

# Applying the Model Driven Architecture to Distributed Real-time and Embedded Applications

Aniruddha Gokhale  
a.gokhale@vanderbilt.edu

Institute for Software  
Integrated Systems  
Vanderbilt University  
P O Box 36, Peabody  
Nashville, TN 37203

Douglas C. Schmidt  
schmidt@uci.edu

Dept. of Electrical  
and Computer Engineering  
University of California  
616E Engineering Tower  
Irvine, CA 92697

Balachandran Natarajan, Nanbor Wang  
{bala, nanbor}@cs.wustl.edu

Dept. of Computer Science  
Washington University  
One Brookings Drive  
St. Louis, MO 63130

## Abstract

*Advances in commercial distributed object computing (DOC) middleware technologies are enhancing the affordability and reducing the time-to-market of distributed real-time and embedded (DRE) applications, such as commercial and military aircraft and satellites control, automobile engines management system, chemical and manufacturing plants process control, and hospital patient monitoring equipment. DOC middleware encapsulates specific services or sets of services to provide reusable building blocks that can be composed to create DRE applications rapidly and robustly. Commercial DOC middleware, such as CORBA, Java RMI and COM+, offer “horizontal” infrastructure services, such as object request brokers, and “vertical” models of domain concepts, such as common semantics for higher-level reusable component-based services, and “connector” mechanisms between components, such as remote method invocations or message passing.*

*Despite the many benefits of DOC middleware, however, developers of DRE applications face two key unresolved challenges:*

- 1. Existing middleware technologies do not yet provide a complete end-to-end solution to build DRE applications. Many DRE applications require that the underlying middleware technology platform be available on heterogeneous platforms and languages, interface with legacy code in different languages and interoperate with multiple technologies from different vendors.*
- 2. Conventional implementations of DOC middleware do not support DRE applications that possess multiple quality of service (QoS) demands, such as efficiency, scalability, dependability, and security that cross-cut multiple layers and require end-to-end enforcement.*

*A promising way to address these challenges is to apply Model-Integrated Computing (MIC) technologies. MIC is a paradigm for developing application functionality and QoS requirements at higher levels of abstraction than is possible with conventional programming languages like C++, Java or C#. Sophisticated MIC generator tools and aspect weavers can be applied to analyze the models and synthesize platform-specific code that is customized for specific middleware and application properties, such as priorities of the critical tasks, thread-pool configurations, timeout policies, end-to-end invocation latencies and reliability properties. A portable, component-based middleware framework that supports configurable and adaptive QoS management is essential to the success of model-integrated computing.*

*The Model-Driven Architecture (MDA) proposed by the Object Management Group (OMG) builds upon years of research on the MIC paradigm to provide standards-based platform-independent models (PIMs) and platform-specific models (PSMs) that streamline platform integration issues and protect investments against the uncertainty of changing platform technology. PIM and PSM descriptions of applications are formal specifications built using modeling standards, such as the Unified Modeling Language (UML). The PIM models are mapped into PSMs via translators. PSMs can be implemented using standard middleware platforms, such as CORBA. The MDA technology is, however, not yet matured to the point of supporting DRE applications.*

*Our presentation provides two contributions to the study of using MDA for DRE applications. First, we illustrate how the MDA paradigm can be applied to simplify the development of DRE applications and reusable, QoS-enabled component services. Second, we describe how pattern-oriented middleware enables modeling and synthesis tools to rapidly develop, as-*

*semble, and deploy DOC middleware and component services tailored for the needs of DRE systems.*

**Keywords:** Middleware, Model-Integrated Computing, Model Driven Architectures, Architectural and CORBA Component Model