AOSD 2005 Demo Proposal

Demo Title:

CoSMIC: Addressing Crosscutting Deployment and Configuration Concerns of Distributed Real-time and Embedded Systems via Aspect-oriented & Model-driven Software Development

Abstract:

Model-driven development (MDD) has been gaining importance as an approach to resolving lifecycle challenges of large-scale distributed real-time and embedded (DRE) systems (e.g., joint emergency response systems), including design, development, testing, maintenance and evolution. DRE systems are characterized by their stringent requirements for quality of service (QoS), such as predictable end-to-end latencies, timeliness and scalability. Delivering the QoS needs of DRE systems entails the need to configure correctly, fine tune and provision the infrastructure used to host the DRE systems, which crosscuts different layers of middleware, operating systems and networks. Addressing these tangled deployment and configuration concerns of DRE systems requires integrating the principles of Aspect-Oriented Software Design (AOSD) with MDD.

This demo showcases a MDD tool suite called CoSMIC that uses AOSD principles to resolve the accidental complexities arising due to the configuration and deployment crosscutting concerns of component middleware-based DRE systems. The demo will showcase a hypothetical emergency response system as a guiding example to illustrate the tools.

Presenters

(1) Name: Aniruddha S. Gokhale (Primary Contact)

Contact information

Assistant Professor
ISIS, Dept. of Electrical Engineering and Computer Science
Vanderbilt University
Box 1829, Station B
Nashville, TN 37235, USA
Email: a.gokhale@vanderbilt.edu
URL: http://www.dre.vanderbilt.edu/~gokhale
Phone: (615) 322-8754
Fax: (615) 343-7440

Brief biography

Dr. Aniruddha S. Gokhale (a.gokhale@vanderbilt.edu) is an Assistant Professor in the Electrical Engineering and Computer Science Department, and a Senior Research Scientist at the Institute for Software Integrated Systems (ISIS), both at Vanderbilt University, Nashville, TN. Dr. Gokhale's research interests are in real-time component middleware optimizations, aspect-oriented and model-driven software development applied to component middleware-based applications, and distributed resource management. He is currently leading DARPA projects involving modeling, AOSD and middleware solutions and distributed dynamic resource management. Dr. Gokhale is heading the R&D efforts on an open source model driven middleware framework called CoSMIC. Dr. Gokhale has published over 50 technical papers, and currently a patent application is pending. Dr. Gokhale was previously with Bell Laboratories, Lucent Technologies, where he worked on high performance fault tolerant CORBA, networked call centers and network element software management solutions. Dr. Gokhale received his Bachelor of Engineering (B.E., Computer Engineering) from University of Pune, India in 1989; Masters of Science (M.S., Computer Science) from Arizona State University, Tempe, AZ in 1992; and Doctor of Science (D. Sc., Computer Science) from Washington University, St. Louis, MO in 1998.

(2) Name: Arvind S. Krishna

Contact information

Graduate Research Assistant
ISIS, Dept. of Electrical Engineering and Computer Science
Vanderbilt University
Box 1829, Station B
Nashville, TN 37235, USA
Email: arvindk@dre.vanderbilt.edu
URL: http://www.dre.vanderbilt.edu/~arvindk
Phone: (615) 322-8754
Fax: (615) 343-7440

Brief Biography

Arvind S. Krishna is a PhD student in the Electrical Engineering and Computer Science Department at Vanderbilt University and a member of the Institute for Software Integrated Systems. He received his MA in management from Birla Institute for Technology and Science (BITS), Pilani, India and his MS in computer science from University of California, Irvine. His research interests include patterns, real-time Java technologies for Real-Time CORBA, model-integrated QA techniques, and tools for partial evaluation and specialization of middleware. He is a student member of the IEEE and ACM.

(2) Name: Doug Schmidt

Contact information

Professor
ISIS, Dept. of Electrical Engineering and Computer Science

Vanderbilt University
Box 1829, Station B
Nashville, TN 37235, USA
Email: d.schmidt@vanderbilt.edu
URL: http://www.dre.vanderbilt.edu/~schmidt
Phone: (615) 343-8197
Fax: (615) 343-7440

