OMG's First Annual
Model-Integrated Computing Workshop
Exploring the Synergy Between MIC and MDA®

Arlington, VA USA - October 12-15, 2004

Workshop Program

TUESDAY – October 12, 2004

TUTORIAL TRACKS

0900 - 1200 ***The Generic Modeling Environment (GME)***
Track 1 James Davis, Research Scientist, Vanderbilt University

The Generic Modeling Environment (GME) is an architecture for developing Model Integrated Computing systems. GME is based on over fifteen years of research in Model Integrated Computing (MIC) at the Institute for Software Integrated Systems at Vanderbilt University. MIC focuses on technology for the development of domain specific languages and domain specific environments. GME is our architecture used to realize domain specific languages and domain specific modeling environments. GME can be used to develop domain specific design environments (DSDE) that are then used to capture system specifications in the user's natural language and to automatically generate or configure target applications in a given domain (i.e. in a particular engineering field). In this tutorial, presentation slides will be used along with the interactive development of a sample DSDE. An overview of GME and its application areas will be given. GME has a modular, component-based architecture that will be discussed. Details on customizing and enhancing GME will be presented. An example domain specific environment will be constructed as part of the tutorial. While the actual model interpreter will not be written during the tutorial, details of an interpreter for the selected environment will be presented. At the conclusion, attendees will be familiar with the GME toolset, how to configure and customize GME, and for what types of systems GME is applicable.

0900 - 1200 ***The SAE AADL Standard: An Architecture Analysis & Design***
Track 2 ***Language for Embedded Real-time Systems***
Peter Feiler, Sr. MTS, Software Engineering Institute & Edward Colbert, Pres., Absolute Software

This tutorial will introduce participants to the Society of Automotive Engineers (SAE) standard Architecture Analysis & Design Language (AADL). The SAE AADL was developed for embedded systems that have challenging resource constraints, strict real-time response requirements that must tolerate faults, specialized input/output hardware, and certification requirements to high levels of assurance. It is a modeling notation used to describe the task and communication architecture of embedded systems as an assembly of application components, and its mapping onto an execution platform. This tutorial will introduce participants to the SAE AADL standard and its use in model-based embedded systems engineering. The tutorial will cover the following topics:

- Objectives, status and participants of the SAE AADL standard
- Model-based embedded system engineering with SAE AADL
- A summary of the language features
- A summary of two case studies
- Illustrated examples from the Avionics domain
- The UML profile of AADL
- AADL XML/XMI & Tool support

1030 - 1045 Morning Refreshments

1200 - 1245 Lunch

1255 – 1300 ***Opening Remarks* – Program Committee Chair**
Fred Waskiewicz, Director of Standards, Object Management Group

1300 – 1500 **Session 1: *Foundations***
Chair: Gabor Karsai, Associate Professor of EECS, Vanderbilt University/ISIS

MIC advocates the use of domain-specific models, expressed in various domain-specific languages. Using many, non-standard languages on a project can be disastrous, so it is a great concern for developers whether MIC is worth the effort. The theme of this session is to present and discuss the fundamental ideas in the approach, how MIC can be placed on a technically sound, formal basis, what happens if multiple modeling languages are used, and what are the resulting costs and benefits.

1300 – 1330 **Domain-Specificity as a Key to Success in Model-Based Design**
Michael S. Moore, Mark Brooks, Greg Willden, Southwest Research Institute
Sandeep Neema, Institute for Software Integrated Systems

This presentation will argue that it is the domain specificity of the Model Integrated Computing approach that is key to the success of model-based design; not only in the satisfaction of technical goals of the application space, but in gaining the acceptance of the domain users that is crucial to the eventual transition of the resulting tools into common practice. To support this claim, the presentation will refer to work done on the Software Defined Radio Open Experimental Platform (OEP) project, which was part of the Model Based Integration of Embedded Systems (MoBIES) program.

1330 – 1400 **Towards Formalizing Domain-specific Modeling Languages**
Kai Chen, Janos Sztipanovits, and Sandeep Neema, ISIS/Vanderbilt University

In this presentation, we are proposing a framework with a set of supporting tools to formalize the DSML design process. In this framework, both the syntax and semantics of the DSML are defined precisely and explicitly. Models of computation (MoC) are DSMLs for a finite set of standard, well-known semantic domains. These MoCs can be used as reusable semantic units, with which higher-order DSMLs are composed. A particular MoC is used as a case study to demonstrate the design process.

