


# *What's Happening in (Real-Time) CORBA?*

*A Briefing to  
the Open Group's  
Real-Time Forum*



Presented by: Dock Allen  
The MITRE Corporation  
Chair of the OMG Realtime SIG  
(781) 271 8216  
Dock@MITRE.org

# *Topics*


- 
- The Story
  - Scope
  - Why Would You Use CORBA?
  - What's Available NOW?
  - What's in the Works
  - Coming Attractions
  - Summary

# *The Story*



- *Once Upon a Time* (in 1995) a Small Group of Real-time Fanatics Banded Together to Co-opt a Commercial Software Standard for Their Own Use.
- We Wanted to Be Able to Use CORBA In Real-time, Fault Tolerant, High Performance, Embedded, and a Bunch of Other Really Interesting Kinds of Systems.
- We succeeded -- (*end of story*)

# Scope

- 
- Scope of the OMG Realtime SIG
    - CORBA and its protocols
    - UML
  - Our Main Focus is Real-Time
  - We also cover:
    - Embedded Systems
    - Fault Tolerance
    - Safety Critical Systems
    - Temporality
    - High Performance Systems
      - parallel processing
      - clusters
      - shared data

# *Why Would You Use CORBA?*



- Technical Reasons
- Other Reasons

# *Why Would You Use CORBA?*

## *Technical*



- Object Orientation, Interoperability, Heterogeneity, Transparent Distribution!
  - OO, *Interoperability*, and Distribution *Out-of-the-box*
  - Interoperability *Across Languages* (Java, C++, C, Ada95, Smalltalk, Common LISP, COBOL,...)
  - Interoperability and Portability *across Operating-Systems and Networks* (CORBA is Available on Virtually Every OS and RTOS That You Might Want to Use)
  - Distribution / Location Transparency are Fundamental
- A Large Set of General-purpose and Domain-specific Services (such as Events/notification, Time, Transactions, Workflow, ...)

# *Why Would You Use CORBA?*

## *Other*

- **OMG's Willingness to Support Non-mainstream Systems**
  - Real-Time (even Hard Real-Time)
  - Embedded Systems      - Safety Critical Systems      -...
  - Fault Tolerance      - Parallel Processing
- **Rapid Turn-around Commercial Open Standards Process**
  - As Little as 15 Months from Requirement to an Adopted Standard
- **Real-Time Extensions to DII/COE Include Real-time CORBA as the Middle-ware**
  - Two Real-time ORBS Are Being Segmented

# *OK. So What's Available NOW?*

- ***Real-Time:*** A Pure Extension to CORBA for Hard Real-Time Systems (Real-Time 1.0)
- ***Time:*** Support for Synchronized Time and Specialized Clocks
- ***Object Interactions:*** Synchronous, Asynchronous and Periodic Execution Models and Events
- ***Embedded CORBA*** (Minimum CORBA)
- ***Fault Tolerance*** (Using Redundancy)
- ***Extensibility*** (Interceptors) API

# *What's Available NOW: Real-Time CORBA 1.0*



- A Pure Extension to CORBA
  - Interoperable with Non-RT CORBA (CORBA Priority Not Used on the Non-RT Side)
- Principle Thrusts:
  - Full *Use of Priorities*, Eliminating Priority Inversion
  - Control Over *Resource* Usage
- Based on *Fixed Priority Scheduling*
- *Products available now* (at least 4 vendors)

