BITS Pilani
Pilani | Dubai | Goa | Hyderabad

# CORBA Addressing and Naming

**Srikanth Gunturu**

Guest Faculty
BITS, WILP

# In this segment

## CORBA Addressing

- Addressing Overview
- Interoperable Object Reference (IOR) format
  - The original CORBA Naming Service
- CORBA Interoperable Naming Services
  - Corbaloc
  - Corbaname

# CORBA Addressing

## Overview

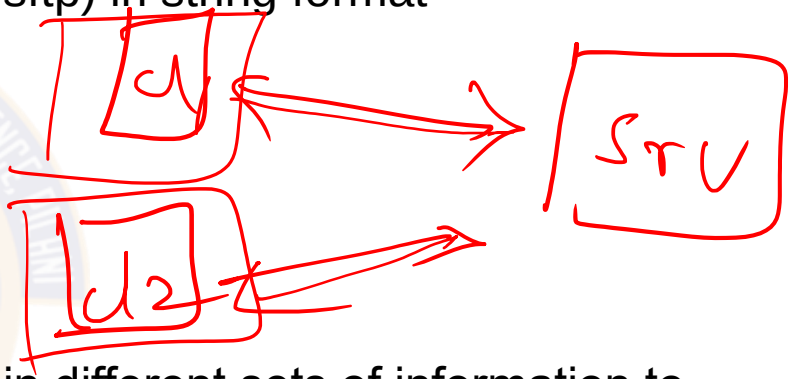
- Major address formats in CORBA
  - Interoperable Object Reference (IOR) format
  - Uniform Resource Identifier (URI) formats from CORBA Interoperable naming services
    - Corbaloc
    - Corbaname
- Address is obtained using “string_to_object” function on client object, which takes address string and returns object reference
- Reference is then narrowed to proxy object using Helper methods
- Other address formats (optional) are:
  - file://filename (name of the file that contains address)
  - http://url (URL that contains address)
  - ftp://filename (location of file that contains address)

```
org.omg.CORBA.Object obj = orb.string_to_object(line);
echo myecho = echoHelper.narrow(obj);
```

# CORBA Addressing

## Interoperable Object Reference (IOR)

- Original addressing format started by CORBA (1991)
- IOR is created on server and shared with client (via email/sftp) in string format
- IOR contains:
  - IP address and port Number
  - Object reference (for skeleton)
  - Unique ID for the Interface
  - Version of Internet Inter-Orb Protocol (IIOP)
- IOR provides “tagged profiles”, each of which would contain different sets of information to obtain address

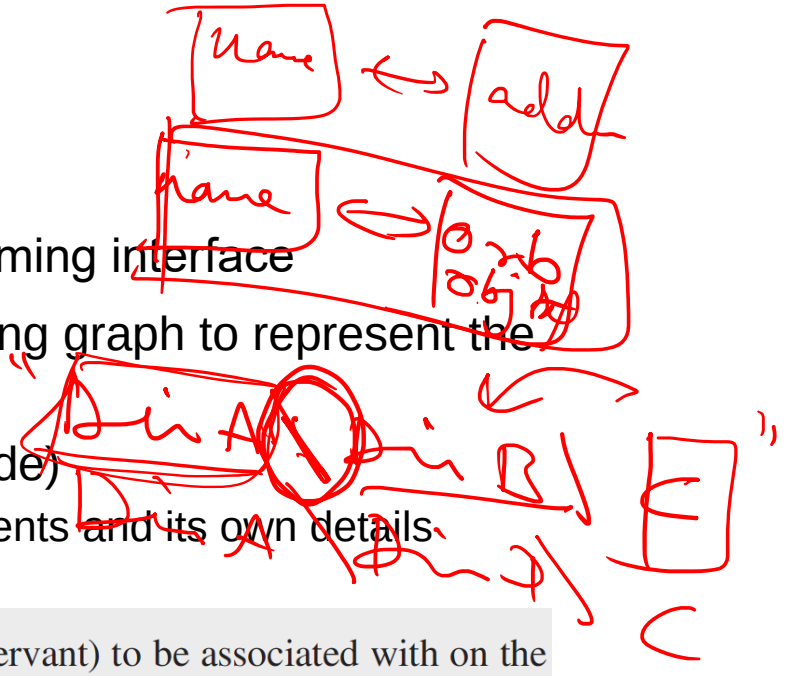



```
FileWriter thefile = new FileWriter("echo.ior");
BufferedWriter bufferedWriter = new BufferedWriter(thefile);
        org.omg.CORBA.Object theref = mypoa.id_to_reference(servant_id);
        String mystr = orb.object_to_string(theref);
bufferedWriter.write(mystr);
bufferedWriter.close();
```

# CORBA Addressing

## The Original CORBA naming service

- Maps domain names to object references (analogous to DNS)
- Naming service runs separately, included by ORB vendors
- Server "binds" name to CORBA object reference using CORBA naming interface
- Naming works hierarchically like directory structure, but uses naming graph to represent the structure (due to CORBA's naming limitation with '/')
- Naming service interface is represented as IDL (in ORB source code)
  - Each node is represented as a struct defined in IDL containing its parents and its own details



```
typedef String Istring;
struct NameComponent {
    Istring id;
    Istring kind;
};
```

```
//Create a name for the CORBA object (and its servant) to be associated with on the
        Naming Service
CosNaming::Name name;
name.length (1);
name[0].id = CORBA::string_dup ("echoservant");
name[0].kind = CORBA::string_dup (""); // make sure the "kind" field is empty

// Rebind the name to its reference on the Naming Service.
// We do a "rebind" in case you did this before and didn't restart the Naming Service
namingcontext->rebind (name, echo_ref);
```

# CORBA Addressing

## CORBA Interoperable naming service

- Introduced in CORBA 2.4 spec (2000)
- Contains two addressing formats: corbaloc and corbaname
- Corbaloc
  - corbaloc:iiop: < host IP address>:< port number>/object_key (default port 2809)
  - corbaloc:rir: < host IP address>:< port number>/object_key (resolves initial references)
  - corbaloc:iiop:1.2@[ipv6 address]:<port number>/object_key
  - Ex: corbaloc:iiop:localhost:4321/theObjectKey
- Corbaname
  - corbaname: <corbaloc format including object key as before> #name string
  - Ex: corbaname::theServer.edu:1234#echo
  - Can be used to find Naming service with "rir"

```
org.omg.CORBA.Object naming_service_ref =
    orb.string_to_object("corbaloc:iiop:1.2@localhost:1163/NameService");

NamingContextExt namingcontext= NamingContextExtHelper.narrow
    (naming_service_ref);
```

```
org.omg.CORBA.Object proxy = orb.string_to_object("corbaname::localhost:1163#echo");
echo myecho = echoHelper.narrow(proxy);
```

# Thank You!

In our next session:

CORBA-RMI Use case