1400 – 1430 **Using Multiple Domain Specific Languages**
Steve Hickman, Senior Scientist, Honeywell

The value of domain specific languages modeling is about getting productivity and flexibility from abstraction. Instead of trying to pick a single language with a large enough domain, pick a set of languages whose domains form a covering set of the domains of relevance to the problem at hand, ensuring that the languages in this set have mutual boundaries where switching from one language to another is relatively painless. This presentation will define what is meant by domain specific and language, describe the requirements all languages must meet, illustrate the domains that various languages fall into, show how and where domain boundaries intersect, display examples of how language choices can impact productivity, explain why this approach is more productive than attempting to use a single language, and detail what our modeling tools must do to accommodate this approach.

1430 – 1500 Session 1 Q&A - Discussion

1500 – 1515 Afternoon Refreshments

1515 – 1755 **Session 2: *Modeling Methodologies***

Chair: Dwayne Hardy, Architecture & Systems Engineering Advisor,
US DoD/Open Systems Joint Task Force

Presentations in this session will review several approaches and key considerations when using a model-based approach throughout the system development lifecycle. The strategies and techniques presented will address: model management, transformation, execution, allocating functions to components and dynamic component deployment; model checking and instrumentation to aid in the specification and evaluation of non-functional systems features and characteristics that are often critical to the success of mission assigned to distributed, real-time and embedded (DRE) systems; implementation using complementary component-based and reuse strategies; and model synchronization to mitigate the challenges associated with the collaborative development in an asynchronous and distributed manner. Also, the role and limitations of existing and emerging model repositories, translators, checkers and related MIC tools that are needed to assist in implementing these approaches will be discussed.

1515 – 1555 **A Component-Minded Route from CIM to Real Implementations, And What Tools Can Do for It**

Manfred R. Koethe, CTO, 88solutions Corporation

If you query for the anticipated benefits of Model Driven Architecture, the top favorite among replies is "Code Generation". However, the high goal of MDA is preservation and reuse of information and efforts during the whole product lifecycle, achieved by applying a well-structured set of abstractions to the problem space. The commonly anticipated brute force generation of arbitrary code out of models stands in contrast to this school of thinking. Component-based systems have demonstrated a higher potential for reuse and sharing on implementation and execution level than "traditional" development approaches. This presentation will show a consistent path from highly abstracted Computational Independent Models to concrete and platform specific component implementations (artifacts) and their deployment into real systems. It will also be shown how existing MDA technologies and future tools can support this process.

1555 – 1635 **Managing Model Evolution in Model Driven Development**

Dan Matheson, Robert France, Roger Alexander, James DeWitt & Nathan McEachen,
Colorado State University

One of the great difficulties of Model Driven Development (MDD) is tracking all the models, the changes to the models and the model relationships. The changes occur as the solution space is explored and as the solution is refined. Utilizing development experience from other discrete manufacturing disciplines, I present an approach of using a repository as a critical means for the support and management of model evolution in MDD. There is a design for this solution and implementation efforts on a prototype are underway.

1635 – 1715 **Testing MDA Platform Independent Models**

Gregory T. Eakman, VP of Research, Pathfinder Solutions

Model Driven Architecture provides the benefits of technology independence, architectural integrity enforced by transformations, improved communications, and maintainability to the development of embedded systems, and adds to the already difficult problem of testing these systems. Just like compiler technology, there is no verification suite that can guarantee that the PIM to PSM transformation rules generate correct code in all cases. This session outlines a development and integration test approach to testing components and systems developed as executable platform independent models. The approach is based on research and practical experience, and built on model transformation processes and formalized design-for-test instrumentation code injected during transformation.

