

A Protocol for Representing Individual Hardware Devices as Objects on a CORBA Network

Presented by: William Nagel

Original Research by:

William Nagel

Patrick Beasley

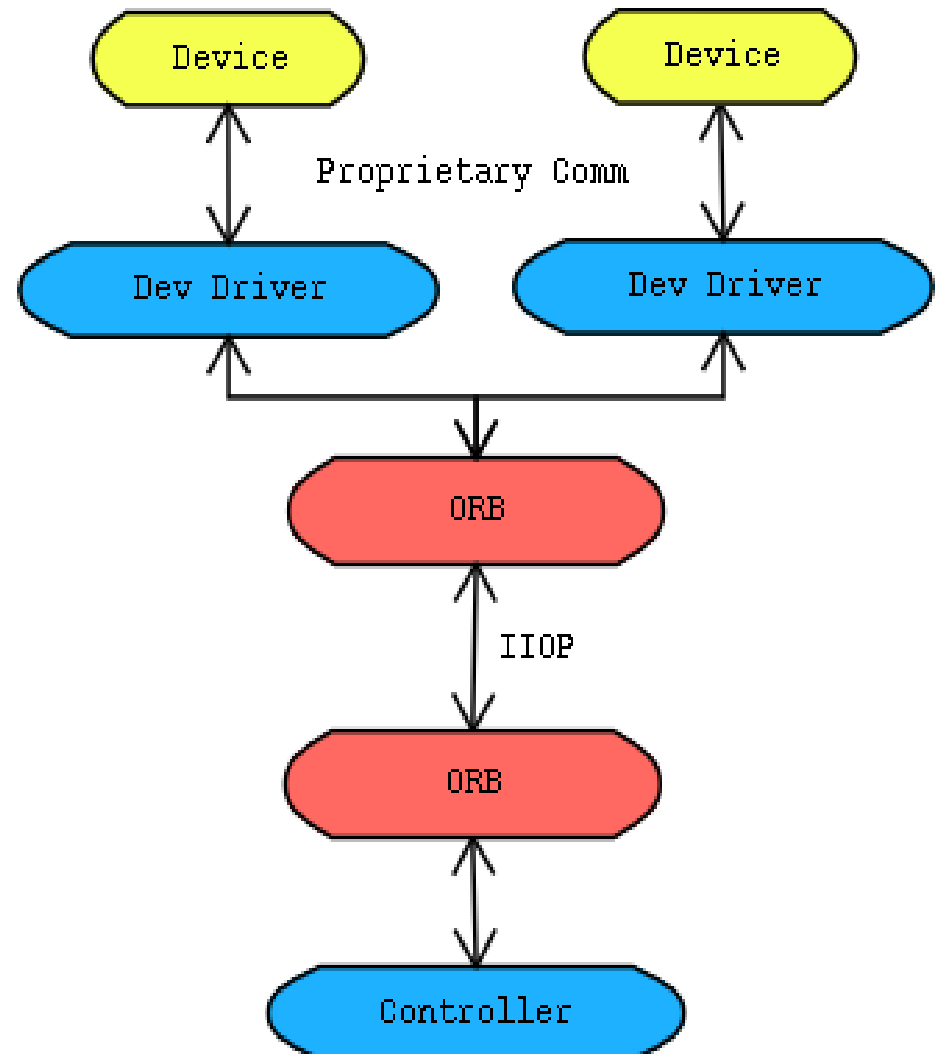
Nicholas Anderson

Overview

- CORBA Device Networks & Pitfalls
- Make the device the object
- Lightweight communication
- Device Object ORB
- Applications & Future Work

CORBA Device Networks

- Device networks can consist of many small devices
- Making these devices work together seems like a perfect application for CORBA