Brief biography

Dr. Douglas C. Schmidt (PI) is a Full Professor in the Electrical Engineering and Computer Science (EECS) Department and a Senior Research Scientist at the Institute for Software Integrated Systems (ISIS) at Vanderbilt University, Nashville, TN. For the past two decades, he has led pioneering research on patterns, optimization techniques, and empirical analyses of object-oriented and component-based frameworks and model-driven development tools that facilitate the development of distributed real-time and embedded (DRE) middleware and applications on parallel platforms running over high-speed networks and embedded system interconnects. Dr. Schmidt is an expert on distributed object computing patterns, middleware frameworks, and Real-time CORBA and has published over 250 technical papers and books that cover a range of topics including high-performance communication software systems, parallel processing for high-speed networking protocols, real-time distributed object computing with CORBA, object-oriented patterns for concurrent and distributed systems, and model-driven middleware tools. In addition to his academic research, Dr. Schmidt has over fifteen years of experience leading the development of ACE, TAO, and CIAO (www.dre.vanderbilt.edu), which are widely used, open-source DRE middleware frameworks that contain a rich set of components and domain-specific languages that implement key patterns for high-performance DRE systems. Dr. Schmidt received B.S. and M.A. degrees in Sociology from the College of William and Mary in Williamsburg, Virginia, and an M.S. and a Ph.D. in Computer Science from the University of California, Irvine in 1984, 1986, 1990, and 1994, respectively. URL: www.dre.vanderbilt.edu/~schmidt. Email: schmidt@dre.vanderbilt.edu.

Demo Details

Demo Synopsis

The proposed demo will provide attendees with a key understanding of the challenges and solutions for modeling crosscutting concerns in Distributed Real-time and Embedded (DRE) systems. The participants will gain hands-on experience using the CoSMIC Model-driven Development (MDD) technology using a hypothetical DRE system as a guiding example. CoSMIC consists of an integrated collection of modeling, analysis, and synthesis tools that address key lifecycle crosscutting challenges of large-scale DRE middleware and applications. A key focus of the CoSMIC demo is the description of modeling abstractions to separate concerns that are scattered across various configuration files, such as the XML descriptions typically encountered in middleware deployment and configuration.

Challenge Problems Addressed by the Demo

Quality of service (QoS)-enabled component middleware, such as lightweight CORBA Component Model (CCM) (and to some extent J2EE and .Net for the domains they are used in) are increasingly being used to develop and control mission-critical, large-scale DRE systems. Figure 1 illustrates a representative DRE system highlighting a joint emergency response system. DRE systems share the following characteristics:

1. **Heterogeneity.** Large-scale DRE systems often run on a variety of computing platforms that are interconnected by different types of networking technologies with varying QoS properties. The efficiency and predictability of DRE systems built using different infrastructure components varies according to the type of computing platform and interconnection technology.

Figure 1: Joint Emergency Response DRE System

2. **Deeply embedded properties**. DRE systems are frequently composed of multiple embedded subsystems. For example, an anti-lock braking software control system forms a resource-constrained subsystem that is part of a larger DRE application controlling the overall operation of an automobile.
3. **Simultaneous support for multiple QoS properties**. DRE software controllers are increasingly replacing mechanical and human control of critical systems. These controllers must simultaneously support many challenging QoS constraints, including (1) real-time requirements, such as low latency and bounded jitter, (2) availability requirements, such as fault propagation/recovery across distribution boundaries, (3) security requirements, such as appropriate authentication and authorization, and (4) physical requirements, such as limited weight, power consumption, and memory footprint. For example, a distributed patient monitoring system requires predictable, reliable, and secure monitoring of patient health data that can be distributed in a timely manner to healthcare providers.
4. **Large-scale, network-centric operation**. The scale and complexity of DRE systems makes it infeasible to deploy them in disconnected, standalone configurations. The functionality of DRE systems is therefore partitioned and distributed over a range of networks. For example, an urban bio-terrorist evacuation capability requires highly distributed functionality involving networks connecting command and control centers with bio-sensors that collect data from police, hospitals, and urban traffic management systems.
5. **Dynamic operating conditions.** Operating conditions for large-scale DRE systems can change dynamically, resulting in the need for appropriate adaptation and resource management strategies for continued successful system operation. In civilian contexts, for instance, power outages underscore the need to detect failures in a timely manner and adapt in real-time to maintain mission-critical power grid operations. In military contexts, likewise, a mission mode change or loss of functionality due to an attack in combat operations requires adaptation and resource reallocation to continue with mission-critical capabilities.