1715 – 1755 **Integrating Model Checking and MIC with a QoS-aware Component Middleware Platform**
Gabriele Trombetti, Aniruddha Gokhale, Douglas C. Schmidt, Vanderbilt University
John Hatcliff, Jesse Greenwald, Gurdip Singh, Kansas State University

Developers of mission-critical DRE systems face a number of vexing challenges that can be alleviated by a combination of model-integrated computing (MIC), QoS-enabled component middleware, and model checking. Although these technologies have traditionally evolved separately, we have been integrating them together in an MIC environment consisting of CoSMIC, which addresses the stages of development of QoS-aware component applications and Cadena/Bogor, which supports advanced analysis and model checking capabilities. This presentation discusses the challenges and solutions we faced when integrating these tools into an open MIC environment.

WEDNESDAY - October 13, 2004

0900 – 0945 **Keynote:** ***The 'Software Wind Tunnel': How to Win the Big Government Contracts for Embedded Software Development***

John Bay, MoBIES Program Manager, DARPA

0945 – 1200 **Session 3:** ***Standards***

Chair: Manfred R. Koethe, CTO, 88solutions Corporation

The trailer in an e-mail received said: "Standards are like toothbrushes, everyone agrees you should have one, but nobody wants to use yours." Unfortunately this is a very widespread mindset. Agreement and conformance to a common set of standards is however the essence of seamless interoperability, and this applies evenly to model-based development. The three presentations in this session highlight just a few and important pieces out of the standards puzzle defining the Model Driven Architecture and Model Integrated Computing paradigms. Please take these presentations as an invitation to explore and adopt the whole set of MDA and MIC standards.

0945 – 1015 **GME-MOF: The MOF-Based GME Metamodeling Environment**
Matthew J. Emerson, Research Assistant & Janos Sztipanovits, Professor, ISIS/Vanderbilt Univ.

This presentation will: 1) discuss our implementation of and experience with MOF as a meta-modeling language for defining DSML-s for GME; 2) evaluate native MOF constructs for meta-model composition; 3) examine the impact of the selection of meta-modeling languages on meta-programmable MIC tools; and 4) describe future directions for the integration of MOF and XMI technology with GME.

1015 – 1045 **Specifying Security and Safety Requirements and Solutions with the AADL**
Edward Colbert, President, Absolute Software Co. & Jim E. Land, High Integrity Solutions

The Society of Automotive Engineers (SAE) is developing a standard Architecture Analysis & Design Language (AADL) that is designed to support the specification and analysis of real–time, fault–tolerant, safety critical, securely partitioned, dynamically reconfigurable multi–processor system architectures. The Goal Structured Network (GSN) methodology enables a top-down approach to the analysis of the safety requirements that conform to the requirements of IEC 61508, and other non-functional engineering requirements, e.g. security. Tools are available that enable the integration of the results of the concurrent activities required by SAE ARP 4761 and the software engineering activities embodied in RTCA DO178B. This presentation will report on a study of the effectiveness of specifying security and safety requirements and solutions with the AADL and the analysis of these non-functional requirements with GSN tools.

1045 – 1100 Morning Refreshments

1100 – 1130 **Dynamic Deployment and Configuration of Component-based Applications**
Akiko Yano & Yoshiaki Tamura Northeastern University, Graduate School of Engineering

In this presentation, we describe "Dynamic Deployment and Configuration of Component-based Application" as an extension to the OMG Deployment and Configuration of Component-based Distributed Application Specification, which enables asynchronous re-configuration, re-planning, re-discovery of artifacts, re-configuration of the artifacts and dynamic launch control. The proposed protocol ensures that the imported artifacts are compatible with the target environment and properly configured on the fly. It allows that the target environment can start the initiative of re-deployment and re-configuration. Special consideration is given to protect the actually running environment in case a dynamic deployment activity would fail.

1130 – 1200 Session 3 Q&A – Discussion

1200 – 1245 Lunch

1300 – 1500 **Session 4: *MIC Tools***
Chair: Fred Waskiewicz, Director of Standards, Object Management Group

As MIC gains broader acceptance the need for tool support of large, complex systems; distributed applications; and networks continues to grow. With that growth comes new challenges. This session offers samples of next-generation MIC tools addressing the issues of collaborative model development and versioning; implementing quality of service requirements; and meeting the challenge of analysis scalability. All tools are developed using Vanderbilt's GME - the Generic Modeling Environment.

