CSC9006 Real-Time Systems

Pattern-Based C++ Network Programming
Part 1: Revisiting CORBA and TAO

Chao Wang

Networked Cyber-Physical Systems Laboratory
Department of Computer Science and Information Engineering
National Taiwan Normal University

June 17th, 2021

NATIONAL TAIWAN NORMAL UNIVERSITY

Agenda

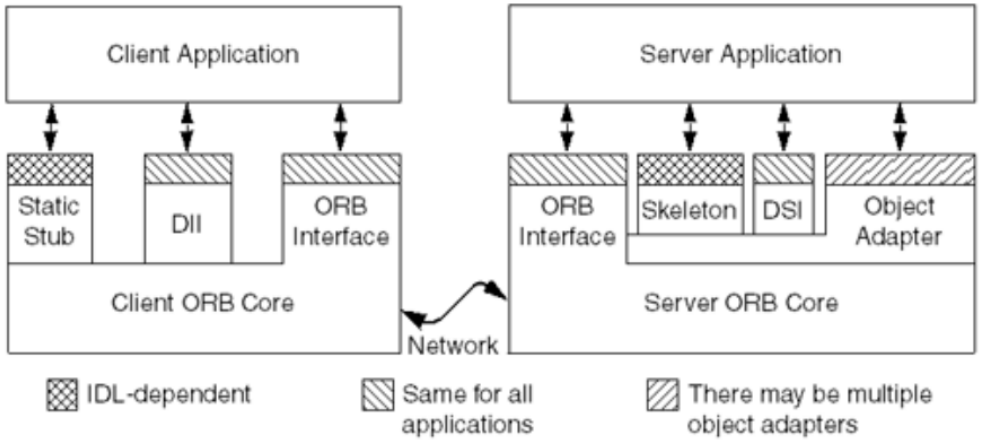
Links to the video recording for this presentation and demo:

- click here for the first half (download for a high resolution version)
- click here for the second half (download for a high resolution version)

1. Revisiting CORBA basics
   - Terminology and Basic Concepts
   - Naming Service
   - Event Service
2. Revisiting TAO and TAO's Real-Time Event Service
3. Demonstrating TAO and TAO's Real-Time Event Service
4. References

CORBA Terminology

- CORBA: Common Object Request Broker Architecture
- A server application contains several CORBA objects, each of which provides some specified operations.
- A client application at run-time may request some operations on a CORBA object.



(Figure 2.3 in [1])

CORBA Terminology (cont.)

- An object reference is a handle for a client to direct its request to a CORBA object.
  - Each object reference refers to an unique CORBA object
  - A CORBA object may have several object references
- A servant is a program implementation of one or more CORBA objects. A servant is part of the server application.
- The following terms will be covered in the next couple slides
  - IDL (Interface Definition Language), Stub, and Skeleton
  - DII (Dynamic Invocation Interface) and DSI (Dynamic Skeleton Interface)
  - Object Adapter and POA (Portable Object Adapter)
  - IOR (Interoperable Object Reference)

Interface Definition Language (IDL)

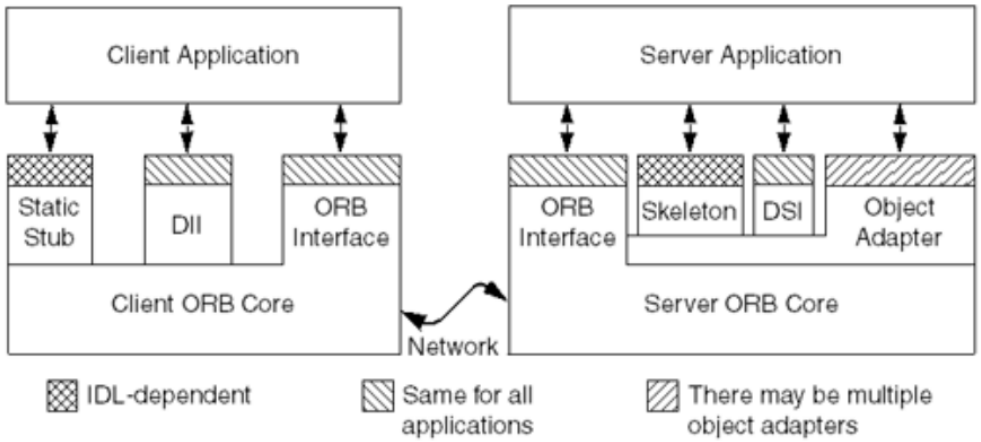
- Goal: enable *interoperability* among applications implemented in different programming languages.
- Approach:
  - First, define object interfaces and their operations in a way that is *independent* of any particular programming language.
  - Then, *compile* the IDL definitions into the code of the specified programming language (C++, for example).