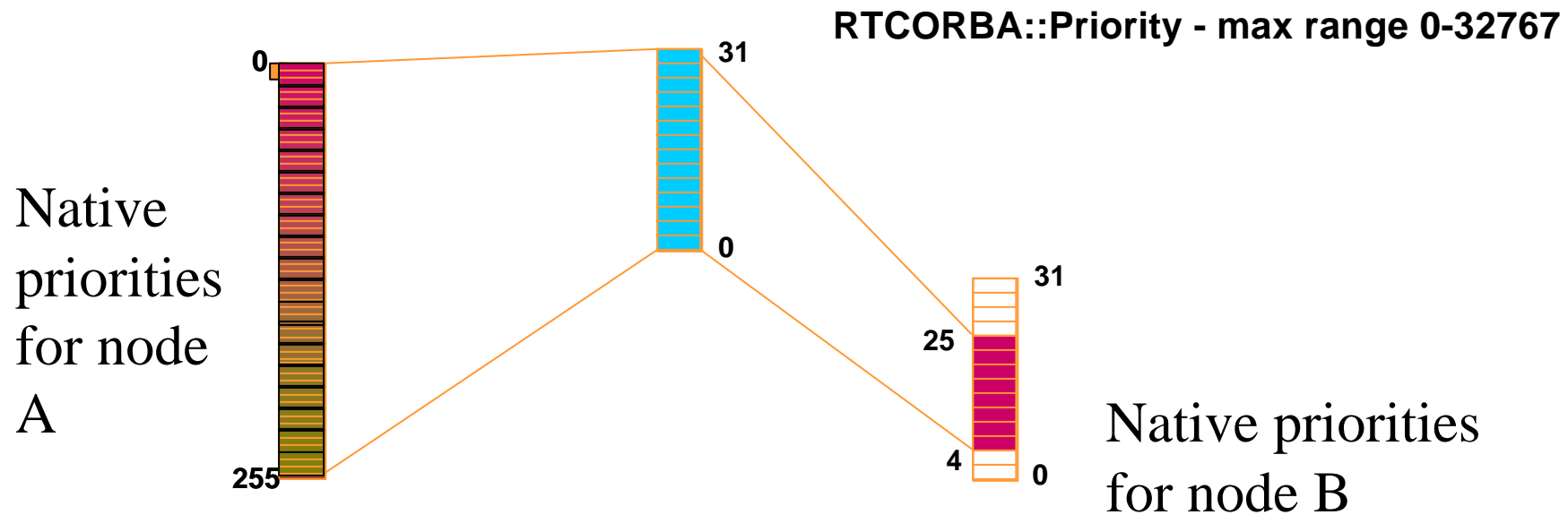
# *Real-Time CORBA 1.0*

## *Features*

- Global ***CORBA Priority***
  - CORBA Priority Can Be Propagated Along a Chain of Requests
- User-defined Priority Range for any ORB Internal Threads
- User-defined ***Thread Pools*** Tied to Priorities
- Control of ***Connection Multiplexing***
  - Including “Priority banding” and No-multiplexing
- Protocol Selection and Configuration
- ***Mutexes*** That the ORB and Applications Can Share
- Optional Scheduler Service

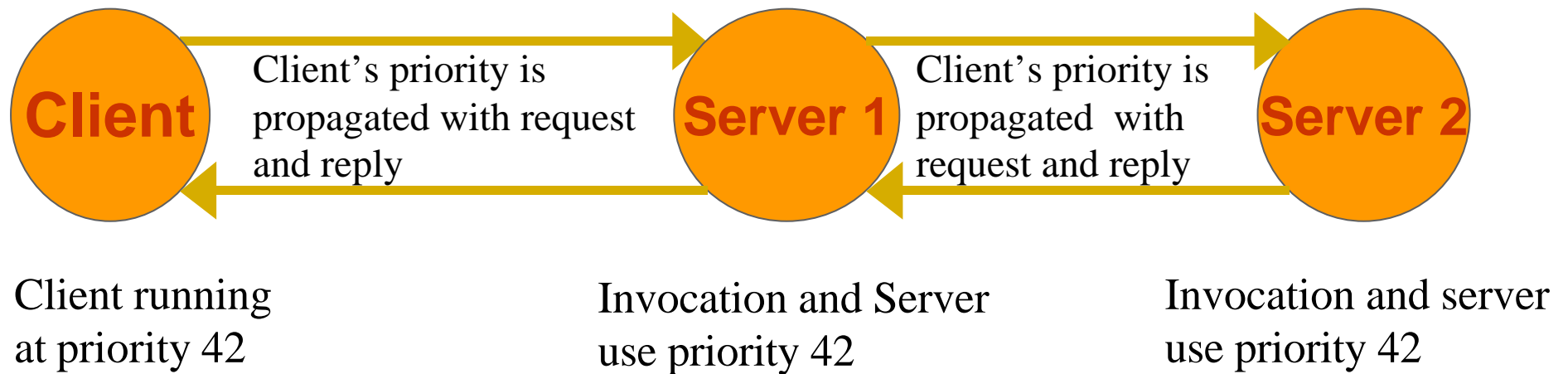
# Real-Time CORBA 1.0: System-wide Priorities

