Applying Model-Integrated Computing & DRE Middleware to High-Performance Embedded Computing Applications

Douglas C. Schmidt (schmidt@uci.edu)
Aniruddha Gokhale (a.gokhale@vanderbilt.edu)
Chris Gill (cdgill@cse.wustl.edu)
Motivation

Context
• HPEC systems are widely used for signal & image processing applications with stringent QoS distributed real-time & embedded (DRE) requirements

Problem
• Due to weight, power, real-time, & footprint constraints, HPEC software has not traditionally been able to leverage advances in COTS middleware

Promising Solution
• Use standards-based DRE middleware supporting multidimensional QoS properties (RT-CORBA, DP-CORBA, load balancing)
DRE Middleware in HPEC Applications

High-performance, real-time, fault-tolerant, & secure systems

Autonomous distributed embedded systems

Power-aware ad hoc, mobile & embedded systems

MISSION-CRITICAL REAL-TIME SYSTEMS

COTS HARDWARE & SOFTWARE
CORBA is a middleware standard.

Real-time CORBA adds QoS to classic CORBA to control:

1. **Processor Resources**
   - Thread pools
   - Priority models
   - Portable priorities
   - Standard synchronizers
   - Static scheduling service

2. **Communication Resources**
   - Protocol policies
   - Explicit binding

3. **Memory Resources**
   - Request buffering

These capabilities address some (but by no means all) important HPEC application development & QoS-enforcement challenges.
Data Parallel CORBA bridges the gap between traditional CORBA applications & high-performance embedded parallel processing applications as follows:

- Enable CORBA applications over clusters of computers
- No change required in software technologies, methodologies, or tools
- Enable massively parallel applications to integrate easily with distributed systems
- Allow parallel applications to benefit from distributed object methodologies, technologies, & tools
- Add parallelism & data distribution to the transparencies offered by CORBA
- Enable a new class of applications e.g., financial, industrial, medical, aerospace, multimedia, and military domains
Problems with Standard DRE Middleware

Too many “standards”
- Proliferation of middleware technologies – CORBA, Java EJB/RMI, COM+/.NET
- No one-size-fits all
- Accidental complexities assembling, integrating & deploying software systems

Lack of coordination in standards committees addressing multiple QoS properties
- Several independent standards each addressing a single dimension of QoS
  - e.g., Real-time CORBA, Fault tolerant CORBA, Load Balancing, Data Parallel CORBA, etc.

Promising Solution
- Integrate Model-Integrated Computing & the OMG Model Driven Architecture with Multi-QoS DRE middleware
Integrating MIC with DRE Middleware

Proliferation of middleware
- UML modeling tools used to model DRE application behavior
- Model-first/generate-next strategy for finer grained control in components

Simultaneous support for multiple QoS dimensions
- Model overall application QoS & partitioning
- Compose application servers
- Model & synthesize components
- Validate & deploy

Accidental Complexities
- Synthesize container QoS configurations & metadata
Model-Integrated Computing (MIC) Overview

- Applies domain-specific modeling languages to computing systems
  - Provides rich modeling environment including model analysis & model-based program synthesis
  - Modeling of integrated end-to-end view of applications with interdependencies
  - Modeling languages & environments themselves can be modeled as meta-models
- **Examples**
  - Generic Modeling Environment (www.isis.vanderbilt.edu)
  - Ptolemy (www.eecs.berkeley.edu)
- Based on DARPA MoBIES program

**Analyze** – different but interdependent characteristics of DRE system behavior

**Synthesize** – platform-specific code customized for DRE application
Model Driven Architecture (MDA) Overview

- OMG standardization of MIC paradigm
  - Defines platform-independent models (PIMs) and platform-specific Models (PSMs)
  - Uses Unified Modeling Language (UML) for modeling
  - Real-time profile
  - Dynamic scheduling profile
- Meta Object Facility (MoF) serves as meta-model repository
- XML Metadata Interchange (XMI) for meta-model exchange
- Efforts are underway to integrate DARPA MoBIES/MIC with OMG MDA
  - Contact John Bay <jbay@darpa.mil>

www.omg.org/mda
Component Integrated ACE ORB (CIAO)

CCM incarnation of TAO Real-time ORB

- Support development via composition
- Providing CCM framework
- QoS-aware
  - Decouple QoS policies specification from component implementations
  - Specify QoS policies in component assembly descriptors
- Configurable
  - Leveraging hardware capabilities
  - Composing QoS supporting mechanisms for CCM application servers

deuce.doc.wustl.edu/CIAO
Integrating DP-CORBA & Load Balancing

Meta Resource Broker

- DP-CORBA incarnation of TAO
- Broker for resources
- Maintains abstractions of concrete resources & their usage
- Uses DP-CORBA to partition data
- Uses TAO load balancing service to make optimal use of resources

QuO (quo.bbn.com) provides flexible QoS management
MRB uses QuO for QoS monitoring and adaptation
Component Synthesis with MIC (CoSMIC)

- Synthesizes code & configuration metadata for the CIAO CORBA Component middleware
  - Reusing components via compositions vs. generating new component implementations
  - Composition of applications components & CIAO plug-ins
  - CIAO helps instantiating application processes

- MDA tool suite
  - UML modeling using GME
  - Analysis & synthesis tools
  - Enhancement to GME tool
  - Uses MDA standards-based approach

- DARPA PCES program seeks solutions to integrate MIC with DRE Middleware
  - Contact Douglas Schmidt
    <dschmidt@darpa.mil>
Summary of Ongoing Work

- RT/CCM (CIAO) being implemented at Washington University & Vanderbilt University
- DP-CORBA being implemented jointly between Vanderbilt University & UC Irvine
- CoSMIC tool suite, MRB being developed at Vanderbilt University
- All results will be available as open-source software

[Links]
- deuce.doc.wustl.edu/CIAO
- www.isis.vanderbilt.edu/{CoSMIC,DP-CORBA}
Benefits of Integrating MIC & DRE Middleware

- Large portions of application code can be composed from prevalidated, reusable middleware components.
- Helps in the rapid assembling and deployment of applications in the face of changing business or government rules and procedures.
- Makes middleware more robust by automating the configuration of QoS-critical aspects, such as concurrency, distributions, security, and dependability.
- MIC helps bridge the interoperability problems between different middleware for which no standard solutions exist.

www.isis.vanderbilt.edu
DRE Middleware Relationship with HPEC GRID

**Grid Characteristics**
- Simultaneous multiple QoS properties
- Multiple resources from multiple providers

**Middleware Examples:**
- Globus, ICENI, Legion

**Grid Challenges**
- Tight coupling with grid infrastructure middleware (GIM)
- Accidental complexities of assembling & deploying grid applications
- Difficulties in resource & QoS assurance

Applying Standard Real-Time & Data Parallel CORBA Enhances HPEC
- Extends Current Grid Capabilities
- Leverages Model Integrated Computing
Concluding Remarks

- DRE application development challenges resolved by combining MIC/MDA & QoS-enabled component middleware
- The CoSMIC & CIAO projects are applying MIC/MDA to support DRE CORBA-based HPEC applications