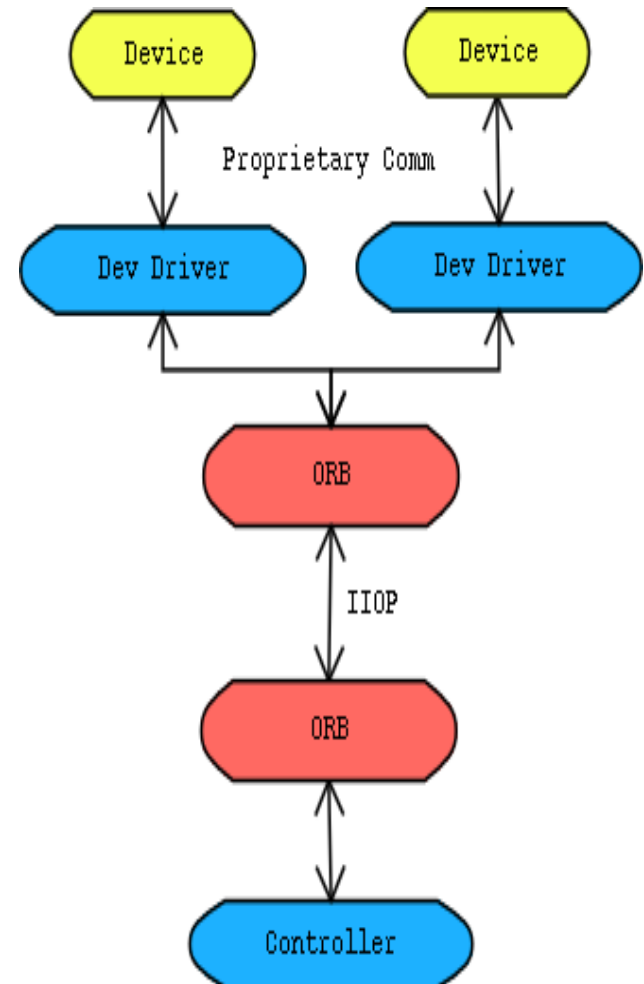


What do we mean by a device

- Very specialized units, built to perform a specific task (or small number of tasks)
- Usually easily represented as an object
- Often controlled by embedded microcontrollers with little or no extra processing/storage capacity
 - Example: the Microchip PICmicro[®] **microcontroller has a max ~128K memory for program and data**

Pitfalls

- Objects are software oriented
 - Implementations generally exist in processes collocated with the ORB
 - Communication to non-local objects usually occurs between ORBS, via GIOP based comm
- A non-local hardware device requires software driver to communicate over proprietary protocol

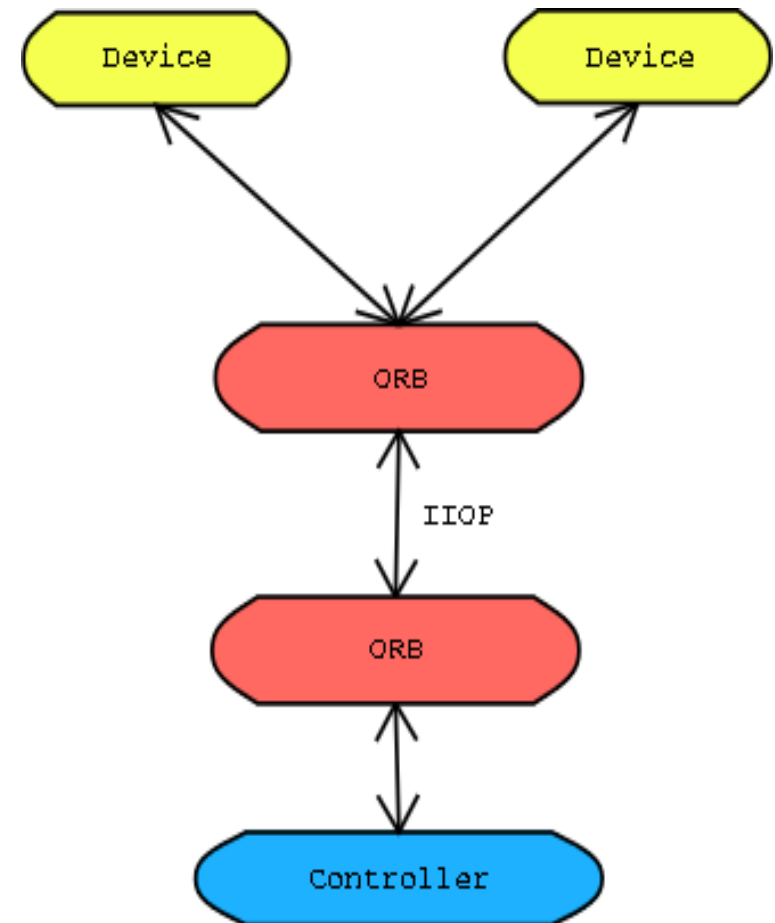


Pitfalls (cont.)

