Use-cases for TAO’s Real-time ORB Endsystem Architecture

Douglas C. Schmidt and David L. Levine
schmidt@cs.wustl.edu and levine@cs.wustl.edu

Washington University, St. Louis

Design Requirements and Forces

- Integrate real-time scheduling/dispatching in ORB and I/O subsystem
- Provide all applications with real-time capabilities, not just event-driven apps
- Minimize context switching and priority inversion
- Remain as CORBA-compliant as possible
- Interoperate with JSF Ada architecture

Request Reception Use-case

- Design Overview
  - Each thread has its own “ORB” instance
  - All ORB state is stored per-thread
  - Thus, no locking is necessary within an ORB/thread
  - Servants can be shared amongst rate groups
  - But they must then provide locking

- Synopsis
  - I/O subsystem uses port numbers to demux requests to queues and RT threads per rate group
  - A Reactor demuxes/dispatches requests for each rate group
Event Channel Reception Use-case

- **Synopsis**
  - Event Channel threads handle event importance and dependencies
  - I/O subsystem and ORB Core handle priorities