1300 – 1330 **Model-Integrated Computing in the Large, Complex Projects**
Akos Ledeczi, Senior Research Scientist, Vanderbilt University

As Model Integrated Computing gains wider and wider acceptance the scalability of the supporting tools becomes a significant issue. Large, complex projects involve many developers who create large and complex models. Supporting large-scale model integrated computing requires two key features currently lacking in available tools: 1) distributed, simultaneous multi-user access to the models; and 2) model versioning. We have implemented these features in the GME, the General Modeling Environment and integrated Clearcase and Visual Sourcesafe support. This presentation will describe our technical approach in detail and demonstrate its use.

1330 – 1400 **A Model-Based Approach to Designing QoS Adaptive Applications**
Jianming Ye & Joseph Loyall, Staff Scientists, BBN Technologies
Sandeep Neema, Sherif Abdelwahed, Nagabhushan Mahadevan, Vanderbilt University

As part of the DARPA MoBIES program, we have been developing a model-based approach for designing adaptive Quality of Service (QoS) in distributed applications. We have developed a prototype modeling tool, the Distributed QoS Modeling Environment (DQME), that captures some of the essential elements of dynamic QoS adaptation. The DQME modeling tool combines the domain-specific modeling capability of the Generic Modeling Environment (GME) with the runtime QoS adaptation mechanisms of the Quality Objects (QuO) middleware framework. DQME clearly separates the design of the QoS concerns of applications from the functional concerns. Integrated code synthesis tools facilitate code generation and model refinement. To demonstrate its capabilities in designing QoS adaptive applications, we have applied DQME to two distributed real-time embedded (DRE) applications: a signal analyzer optimization and a multi-UAV surveillance and target tracking system.

1400 – 1430 **SeMA: A Model-driven Multi-Paradigm Integrated Simulation Framework For Analysis of Communication Networks**
Amogh Kavimandan & Aniruddha Gokhale, Vanderbilt University -
Marina Thottan, Wonsuck Lee & Ramesh Viswanathan, Bell Labs, Lucent Technologies

This presentation describes the Service Modeling Architecture(SeMA)MIC tool suite used to model and analyze large network topologies. SeMA comprises a novel modeling paradigm that allows representing network behavioral dynamics and multilayered structure as a composition of hybrid systems models. The modeling paradigm is developed using the Generic Modeling Environment (GME) framework. The model interpreters can synthesize platform-specific aspects, such as ns2 scripts, or hybrid simulations in MatLab, Stateflow/Simulink or Modelica.

1430 – 1500 Session 4 Q&A - Discussion

1500 – 1900 *Demonstration Area Open*

1500 – 1530 Afternoon Refreshments

1530 – 1730 **Session 5:** *Tool Integration*
Chair: Fred Waskiewicz, Director of Standards, Object Management Group

No single tool will be able to provide all of the capability for every MIC project, especially the large and complex ones. This session offers three experience reports on tool integration - combining a suite of interoperable tools to meet project challenges. The first provides insight into integrating model transformation and execution tools with a development framework. The second offers an approach to development of model-driven integrated support architectures. The third discusses the integration of MDA tools to develop a product line architecture.

1530 – 1600 **Building an MDA Toolchain**
Gregory T. Eakman, VP of Research, Pathfinder Solutions

There are many tools associated with system development in an MDA environment: model editors, code generators, simulations and analysis tools, testing tools, deployment components, target execution environments, and domain specific tools. No single vendor will be able to provide all the necessary tools for every project, and integrating these tools should not be left to project teams. Current OMG standards such as UML 2.0, XMI, and others do not address interfaces into all of these tools. This presentation discusses our experiences in integrating our model transformation and execution tools with other tools using XMI, transformations, and more recently, the Eclipse framework.

1600 – 1630 **Development of Model-driven Integrated Support Architectures**
Stanley Ofsthun, Associate Technical Fellow, Boeing Phantom Works

This presentation will share practical experiences associated with Boeing's application of model-driven technologies to the development and integration of the various support aspects of an aerospace vehicle. Our initial efforts have focused on developing failure domain models that support both multi-disciplined engineering analysis and embedded vehicle health assessment. Using a generic fuel system example, it will be shown how models are constructed, the engineering analyses that they support, and how they can be auto-coded into executables for embedded, real time computing. Performance issues and results from both our engineering environment and our hardware/software laboratory will be discussed.