- Placing the ORB locally requires too much overhead for many low power devices/low bandwidth communications
 - With <128K memory, actually device controller software can edge out space for ORB
 - Inter-ORB communications via GIOP can be (relatively) expensive, with a small request easily 40-50 bytes (+ transport overhead)

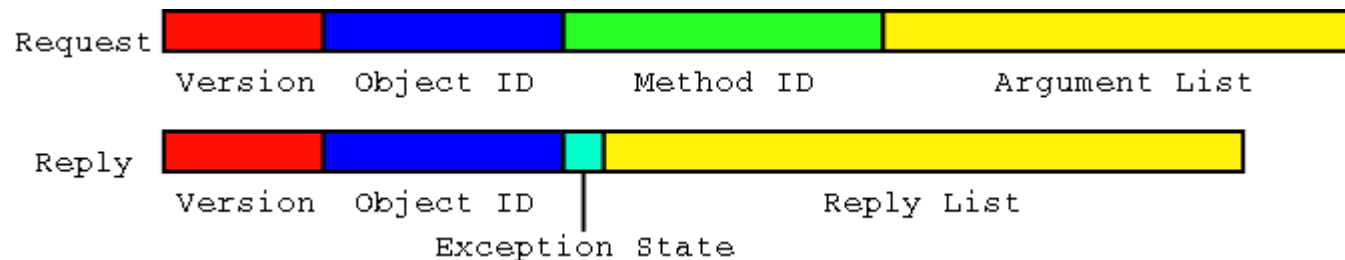
Devices as Objects

- Make the device be the object implementation
- Have devices talk directly to the ORB
- Eliminates need for proprietary drivers/bridges
- Increases flexibility for developers

