

Extending Real-Time CORBA for Next-Generation Distributed Real-Time Mission-Critical Systems

Christopher Gill and Ron Cytron

{cdgill, cytron}@cs.wustl.edu

Department of Computer Science

Washington University, St.Louis*

Next generation distributed real-time mission-critical systems, such as integrated distributed medical information systems, collaborating teams of manufacturing robots, and wide-scale sensor/actuator grids for system damage detection and management, must adapt swiftly to changing environmental conditions. Greater coordination allows elements at all levels to identify and effectively respond to transient opportunities and hazards. Achieving significant levels of coordination requires: (1) systems with an empirically demonstrated ability to accommodate unplanned tasks and changing task priorities in an evolving distributed information and resource availability environment; (2) the ability to share information and control at multiple scales of distribution and timeliness; (3) the ability to trade performance of individual elements for system-level real-time performance objectives; (4) the ability to perform adaptive resource reallocations within bounded time-scales; and (5) supporting distribution infrastructure to manage resources effectively across both distribution and time-scale boundaries.

These types of systems have historically been developed largely from scratch, using handcrafted optimizations on each endsystem to achieve the coordination and performance goals outlined above. Unfortunately, expectations of increasing scale and decreasing development cycles make it hard to sustain this development model in a cost-effective manner over long system lifecycles. Solutions built instead using COTS middleware—in particular Real-time CORBA, offer greater reuse of software architectures, patterns, frameworks, analysis techniques, and testing and certification results across entire families of systems. However, these next-generation distributed real-time mission-critical systems require explicit interfaces and mechanisms for key capabilities, such as fine-grain adaptive admission control, that are not available in current-generation COTS middleware solutions, such as Real-time CORBA 1.0. Emerging COTS middleware approaches, such as Dynamic Scheduling Real-Time CORBA and the Real-Time Specification for Java[™], add essential elements for implementing these capabilities, e.g., enhanced threading models and real-time behavioral descriptors. However, unified higher-level approaches and services are needed to realize the

full real-time performance benefits achievable with closer integration of mechanisms for scheduling and admission control in middleware.

To achieve both (1) re-use and flexibility across families of systems and (2) optimized real-time performance in these systems, this presentation will describe the following key enhancements to the Real-time CORBA 1.0 and Dynamic Scheduling Real-Time CORBA approaches: (1) combining static scheduling techniques with flexible “modeless” execution, to relieve requirements for a priori knowledge of exact resource allocations and the order of transitions between allocations; (2) support for variable period tasks, to exploit degrees of freedom in performance of individual elements to achieve system-wide real-time properties; (3) flexible strategies for selecting rates and determining execution eligibility, to provide applicability of this approach across arbitrary operation characteristics, e.g., criticality and projected execution time; (4) cooperative interaction between resource managers at multiple architectural levels, to allow effective management of resources across different time-scales; and (5) strategized integration of mechanisms for admission control and resource reallocation, to achieve rapid local adaptation to run-time variations in system requirements and resource availability. This presentation will also discuss how these techniques can be applied to various operational distributed real-time mission-critical systems, such as medical information systems, collaborating robot teams, and wide-scale sensor/actuator grids.

*This work was funded in part by Boeing.