CORBA Application in Real Time Embedded Systems

By: Yanfeng Gong
Outline

• CORBA Technology
• Contemporary Embedded Systems
• Minimum CORBA
• Real-time CORBA
• Conclusion
Overview of CORBA

- Common Object Request Broker Architecture (CORBA) specified by OMG.
- CORBA is a standard and NOT a product, defines interface NOT implantation.
- CORBA simplifies development of distributed applications by automating/encapsulation:
  - E.g. Object location, connection, object/server activity.
- CORBA shields application from heterogeneous platform dependency:
  - E.g. Programming languages, operating systems, network protocols, hardware.
Architecture of CORBA

- Client
  - DII
  - IDL Stubs
  - OBJ REF
  - ORB Interface
  - GIOP/IIOP

- Object (Servant)
  - IDL Skeleton
  - DSI

- ORB CORE

Operation():
in args
out args + return value
Servant (known as object) - CORBA programming entity that has:

- identity
- interface
- implementation

Client - invokes an operation on an object implementation. Accessing the services of a remote object should be transparent to the caller. Ideally, as simple as obj->op(args).
Architecture (3/6)

Object Request Broker (ORB) - object bus over which objects transparently interact with other objects located locally or remotely.

General Inter-ORB Protocol (GIOP) - specifies transfer syntax and a set of standard message formats.

Internet Inter-ORB Protocol (IIOP) – GIOP implementation of GIOP over TCP/IP.
**Architecture (4/6)**

IDL stubs and skeletons – interfaces between ORB and Client/Servant

---

**Stubs** create and issue requests on behalf of a client

**Skeletons** deliver requests to the CORBA object implementation.

Invocation through stubs is called static invocation
Architecture (5/6)

**Dynamic Invocation Interface (DII)** - allows the client to dynamically issue requests to objects without the compile-time knowledge of object implantation.

**Dynamic Skeleton Interface (DSI)** - provides dynamic dispatch to objects: allows an ORB to deliver requests to an object without the compile-time knowledge of object implantation.

Invocation through DII is called dynamic invocation.
Object Adaptor - demultiplexes requests to the correct servant and registers classes it supports and their run time instances
CORBA interfaces are defined in Interface Definition Language (IDL).

IDL is a declarative language independent to any other programming language.
Traditional Embedded System
Distributed Real-time Embedded System

Large scale switch system

Industry process control

Avionics

Missile defense system
Characteristics of DREs

- Network centric
- Resource constrained
- Real-time critical
Motivation for CORBA

- Hardwires keep getting smaller, faster, and cheaper
- Systems keep getting bigger, more heterogeneous, and more expensive
- Success of CORBA in enterprise computing arena
- Leveraging the highly developed, technically robust, and widely compatible technology
Requirements/limitations of CORBA application in DREs

- **Requirements**
  - Location transparency
  - Performance predictability
  - Small memory footprint

- **Limitations**
  - Lack of real-time programming feature
  - Lack of performance optimization
Minimum CORBA

- Completely compatible with full-size CORBA applications
- The features of dynamic interface definition and dynamic invocation are removed by minimumCORBA
- Example, ORBExpress CORBA has a static memory footprint of less than 100k on some platform
Real-time CORBA

• Add priorities to CORBA applications

• Coordinate with RTOS to manage the processor, memory, and network resources
CORBA priorities are consistently mapped down to the priorities of underlying RTOS task/threads.
Memory Management

- Allows memory resources to be used in a predictable fashion through the control over thread pools.
- Threads that are used by a CORBA sub-system and the amount of memory each thread takes up to be controlled.
Network Resource Management

• The application can select and/or configure the available network protocols
• Client can obtain a private transport connection to a server
• Client can communicate with a server via multiple connections
Conclusions

• Successful cases have proven that CORBA improves DRE software development quality, productivity, and adaptability