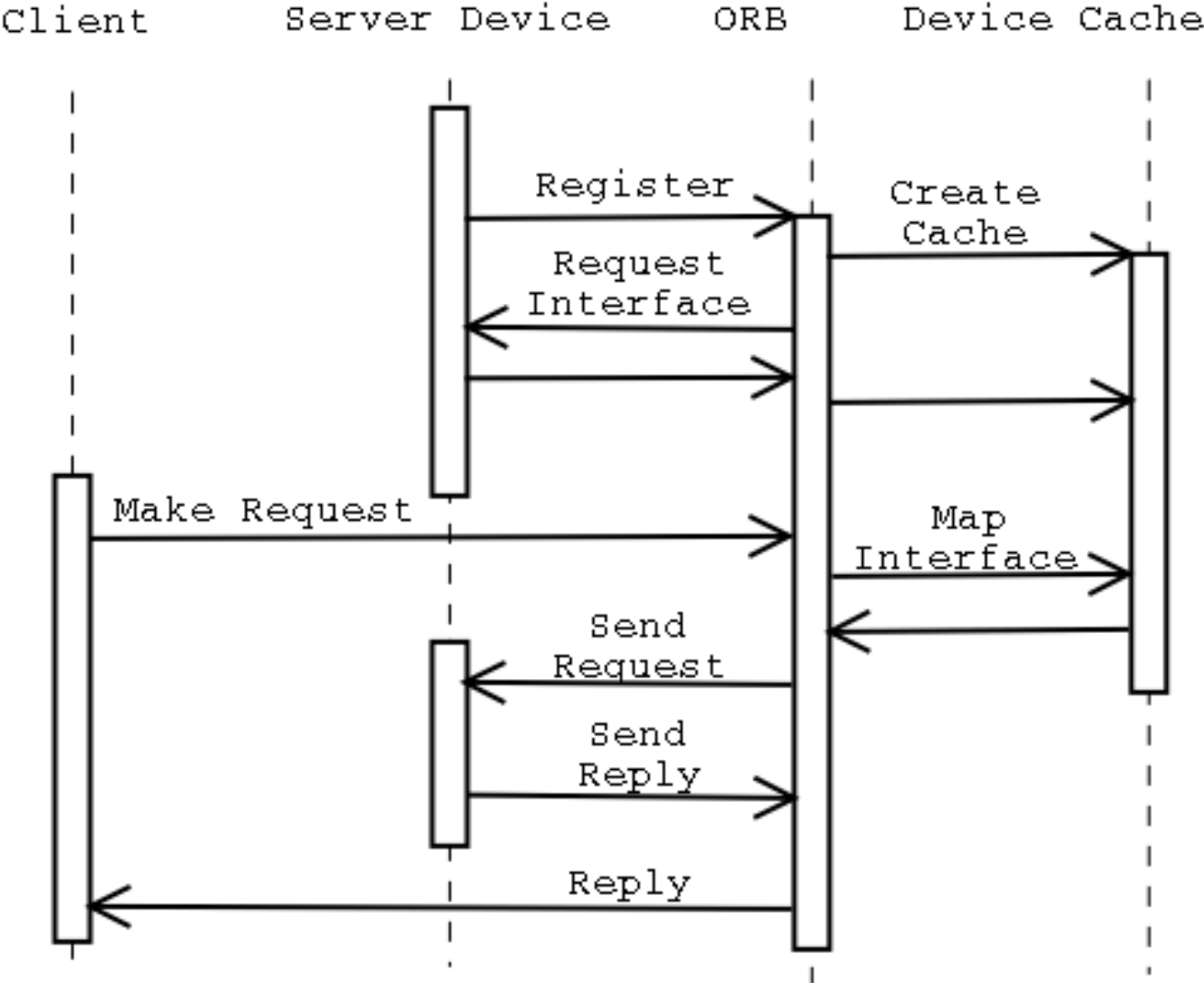


Lightweight Communication

- Use early communication of overhead
 - Interfaces & Object Ids
- Map high overhead data to smaller integer identifiers to reduce communications
 - Request as little as 11-12 bytes (+ transport)
- Allow fallback to high-overhead GIOP



Communications Sequence



The Device Object ORB

- Resides on an embedded device with sufficient processing power
- POA stores cached device information
- Supports external object registration, for devices without the capacity for self-registration
- Supports connectionless oneway method invocation for devices on simplex comm

Advantages

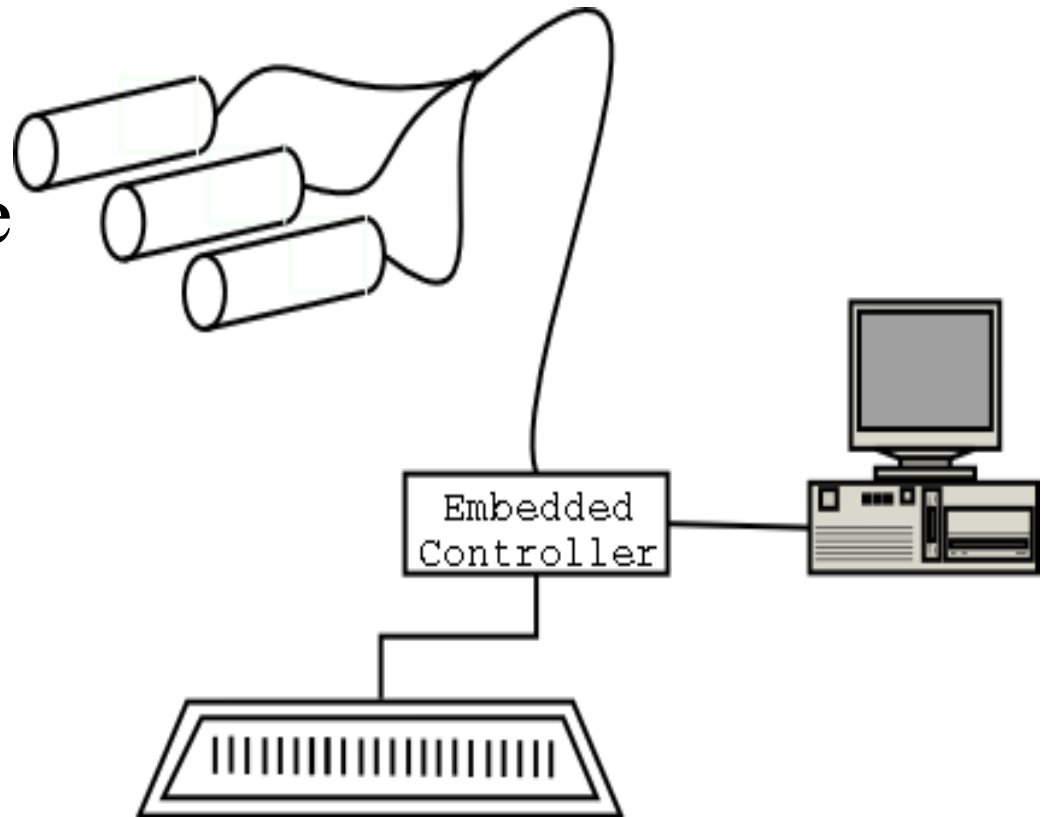
- Reduced development costs
 - OO development of device networks
 - Interface compatibility and reuse
- Reduced production costs
 - Smaller, cheaper devices
 - Need for fewer embedded systems running expensive ORBs
- Reduced communication overhead
 - Up to 400% less data directly to device

Disadvantages

- Requires specialized ORB
 - Can be made to work with a wrapping server to an ORB with another language binding
- Smaller comm protocol can lose some features/flexibility of full CORBA
- Centralized ORB can provide a single point of failure for multiple devices

Applications

- Theatrical lighting system
- Vehicle control system
- Manufacturing



Future Work

- Further refinement of comm protocols
- Build ORB into chip on device
- Lightweight inter-ORB communications