Relevance to AOSD: Concern Separation in DRE Systems via Model Driven Development

MDD technologies, such as the Object Management Group (OMG)'s Model-driven Architecture (MDA), have emerged to address the different lifecycle challenges of DRE systems outlined above, which includes satisfying both the functional and QoS needs of DRE systems during design, development, testing, maintenance, and evolution stages. Figure 2 illustrates how a MDD tool chain coordinates with design-time analysis tools and run-time infrastructure (such as middleware, OS and networks) to address DRE system lifecycle challenges described below:

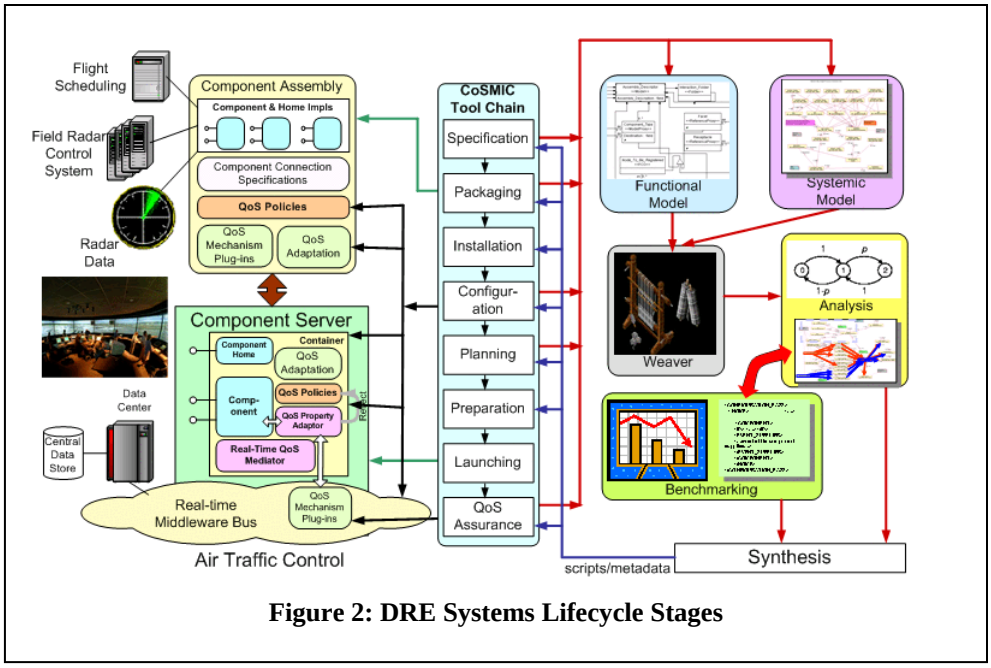

Figure 2: DRE Systems Lifecycle Stages

- *Specification and implementation*, which enables application functionality specification, partitioning, and implementation as components.
- *Packaging*, which allows bundling a suite of software binary modules and metadata representing application components.
- *Installation*, which involves populating a repository with the packages required by the application.
- *Configuration*, which allows configuration of the packages with the appropriate parameters to satisfy the functional and systemic requirements of application without constraining to any physical resources.
- *Planning*, which makes appropriate deployment decisions including identifying the entities, such as CPUs, of the target environment where the packages will be deployed.
- *Preparation*, which moves the binaries to the identified entities of the target environment.
- *Launching*, which triggers the installed binaries and bringing the application to a ready state.
- *Adaptation*, which enables run-time reconfiguration and resource management to maintain end to-end QoS.

Figure 2 also demonstrates how the various stages of the DRE systems lifecycle are tangled with different layers of the infrastructure that host the DRE systems. MDD tools that use AOSD techniques can be used in untangling the crosscutting concerns at each stage of the DRE lifecycle,

which helps improve productivity and time-to-market. This demo will showcase how the CoSMIC AOSD & MDD tool suite addresses DRE systems lifecycle tangled concerns. The demo will use an emergency response system as a guiding example. Below we provide a brief overview of the CoSMIC and C-SAW tools.

The Design and Implementation of CoSMIC AOSD & MDD Tool suite

The deployment and configuration of middleware typically involves manual modification to meta-data that is split across multiple XML descriptors. A key evolution and change management problem exists because of the inter-dependencies and crosscutting between descriptors, and the fact that manual modification of large XML files is error-prone. The *Component Synthesis with Model Integrated Computing* (CoSMIC) tools are developed using the Generic Modeling Environment (GME), which is a metamodeling environment that defines the modeling paradigms for each stage of the CoSMIC tool chain. The CoSMIC MDD toolsuite illustrated in Figure 3 is a collection of domain-specific modeling languages (DSMLs) and generative tools to address the concerns at different stages of DRE systems lifecycle shown in Figure 2.