- CORBA Priority
  - Global, OS-independent Priority
  - Node-specific Functions to Map between CORBA and Native OS Priorities
    - Mapping Onto Native OS priorities Can be User-provided
  - CORBA Priority is Set Within the Current Thread



# *Real-Time CORBA 1.0: Server Priorities*

- Client CORBA priority propagation model (default)

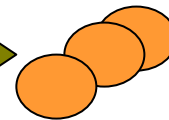
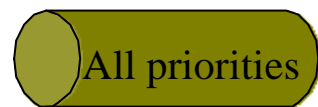
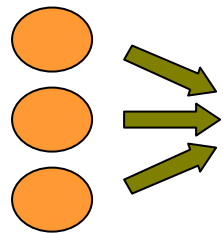


# Real-Time CORBA 1.0: Connection Management

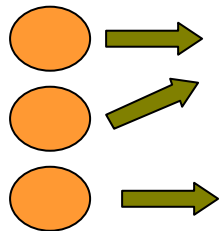
Clients

Connections

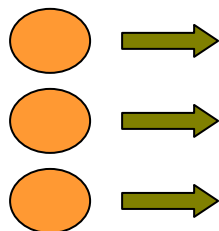
Server(s)



Single connection forces requests into FIFO order



Priority banding allows the user to control which priorities share each connection, controlling priority inversion



Dedication connections

# *What's Available NOW: Enhanced Support for Time*



- Traditional *Calendar* Time
- *User-defined Clocks* (for Your System Clock, Real-Time Clock, Mission Time, Etc)
- *Synchronized Time*

## *What's Available NOW:*

### *Asynchronous / Periodic Execution / Events*



- *Synchronous* and Deferred Synchronous  
Invocations with timeouts
- *One Ways* (Fire and Forget)
- Various Forms of *Asynchrony* with timeouts
- *Periodic* Execution
- *Events* / Notifications (Publish/Subscribe)

# *What's Available NOW:*

## *Minimum CORBA*

A decorative graphic consisting of a horizontal bar with a gradient from dark purple to orange, ending in a rounded arrowhead pointing to the right. The bar is positioned below the title and above the list.

- A *Pre-defined Profile* of CORBA for Embedded Systems
- Eliminates Support for Dynamic-style of Invocations
- A “Portable” Subset that you can Rely on for Embedded Systems

# *What's Available NOW:* *Fault Tolerance Using Redundancy*



- Ability to *Replicate Software Objects* for Fault Tolerance
- *Hot, Warm, and Cold* Spares / Active and Passive Redundancy


## *What's Available NOW:*

### *Extensibility (CORBA Portable Interceptors)*



- Well-defined Points for “*Intercepting*” CORBA Requests
- Allows Users and Third-parties to Add Capabilities Such as:
  - Performance Monitoring
  - Debugging and Testing
  - Security
  - Scheduling

# *What's in the Works*

- 
- *Real-time UML* Support
  - *Dynamic* Real-Time and Distributable Threads
  - Real-time *Notification*
  - Data *Parallel* Support
  - *Multi-cast*
  - *Extensible Transports*