1630 – 1700 **MDA Tool Integration for Development of Real-Time Embedded Systems**
Theckla Louchios & Christopher Andrews, Software Engineers, Lockheed Martin

This presentation will discuss how we leverage the strengths of a variety of existing MDA tools to develop a product line architecture at a high enough level of abstraction that allows it to remain relevant for an extended (15+ year) product life, while optimally leveraging the strengths of current implementation technologies and providing a detailed model to allow for useful system analysis at early stages of development. Future directions that MDA tools should take will also be examined in the presentation.

1700 – 1730 Session 5 Q&A - Discussion

1730 – 1830 ***Demonstration Area Reception*** ***hosted by*** 

THURSDAY, October 14, 2004

0900 - 1200 **Session 6: *Modeling Languages***
Chair: Ben Watson, Research Program Manager, Lockheed Martin

Complex system design necessitates a variety of complementary models to express the relationships, constraints and functionality of the system. These complementary models and model views are constructed on a base of Domain Specific Modeling Languages (DSML). This session includes four presentations that describe how DSMLs are tailored to meet specific purposes such as real-time analysis, distributed processing architecture and multiple models of computation. The last presentation describes an end-to-end tools chain composed on multiple DSMLs.

0900 – 0930 **Embedded Control Systems Language for Distributed Processing (ECSL-DP)**
Sandeep Neema, Research Scientist & Gabor Karsai, Asst. Professor, ISIS, Vanderbilt University

This presentation will describe ECSL-DP, a domain-specific modeling language for designing distributed embedded automotive applications. The engineering process for embedded automotive systems design involves several complex activities supported by different tools. A difficult challenge is in provisioning integrated tool-chains with automated model transformation between different tools. ECSL-DP has been specifically designed to be extensible and integratable in a tool-chain, and the presentation will emphasize on the integrated tool-chain supporting the key activities in an end-to-end engineering process.

0930 – 1000 **Pattern-Based Analysis of an Embedded Real-time System Architecture**
Peter Feiler, Senior MTS, Software Engineering Institute

The Society of Automotive Engineers (SAE) Architecture Analysis & Design Language (AADL) standard is an architecture modeling language for real-time, fault-tolerant, scalable, embedded, multiprocessor systems. The AADL enables the development and predictable integration of highly evolvable systems as well as analysis of existing systems. This presentation discusses the role and benefits of using the AADL in the process of analyzing an existing avionics system. AADL is used to describe architecture patterns in the system and to identify potentially systemic issues. Some of the findings related to timing, scheduling, and fault tolerance will be discussed. Additionally, the benefits of working with architecture abstractions that are reflected in the AADL notation, in particular the separation of architecture design decisions from implementation decisions, will be highlighted.

1000 - 1030 Morning Refreshments

1000 - 1600 ***Demonstration Area Open***

1030 – 1100 **SRML, a Domain-specific Modeling Language**
Zsolt Kalmar, Sandeep Neema & Gabor Karsai, Institute for Software Integrated Systems
Michael S. Moore & Mark Brooks, Southwest Research Institute

This presentation will describe SRML, a domain-specific modeling language, which coupled with a suite of model-based tools in an integrated tool-chain facilitate the process of constructing Signal Analyzer. SRML, in addition to being a valuable artifact to the Signal Analyzer domain engineers, also brings to bear quite a few modeling and analysis techniques that are widely applicable, and would be of interest to the MIC community in general. The presentation will elaborate upon techniques such as integrating multiple models of computation, type polymorphic component models, automated design-time type resolution, model optimization by graph refactoring, and generative modeling.

1100 – 1130 **Embedded Systems Modeling Language and End-to-End Development Tool Chain**
Feng Shi, Sandeep Neema, Gabor Karsai, Zsolt Kalmar, Attila Vizhanyo, Aditya Agrawal, ISIS/Vanderbilt University

This presentation will talk about Embedded Systems Modeling Language (ESML) and an end-to-end development tool chain. ESML is a domain-specific graphical modeling language designed in MIC approach for modeling real-time mission computing embedded avionics applications. ESML is also designed for use in conjunction with other tools and languages. In order to support the transition between ESML models and other format model, an end-to-end tool chain is developed to form an integrated architecture for modeling, synthesizing and analysis processes.