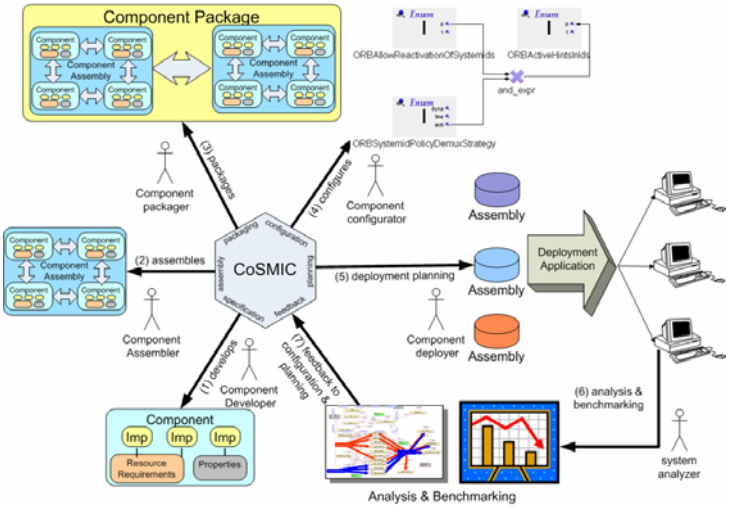

Figure 3: CoSMIC AOSD & MDD Tool suite

Underlying Implementation Techniques and Technologies

The CoSMIC tool suite uses GME to enforce "correct by construction" techniques, as opposed to the "construct by correction" techniques commonly used by post-construction tools, such as compilers, source-level debuggers, and script validators. CoSMIC ensures that the rules of construction – and the models constructed according to these rules – can evolve together over time. Each CoSMIC tool synthesizes metadata in XML for use in the underlying middleware. The CoSMIC tool suite currently uses a platform-specific model approach that integrates the modeling technology with QoS-enabled component middleware, such as CIAO (available from Vanderbilt University). CoSMIC provides the capability to inter work with third party model checking tools, such as Cadena from Kansas State University, and aspect model weavers, such as C-SAW.

Relation to other Industrial or Research Efforts

Related efforts to CoSMIC tool suite involve the upcoming commercial Software Factories tool suite, which will be bundled with Microsoft Visual Studio 2005, as well as open source research

efforts, such as Eclipse Modeling Framework. CoSMIC is also closely aligned with the OMG's Model Driven Architecture.

Relevancy for Practitioners

The CoSMIC tool suite showcases a model-driven software development technology that addresses some of the very important crosscutting concerns of DRE systems that appear at various stages of its lifecycle.

Demo Format

The demo will cover the topics according to the prescribed schedule listed below.

10 mins	Description of stages of DRE systems lifecycle elaborating on the deployment and configuration crosscutting concerns; Introducing the hypothetical emergency response system guiding example; Introduction to the component modeling, assembly and packaging DSML of CoSMIC called PICML; A brief overview of other DSMLs in CoSMIC.
20 mins	**CoSMIC AOSD & MDD tool suite** Demonstrate metamodeling in GME by focusing on the PICML DSML of CoSMIC. Illustrate the use of pluggable generators in GME by showcasing CoSMIC's generative tools. Demonstrate modeling a hypothetical emergency response architecture using PICML highlighting its ability to separate crosscutting deployment and configuration concerns. Showcase generative capabilities that weave configuration and deployment artifacts into middleware-based DRE systems.
5 mins:	**Wrap-Up and Remaining Questions from Attendees**

Expected Audience:

Industry representatives and academics interested in understanding the role and promise of AOSD in conjunction with model-driven software development.

Prerequisites:

Preliminary understanding of AOSD concepts, modeling languages, such as UML, and middleware platforms, such as J2EE, .Net or CORBA Component Model

Previous Venues:

The material has been presented at DARPA-sponsored meetings and industrial sponsor locations.

Required Equipment for Presenter:

PC/laptop projector

Additional Material:

We will make available a poster of CoSMIC during the conference for attendees to make observations and discuss these topics with the presenters.

Background Material:

Component Synthesis with Model Integrated Computing (CoSMIC)

http://www.dre.vanderbilt.edu/cosmic

Component Integrated ACE ORB (CIAO)

http://www.dre.vanderbilt.edu/CIAO

The Generic Modeling Environment (GME)

http://www.isis.vanderbilt.edu/Projects/gme/