Example

```
interface Time {
  long current_time (); // First operation
  long count_requests (); // Second operation
  oneway void shutdown (); // Third operation
};
```

Static Invocation vs. Dynamic Invocation

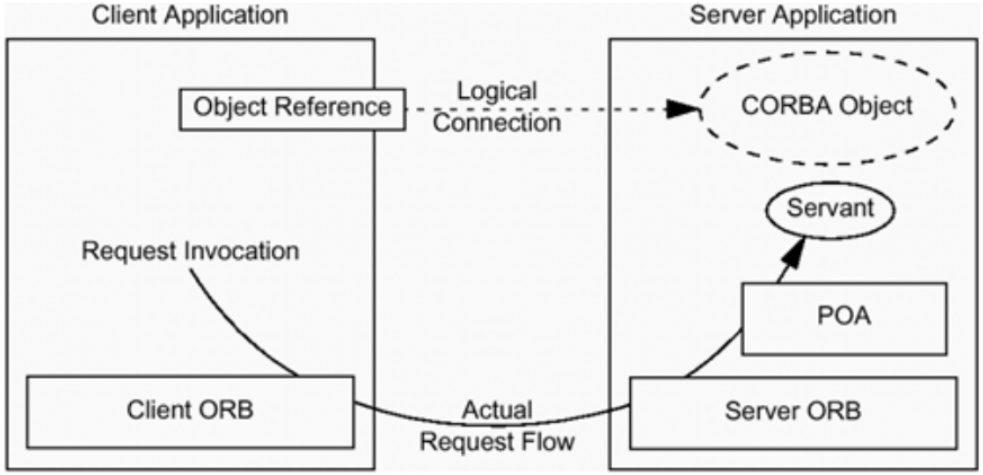
- Static invocation and dispatch
  - The IDL compiler translates the IDL definitions into client-side functions (i.e., stub) and server-side functions (i.e., skeleton), which will be complied into applications.
- Dynamic invocation and dispatch
  - Both the construction and dispatch of CORBA requests are made at run-time. At client-side it is made possible through DII; at server-side, DSI. In this course we do not use dynamic invocation and dispatch.



(Figure 2.3 in [1])

Object Adapter and POA

- Adapter is a classic design pattern[1].
- The object adapter glues the ORB and the servant, so that the ORB may invoke request on an object without knowing the object's true interface.
- POA stands for the portable object adapter and it replaced the obsolete basic object adapter; see Section 2.4.5 and Chapter 11 in [1] for more detail.



(Figure 11.2 in [1])

[1] See the description from Wikipedia (link)

Interoperable Object Reference (IOR)

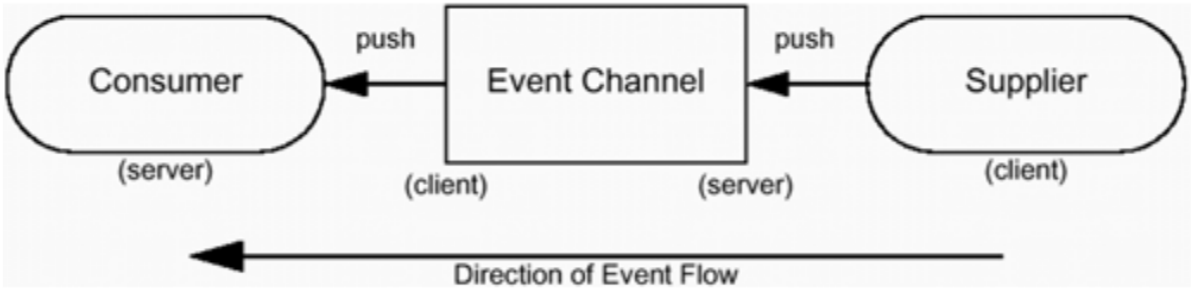
- It is essential for remote ORB applications to communicate with each other, and for different ORB vendor to operate with each other. The General Inter-ORB Protocol (GIOP) is an abstract protocol for this purpose.
- The Internet Inter-ORB Protocol (IIOP) specifies how GIOP is implemented over TCP/IP.
- Interoperable Object Reference (IOR) is the standard object reference format, which contains information for ORBs to establish communications between clients and target objects.
- An IOR for IIOP contains a host name, a TCP/IP port number, and an object key that identifies the target object.