1130 – 1200 Session 6 Q&A - Discussion

1200 - 1245 Lunch

1330 – 1530 **Session 7: *Transformations***
Chair: Ben Watson, Research Program Manager, Lockheed Martin

When multiple model views and multiple Domain Specific Modeling Languages (DSML) are used to design and implement a complex system, it is essential that the various model representations be synchronized (for consistency). Model-to-model transformations provide a mechanism for automating the required mapping from one representation to another. The presentations discuss meta-model based transformations and aspect weaving transformations.

1330 – 1400 **A Meta-model based Model Transformation Tool: GreAT**
Attila Vizhanyo, Gabor Karsai, Aditya Agrawal, Feng Shi, Zsolt Kalmar & Sandeep Neema
Institute for Software-Integrated Systems, Vanderbilt University

Model-Integrated Computing emphasizes the use of model transformations, in the spirit of MDA. We have developed a meta-programmable tool to implement model transformations, and have demonstrated its use on several examples. The tool is based on a transformation language: GReAT that uses concepts and techniques from graph transformations, but aims at building highly efficient transformations with the help of the Code Generator tool (part of the GReAT toolsuite), which compiles a GReAT program into a high-performance C++ program.

1400 – 1430 **MIC – Scaling with Multiple Languages and Multiple Models**
Sandeep Neema, Shweta Shetty, Ted Bapty, ISIS, Vanderbilt University

One of the key challenges in applying MIC to large-scale embedded system design is the confining need for providing a "super" modeling language that covers all aspects of the system design. This presentation illustrates techniques for breaking this barrier by providing a suite of small modeling languages, and developing mechanisms and tools for establishing and consistently maintaining linkages between models representing different aspects of a complex system. The presentation will demonstrate the concepts with examples from a large scale real-time physics systems being developed at Fermilab, Chicago.

1430 – 1500 **Concern Separation in Model Integrated Computing**
Jeff Gray, Assistant Professor, University of Alabama at Birmingham
Aniruddha Gokhale, Assistant Professor, Vanderbilt University

This presentation highlights the integration of the CoSMIC model-driven middleware development tool suite with the C-SAW aspect-oriented domain modeling framework to scalably and reliably evolve and adapt distributed real-time and embedded (DRE) systems. The CoSMIC tool suite comprises a set of modeling languages and platform-specific generative tools that address the configuration and deployment challenges of DRE systems. The Constraint-Specification Aspect Weaver (C-SAW) is applied to CoSMIC models to transform a model according to the characteristics of crosscutting modeling concerns. We will demonstrate the capability to quickly insert/remove new properties and policies into a model without extensive manual adaptation.

1500 – 1530 Session 7 Q&A - Discussion

1530 – 1600 Afternoon Refreshments

1600 – 1730 **Panel: *Semantic Foundation for Model Integrated Computing***
Moderator: Janos Sztipanovits, Professor, ISIS/Vanderbilt University

Modeling, model analysis and model transformation tools have fundamental roles in model-based design. These tools use Domain-Specific Modeling Languages (DSML) with varying syntax and semantics. The objective of this panel is to discuss core components of an infrastructure, which can facilitate the precise semantic specification of DSML-s without sacrificing flexibility.

Panelists: TBA

1730 - 1930 ***Workshop Reception*** ***hosted by*** 

FRIDAY, October 15, 2004

0830 - 1030 **Session 8: *Model Based Synthesis***

Chair: Sandeep Neema, Research Scientist, ISIS, Vanderbilt University

Code generation is one of the more important aspects of model-based system design. The ability to construct low-level programming artifacts, which have the potential to be "correct-by-construction" makes model-based system design attractive to the practitioner. The benefit of an abstract and semantically precise model-based representation is not completely realizable without sound techniques for code synthesis from models. This session will provide an overview of some of these techniques realized in the context of embedded systems. Presentations by technology leaders and researchers will provide the in-depth insight and the practical application of the techniques. Discussions will be geared towards understanding the best-practices, directions where these techniques may evolve, and more notably formalization of the code generators such that correctness could be guaranteed a priori alleviating the need for testing or verification of the generated programming artifact.

