[simula . research laboratory]

Work group meeting no. 2
-Call back and CORBA naming service

INF5040 (Distributed systems)

Name: Sten L. Amundsen
Date: 7 September 2004

e-mail: stena@simula.no

[simula . research laboratory]

Agenda

- Call back interface on client
- Object reference using CORBA naming service

simula . research laboratory

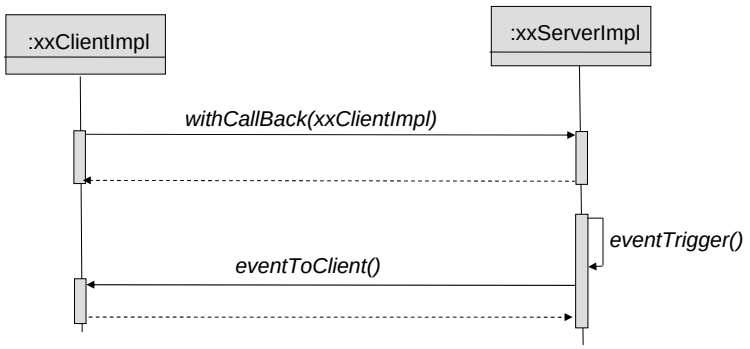
Call back interface

- To achieve asynchronous communication one can use call back interface.
- One sends over the object reference to the server side as a parameter in the method invocation.
- Server do remote invocation on the client object using this object reference.

simula . research laboratory

Call back interface –client/server

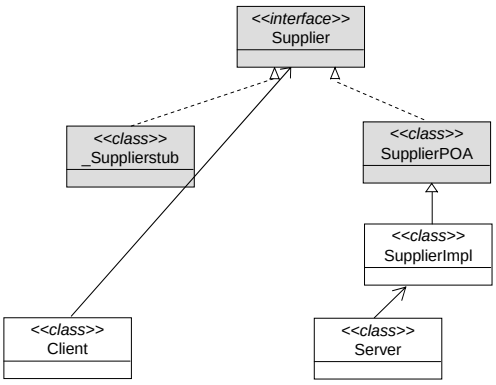
- Message sequence diagram illustrates the concept with call back between client and server.



simula . research laboratory

Call back interface –relationship (1)

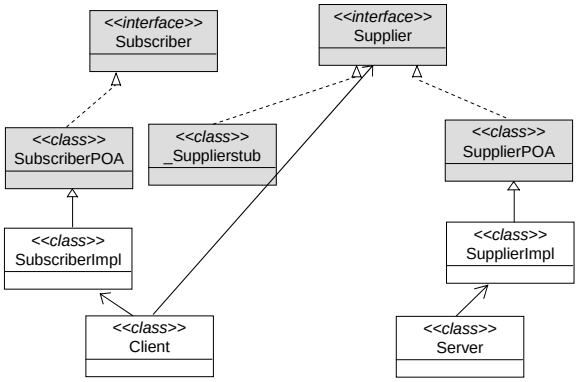
- First a normal relationship between client and servant.



simula . research laboratory

Call back interface –relationship (2)

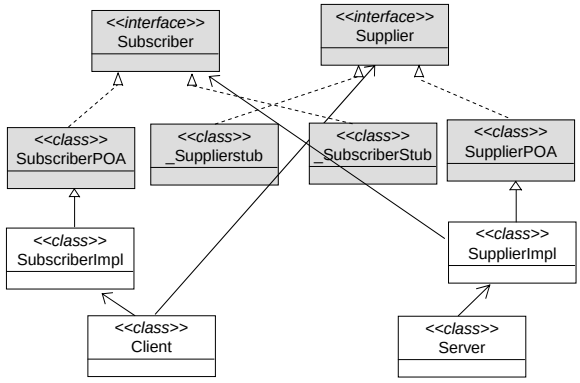
- Client initiate a servant and make it know to the POA and ORB.



simula . research laboratory

Call back interface –relationship (3)

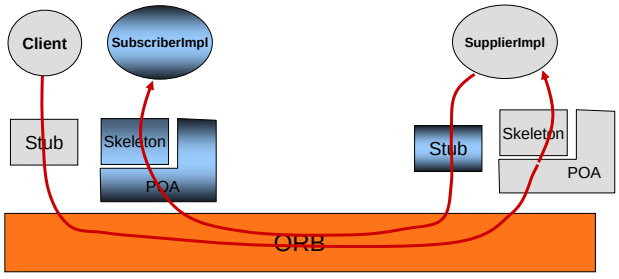
- Reference to the CORBA interface is transferred.
- When this is called, the communication goes through the stub.



simula . research laboratory

Call back interface –simplified view

- Omitting the ORB interface.



simula . research laboratory

CORBA naming service

- CORBA has services to reduce the programming complexity.
- **Naming service** is easy to use and very useful.

- Other services are:
  - Collection service
  - Concurrency service
  - Enhanced view of time
  - Event service
  - Externalization service
  - Licensing service
  - Life cycle service
  - Management of event domains
  - Notification service
  - Persistent state service
  - Property service
  - Query service
  - Relationship service
  - Security service
  - Telecoms log service
  - Time service
  - Trading object service
  - Transaction service

simula . research laboratory

CORBA naming service -overview

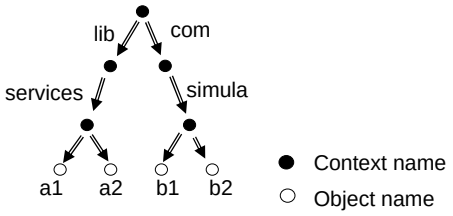
- Object references cumbersome to fetch from a file.
  - Name service used by clients to get service/resource/object using a name.
- Examples of name services:
  - DNS for obtaining IP address from a name (URL, e-mail, layer 4 sessions).
  - CORBA naming service for obtaining reference to CORBA objects
- Name service different from directory services.
  - Name service is limited to search after a name.
  - Directory service can in addition search using attributes describing the service/resource/object or even data element.
  - Examples of directory services are:
    - X.500
    - LDAP
    - CORBA Trader

simula . research laboratory

CORBA naming service –naming graph

- Hierarchy is used to structure the name space.
  - In DNS use domains, e.g. www.ifi.uio.no/<path>/<filename>
  - In JDNI use packages, e.g com.simula.QuA.<component name>
  - In CORBA use naming context, e.g /com/simula/<object name