# *In the Works:* *Real-Time UML*



- What is it?
  - Extensions/Profiles for UML to Support
    - expressing *time*
    - *schedulability* analysis
    - *performance* analysis
- Current Status
  - Initial Submission was Presented, *Final* Submission is Coming up

# *In the Works:*

## *Dynamic Scheduling / Distributable Threads*



- What is it?
  - Extension to Real-time CORBA to Allow Other Forms of Scheduling (EDF, LL...)
    - Needed for Systems That Can't Be Statically (a Priori) Scheduled; Systems With Dynamic Work Loads
  - Includes a Concept of “Distributable Threads”
- Current Status
  - Initial Submissions Have Been Presented, *Final* Submission Is Coming up

# *In the Works:*

## *Real-Time Notification*



- What is it?
  - Integrate the CORBA Notification Service with Real-time CORBA
  - Possible Simplification of the Notification Service
- Current status
  - RFP Issued, *Initial* Submissions Coming up

# *In the Works:*

## *Data Parallel CORBA*



- What is it?
  - Add Support for “Parallel” Objects in CORBA
  - Support Interactions Between Normal Objects and Parallel Objects
  - Based on DARPA Data Reorganization Work
- Current Status
  - Initial Submission Presented, *Final* Submission Coming up

# *In the Works:*

## *Multi-cast*



- What is it?
  - Provide Support for CORBA Over Unreliable-multicast Protocols
- Current Status
  - Initial Submissions Presented, *Final* Submissions Coming up

# *In the Works:*

## *Extensible Transport Framework*



- What is it?
  - A “Framework” to Allow Users and Third Parties to Plug in Their Transports Into ORBs Using a Standard Interface
  - Useful for Specialized Protocols, Legacy Systems, and Performance Tuning
- Current Status
  - RFP Issued, *Initial* Submissions Coming up

# *Possible Future Attractions*



- Improving Real-Time and Embedded Support
- Expanding the Availability of Real-Time Support to New Classes of Applications
- High Performance Initiatives
- More Work in the “-ilities”

*Possible Future Attractions:*  
*Improved Real-Time and Embedded Support*



- *Mapping Real-time Java* onto Real-Time CORBA
- Additional *Scheduling Disciplines*
- *QoS for UML and XML*
- CORBA Support for *Smart Transducers*

*Possible Future Attractions:*  
*Expanded Availability of Real-Time Support*



- Real-Time *Transactions*
- *Temporal Reasoning* Support
- Real-Time *Security*
- Real-Time *Fault Tolerance*

# *Possible Future Attractions: High Performance Initiatives*



- *Load Balancing*
- *Reliable Multicast*
- *Data-centric Publish/Subscribe*

*Possible Future Attractions:  
More Work on the “ilities”*



- *On-line Updates* (for non-stop systems)
- *UML Profiles for Fault Tolerance*
- *Safety Critical Systems Support*

# *Summary*



- Real-Time CORBA is Here Now
- and It Just Keeps Getting Better
- So, WHAT DO YOU NEED NEXT?

---

---

---

---

# *Straw Poll on a few of the RT SIG New Work Items*



- |                          |                                      |
|--------------------------|--------------------------------------|
| ___ RT Java Mapping      | ___ Load Balancing                   |
| ___ Schedulers           | ___ Reliable Multicast               |
| ___ QoS for UML and XML  | ___ Data Pub/Sub                     |
| ___ RT Fault Tolerance   | <u>2</u> Online Updates              |
| ___ Smart Transducers IF | <u>a few</u> Reliable Multicast      |
| <u>1</u> RT Transactions | ___ UML for Fault Tolerance          |
| ___ Temporal Reasoning   | <u>a few</u> Safety Critical Support |
| <u>a lot</u> RT Security |                                      |

*We took a straw pole on a limited set of the new work items (in red)  
There was not enough time to poll on the rest.*

*End of Presentation*

