. Gabbay and Stephen D. Senturia, "Automatic Generation of Dynamic Macro-Models Using Quasistatic Simulations in Combination with Modal Analysis", 1998 Solid-State Sensor and Actuator Workshop, Hilton Head, SC, pp 197-220.

[7] Lynn D. Gabbay, "Computer Aided Macromodeling for MEMS", Ph.D., June 1998, MIT.