Sum of all name contexts and bounded object name is called a Naming graph.



simula . research laboratory

CORBA naming service – Reference

- CORBA Object references (IOR –interoperable object reference) are stored in the naming server, together with:
  - Value attribute: the name given to this reference (reference can be a context or an object).
  - Kind attribute: Information about the role/semantics of the CORBA object.
- The Naming Service interface for creating context and object references is expecting the NameComponent object.

Constructor in the Java implementation.

```
NameComponent(String __id, String __kind)
```

simula . research laboratory

CORBA naming service – IDL interface

- Interfaces (operations and parameter) to the CORBA naming server is specified by OMG in an IDL.
- Example of operations:

```
bind_new_context(NameComponent[] n)
        // Operation creates a new context and binds it to supplied argument

bind(NameComponent[] n, Object obj)
      //Creates a binding of a name and an object in the naming context.

resolve(NameComponent[] n)
      //The operation retrieves object bound to a name in a given context
```

simula . research laboratory

Example – callback & nameservice - IDL file

- IDL file must define both interfaces.
- In parameter to operation availableSubscription is the call back interface (to the CORBA object)

```
module callback {

   // The name of a CORBA object.
   interface Subscriber {
        //signature for plain method/operation invocation
        void magazineSubscription(in long id, in string magazine);
   };//end interface

   // The name of a CORBA object
   interface Supplier {
        //signature with callback interface as input
        void availableSubscription(in Subscriber sub);
   };//end interface
}; //end package
```

simula . research laboratory

Example – callback & nameservice - Invoke remote method

- Call back, requires the servant to be ready.
- Method invocation as normal.

```
public void prepareCallBack() {
        SubscriberImpl subscriberImpl = new SubscriberImpl();
        subscriber = subscriberImpl._this(orb);
```

```
// Since corbaObj is to remote machine, the Helper class
// returns reference to the stub (_SupplierStub).
 Supplier supplier = SupplierHelper.narrow(corbaObj);
 supplier.availableSubscription(subscriber);
```

simula . research laboratory

Example – callback & nameservice - Use callback interface

- To receive and use call back interface easy to implement.

```
public void availableSubscription(Subscriber sub) {
        // Call operation on callback interface.
        sub.magazineSubscription(1, "IEEE Communication magazine");
}
```

simula . research laboratory

Example – callback & nameservice - Reference to Naming service

- Get reference to name service.
- Create name context in the context to name service.

```
org.omg.CORBA.Object obj =
        orb.resolve_initial_references("NameService");

NamingContext nc = NamingContextHelper.narrow(obj);
```

```
//Create context com
NameComponent[] nc1Name = new NameComponent[1];
nc1Name[0] = new NameComponent();
nc1Name[0].id = "com";
nc1Name[0].kind = "";
NamingContext nc1 = nc.bind_new_context(nc1Name);
//then create com.simula  and bind to com.
```

simula . research laboratory

Example – callback & nameservicee - Store object ref in naming service

- Bind object reference to a name for the context.
- Create a naming graph, with */com/simula/Supplier*

```
// Bind names with the Naming Service
NameComponent[] aName = new NameComponent[3];
aName[0] = new NameComponent();
aName[0].id = "com";
aName[0].kind = "";
aName[1] = new NameComponent();
aName[1].id = "simula";
aName[1].kind = "";
aName[2] = new NameComponent();
aName[2].id = "supplier";
aName[2].kind = "";
nc.bind(aName, supplier);
```

simula . research laboratory

Example – callback & nameservice - Retrive object ref from naming service

- Get reference to name service.
- Search for object reference for a context (/com/simula)
- State the name the object reference is bound to.

```
// Establish connection to ORBacus Name service,
// which is running as a stand-alone server.
org.omg.CORBA.Object obj =
    orb.resolve_initial_references("NameService");

NamingContext nc = NamingContextHelper.narrow(obj);

// Object name and attribues bound to /com/simula
// Create an array that gives path in the name graph.
NameComponent[] cName = new NameComponent[3];
cName[0] = new NameComponent();
cName[0].id = "com";
cName[0].kind = "";
cName[1] = new NameComponent();
cName[1].id = "simula";
cName[1].kind = "";
cName[2] = new NameComponent();
cName[2].id = "supplier";
cName[2].kind = "";
corbaObj = nc.resolve(cName);
```

simula . research laboratory

Example – callback & nameservice - Start ORBacus Name service

- The Name Server is a stand-alone server.
- Listens on a defined port.
  - Use command line option –OAport

```
java com.ooc.CosNaming.Server -OAport 7007
```

simula . research laboratory

Example – callback & nameservice - Start server with Name service

- Both client and server need to know the hostname, port and name of the CORBA Name service.
- Location (host and port) and name of the Naming Service, must be fed to the ORB when it is started.
  - Use the arguments in Main method to get details into application

```
java callback.Server
        -ORBInitRef NameService=corbaloc::simula-dhcp-081:7007/NameService
```

Hostname

Port

Name to CORBA service

simula . research laboratory

Example – callback & nameservice

- To understand the implementation, do the exercise.