Naming Service

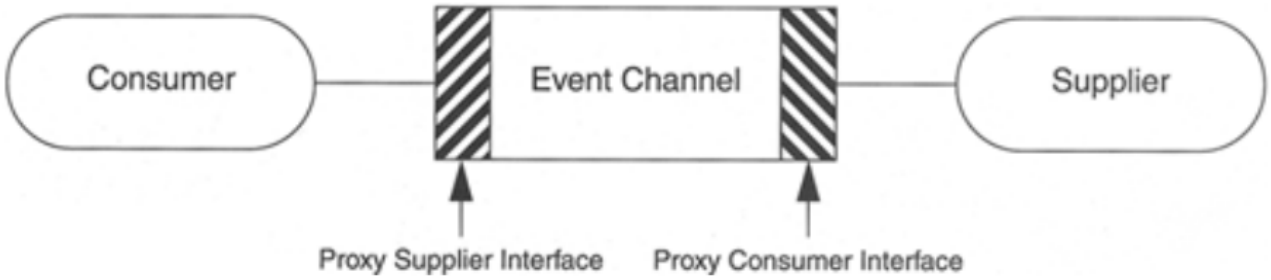
- Motivation: Use a meaningful name instead of the opaque object reference string for object request invocation.
- Approach: Similar to how the Internet Domain Name Service (DNS) provides a mapping from domain names to IP addresses, the naming service provides a mapping from names to object references.
- Applications can use the naming service to lookup the object reference of a certain name.

Event Service (a.k.a. Event Channel)

- Motivation: as described in [3]
- Approach: Event channel acts as a middleman, taking requests from suppliers and, accordingly, issuing requests to consumers.



(Figure 20.1 in [1])



(Figure 20.8 in [1])

TAO, The ACE ORB

- TAO[2] is a mature, widely used C++ implementation of CORBA. It is built upon ACE[3], the adaptive communication environment.
- TAO's IDL compiler creates C++ stubs and skeletons from an IDL file.

Example output of TAO's IDL compiler

(assuming Messenger.idl)

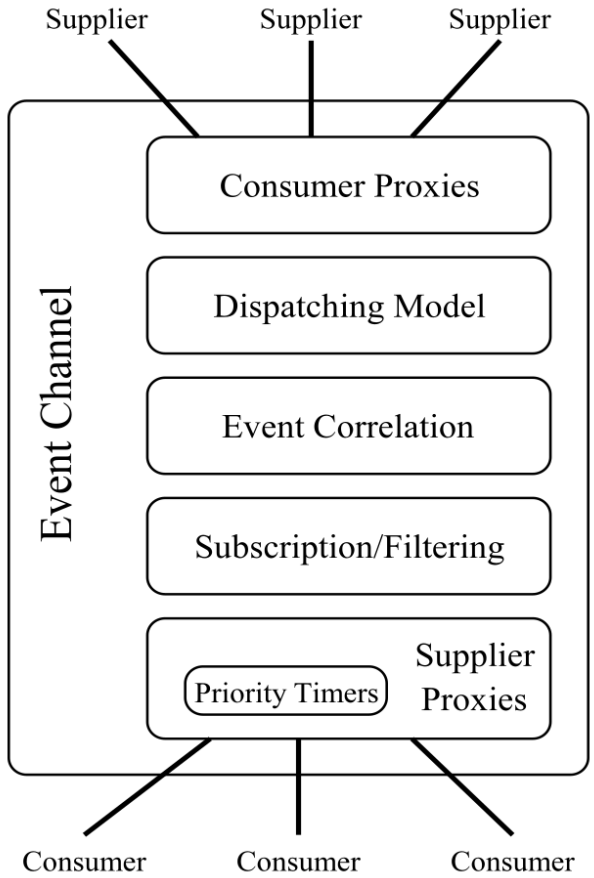
File Name	Contents
`MessengerC.h`	Stub class definitions.
`MessengerC.inl`	Inline stub member function definitions.
`MessengerC.cpp`	Stub member function definitions.
`MessengerS.h`	Skeleton class definitions.
`MessengerS.inl`	Inline skeleton member function definitions.
`MessengerS.cpp`	Skeleton member function definitions.

(Table 4-1 in [2])

[2] http://www.dre.vanderbilt.edu/∼schmidt/TAO.html
[3] http://www.dre.vanderbilt.edu/∼schmidt/ACE.html

TAO's Real-Time Event Service

- See the figure on the right. Notice how the names for proxies differ from those in [3]. The version here aligns with the class names in the code.



(Fig 24-1 in [3])

Demonstrating Some Usage of TAO

- Example folder: `$TAO_ROOT/examples/Simple`
- See the demo in the video clip for this presentation (click the link to the first-half of the recording on page 2)

Demonstrating TAO Real-Time Event Service

- Example folder: `$TAO_ROOT/orbsvcs/examples/RtEC/Simple`
- See the demo in the video clip for this presentation (click the link to the second-half of the recording on page 2)

References

[1] Henning, Michi, and Vinoski Steve. Advanced CORBA programming with C++. Wesley Longman, 1999.

[2] The TAO Developer's Guide, from Object Computing, Inc. https://objectcomputing.com/products/tao/tao-developers-guide

[3] T. Harrison, D. Levine, and D.C. Schmidt, The Design and Performance of a Real-time CORBA Event Service, ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications, October 1997.