0830 – 0900 **Software Producibility Using Model-Based Design, Analysis and Synthesis of RT Systems**
Raj Rajkumar, Professor, Carnegie Mellon University

Time Weaver is a model-driven environment developed at Carnegie Mellon University under the DARPA MoBIES (Model-Based Integration of Embedded Systems) program for the design, analysis and synthesis of component-based real-time systems. The Time Weaver framework defines clear compositional (decompositional) semantics to integrate (decompose) functional and para-functional behaviors. This presentation will describe Time Weaver's code generation, QoS management, and schedulability analysis functions.

0900 – 0930 **Model-based Code Generation**
Jesung Kim, Researcher, University of Pennsylvania

Model-based code generation allows the system designer to focus on high-level design issues by automating the time-consuming and error-prone coding procedure. However, the design of a code generation tool imposes on us challenges when the model is a hybrid automaton that combines continuous dynamics with the FSM-based discrete control. In this presentation, we describe a code generation framework that addresses formal relationship between the model and the generated code consisting of concurrent components.

0930 – 1000 **Transformation of Executable Platform Independent Models**
Bruce Levkoff, Principal Consultant, Pathfinder Solutions

Executable models comprise a platform independent representation of full system behavior. Pathfinder Solutions uses imperative transformation rules expressed as templates combining clear text with model transformation directives to map these executable models to their target platform. These transformation rules use a domain specific language to navigate the model elements and convert the model data to text. Implementation patterns for each model element, including structural, architectural, and behavioral elements, are expressed in this language. A transformation engine accepts the templates, models, and markings and executes the template instructions to produce the target documents (code). This session describes the implementation patterns and the language used to represent them.

1000 – 1030 Session 8 Q&A – Discussion

1030 – 1045 Morning Refreshments

1045 - 1245 **Session 9: *MIC Applications and Case Studies I***

Chair: Michael S. Moore, Senior Research Engineer, Southwest Research Institute

In this double session, presenters will provide insightful examples of MIC applications from a broad array of disciplines. Examples of deeply embedded applications, such as distributed sensor networks, ultra-high performance instrumentation, and communications systems will be presented, as well as applications to the cutting edge research areas of polymorphous computing and adaptive learning systems. The breadth of applications present in this session are demonstrative of the huge potential impact of MIC on the way that computer based systems are designed, implemented, and maintained. This session should convey to the audience the benefits of MIC, and the crucial need for model based technologies to achieve general acceptance in mainstream system development.

1045 – 1125 **Model-Integrated Development of Sensor Network Applications**

Sebestyen Dora, Miklos Maroti, Peter Volgyesi & Akos Ledeczi,
Institute for Software Integrated Systems, Vanderbilt University

TinyOS is a component-based configurable operating system with a very small footprint specifically designed for severely resource-constrained devices such as the nodes in a typical sensor network. TinyOS consists of a large set of software components implementing the basic functionalities that an application might need from the given device, such as basic I/O, timers, wireless communication etc. Each application consists of application-specific components written by the application designer and a subset of the TinyOS components. This presentation will introduce TinyOS and how its concepts can be mapped to a model integrated approach. We'll detail how the design space of a sensor net application can be captured along with non-functional requirements. Finally, we'll demonstrate the GRATIS environment with special emphasis on design-space exploration.

1125 – 1205 **A Model-Integrated Design Tool for Polymorphous Embedded Systems**

Brandon Eames & Esteban Osses, Research Assistants, Institute for Software Integrated Systems, Vanderbilt University

Polymorphous embedded systems utilize coarse-grained reconfiguration offered through a new class of computer architectures to improve application performance. Due to the increased implementation complexity introduced through reconfiguration, sophisticated design tools have been developed to facilitate the development of efficient applications. This presentation introduces a model-integrated design toolset for polymorphous computing, consisting of a visual modeling language for application specification, and model-based transforms to map the problem specification onto separate design space exploration tools and compilation tools.

1205 – 1245 **Experiences in Developing Model-Integrated Tools and Technologies for Large-Scale Fault Tolerant Real-Time Embedded Systems**

Steven G. Nordstrom, Institute for Software Integrated Systems, Vanderbilt University

Conventional redundancy-based fault tolerance approaches are not appropriate for high energy physics experiments due to tremendous system cost (fault tolerance is limited to 10% overhead). A set of tools was developed at Vanderbilt University for specifying, simulating, and implementing a range of low overhead failure adaptation strategies within the context of a large scale embedded environment. The tools embrace a model-integrated approach that combines design specification, behavioral modeling, and code-generation to produce both simulations and system implementation. Experiments were conducted to evaluate the behavioral correctness, performance, and adaptability of the tools under a variety of failure scenarios.

1245 - 1330 Lunch

1345 - 1545 **Session 10: *Case Studies/Experience Reports II***
Chair: Michael S. Moore, Senior Research Engineer, Southwest Research Institute

1345 – 1425 **A Learning Technology Application of Model Integrated Computing**
Larry Howard, Sr. Research Scientist, Institute for Software Integrated Systems, Vanderbilt U.

We have employed the MIC strategy to create an authoring environment called CAPE and a deployment infrastructure called eLMS. The Courseware Authoring and Packaging Environment (CAPE) provides a visual authoring language for constructing interactive, adaptive web-based learning experiences extended, for expressiveness, with a dynamic programming language. Instructional design patterns capture canonical pedagogical strategies as abstract models, guiding development based on learning science precepts. The experimental Learning Management System (eLMS) provides a model-based delivery engine for enacting CAPE-authored learning experiences that produces a detailed delivery record for each learner. Such records inform assessments of efficacy and decisions concerning future refinements by educators.

1425 – 1505 **Applying MDA to Constrained Environments**
Gregory T. Eakman, VP of Research, Pathfinder Solutions

MDA can express fully executable system behavior in a platform independent model. This higher level of abstraction yields many benefits, but if the mapping to the target platform does not meet space and time non-functional requirements, the application of MDA to these resource-constrained systems will fail. Application developers must have access to all aspects of the platform mapping since a vendor cannot anticipate the application profiles or performance requirements of all systems. This session will cover the optimizations of implementation patterns and transformation rules we have developed with our clients in a wide range of application domains, including medical devices, telecommunications, control systems, defense, and automotive. It will also discuss the use of instrumentation to collect execution profile information to analyze performance issues.

1505 – 1545 **Applying Domain-Specific Modeling Languages to Develop Distributed, Real-time, and Embeddded Systems**
Krishnakumar Balasubramanian, Jaiganesh Balasubramanian, Jeff Parsons, Aniruddha Gokhale & Douglas C. Schmidt, Vanderbilt University

Domain-specific modeling languages (DSMLs) raise the level of abstraction at which software is developed to include domain entities as first-class language elements. Few successful DSMLs have been developed to support distributed, real-time and embedded (DRE) systems, such as avionics mission computing systems and total ship computing environments. This presentation introduces the Platform-Independent Component Modeling Language (PICML), which is a DSML in the Component Synthesis with Model Integrated Computing (CoSMIC) tool suite that integrates key development, configuration, and deployment steps in component middleware and applications for DRE systems. We demonstrate PICML in the context of applying DSMLs to develop complex DRE systems

1545 - 1550 Wrap-up / Closing Remarks

Program Committee

Dock Allen, ***MITRE***

Sushil Birla, ***General Motors***

Robert France, ***Colorado State University***

Jeff Gray, ***University of Alabama-Birmingham***

Dwayne Hardy, ***DoD-OSD***

Joe Jacob, ***Objective Interface Systems***

Manfred Koethe, ***88Solutions***

Akos Ledeczi, ***Vanderbilt University/ISIS***

Kevin Loughry, ***Object Management Group***

Bill Milam, ***Ford Motor Co.***

Michael Moore, ***Southwest Research Institute***

David Oglesby, ***Honeywell***

Raj Rajkumar, ***Carnegie Mellon University***

Mark Schulte, ***Boeing***

Oleg Sokolsky, ***University of Pennsylvania***

Fred Waskiewicz, ***Object Management Group*** **(PC Chair)**

Ben Watson, ***Lockheed Martin***

Don Wilson, ***Raytheon***

Jianming Ye, ***BBN Technologies***