

Slides for Chapter 20: CORBA Case Study



fourth edition
DISTRIBUTED SYSTEMS
CONCEPTS AND DESIGN
George Coulouris
Jean Dollimore
Tim Kindberg

From **Coulouris, Dollimore and Kindberg** **Distributed Systems:** **Concepts and Design**

Edition 4, © Addison-Wesley 2005

Edited and supplemented by Jonne Itkonen,
for TIE427 (2011)

CORBA

■ Common Object Request Broker Architecture

- Object Management Groupin ylläpitämä *kuvaus* hajautetun (komponentti- ja) oliojärjestelmän arkkitehtuurista
 - Jep, sittemmin UML siirtyi OMG:n kaitsittavaksi...
- Ensimmäinen versio vuonna 1991, ”kulta-aika” vuosituhatosen vaihteessa
- ”Ensin toteutus, sitten speksi...”
 - Toimi alussa, sittemmin ei
- Usein haukuttu raskaaksi ja vaikeaksi, varsinkin kilpailijoiden toimesta (kts. wikipedia)
 - Turhankin laajat speksit, mutta perusteet ovat erittäin helpot
 - C++ tekee kaikesta vaikeata muutenkin, ei vain CORBA-toteutuksista

CORBA

■ Koostumus

- IIOP – Internet Inter-ORB Protocol
 - | viestintä olioiden välillä TCP/IP:llä
 - | myös muut kuin Internet-versiot mahdollisia
 - | GIOP – General IOP
 - | käytössä myös ei-CORBA välikerroksissa
- rajapintakuvaukset
 - | IDL-kieli, operaatiot, tietotyypit, koostetyypit, ...
- IOR – Interoperable Object Reference (olioviite)
- pseudo-oliot
 - | ORB core – rajapinta ORB:iin
 - | olioadapterit – olioiden standardoitu liityntä CORBA-toteutukseen

THES427 – Hajautetut järjestelmät 2011
Jonne Itkonen

CORBA

■ Koostumus jatkuu...

- palvelut
 - | nimi-, oliovarasto-, tapahtuma-, aika-, ...
- kohdealuemallit (domain specifications)
- <http://www.omg.org/gettingstarted/corbafaq.htm>
 - | kuuluu kurssin luettavaan materiaaliin
- Lisätietoa:
 - | <http://www.corba.org/>
 - | <http://www.ciaranmchale.com/corba-explained-simply/index.html>

THES427 – Hajautetut järjestelmät 2011
Jonne Itkonen

CORBA

Toteutuksia

- Java SDK & RT sisältävät CORBA-toteutuksen
- omniORB – C++ ja Python sidokset, avoin
- JacORB – kattavampi Java-toteutus, avoin
- Tao & AceORB – realtime, avoin
- Mico, Orbit, Fnorb, IIOP.net
- Visibroker – kaupallinen
- <http://corba-directory.omg.org/>

Tulevaisuus?

- <http://c2.com/cgi/wiki?WhatsWrongWithCorba>
- "CORBA is dead... like Cobol..." – legacy code
- kritiikkiä tulee kilpailijoilta ja heiltä, jotka ovat viimeksi käyttäneet huonoja ORBeja tai versiota 1.0, kuten yleensä
- Speksi on laaja ja opettavainen, hyvässä ja huonossa...
- dynamic services, transactions, events & messaging, webservice bridge, component model, ...

THES427 – Hajautetut järjestelmät 2011
Jonne Itkonen

CORBA

- IDL:


```

module Foo {
  struct Bar {
    fields;
    ...
  };
  ...
  interface Baz: A,B {
    ...
    methods & attributes
    ...
  };
  ...
  typedef description-with-available-types new-typename;
  ...
  exception Voih { ... };
};

```
- IDL method:


```

[oneway] <return_type> <method_name>(parameter1, ..., parameterL)
[raises(exception1, ..., exceptionN)] [context(name1, ..., nameN)]

```
- Constructed types: sequence, string, array, record (struct), enumerated, union

THES427 – Hajautetut järjestelmät 2011
Jonne Itkonen

Figure 20.1
IDL interfaces Shape and ShapeList

```

struct Rectangle{                               1
    long width;
    long height;
    long x;
    long y;
};

struct GraphicalObject {                       2
    string type;
    Rectangle enclosing;
    boolean isFilled;
};

interface Shape {                               3
    long getVersion();
    GraphicalObject getAllState();             // returns state of the GraphicalObject
};

typedef sequence <Shape, 100> All;              4
interface ShapeList {                          5
    exception FullException{ };               6
    Shape newShape(in GraphicalObject g) raises (FullException); 7
    All allShapes();                           // returns sequence of remote object references 8
    long getVersion();
};

```

Instructor's Guide for Coulouris, Dollimore and Kindberg Distributed Systems: Concepts and Design Edn. 4
© Pearson Education 2005

CORBA esimerkki

<https://yousource.it.jyu.fi/ties427-distributed-systems/ties427-luontosimerkit/trees/master/corba/echo>

```

$ idlj -fall echo.idl
...
$ omniidl -bpython echo.idl
$ ls
Echo.class          EchoServer.class  echo_idl.py
Echo.java           EchoServer.java   echo_idl.pyc
EchoClient.class   EchoServerImpl.class  echo_srv.py
EchoClient.java    Example           example_echo_idl
EchoHelper.class  Example__POA      example_echo_clt.py
EchoHelper.java    _EchoStub.class   example_echo_coloc.py
EchoHolder.java    _EchoStub.java    example_echo_idl.py
EchoOperations.class  _GlobalIDL       example_echo_nsclt.py
EchoOperations.java  _GlobalIDL__POA  example_echo_nssrv.py
EchoPOA.class      echo.idl          example_echo_srv.py
EchoPOA.java       echo_clt.py
ji$

```

Esimerkki omniORBpy:stä,
yliviivatuista ei tarvitse välittää.

Instructor's Guide for Coulouris, Dollimore and Kindberg Distributed Systems: Concepts and Design Edn. 4
© Pearson Education 2005

CORBA esimerkki

```

ji$ java EchoServer &
IOR:
00000000000000d49444c3a4563686f3a312e3000000000000000100000000000008a00010
2000000000f3133302e3233342e3136312e31380000eaeaf000000000031afabcb00000000206b
d2dba40000000100000000000000100000008526f6f74504f410000000008000000010000000
0140000000000002000000010000002000000000000100010000000205010001000100200001
0109000000010001010000000026000000020002
(IOR:000...20002 yhtenä rimpuna, powerpoint hienostelee tuossa...)

ji$ java EchoClient IOR:00000000...00020002
echoString() called with message: kissa istuu
kissa istuu
ji$ python echo_clt.py IOR:000000000...002
Traceback (most recent call last):
  File "echo_clt.py", line 6, in <module>
    from omniORB import CORBA
ImportError: No module named omniORB
ji$ ... jahas, pitää korjata omniORBpy asennus...

```

THES427 – Hajautetut järjestelmät 2011
Jonne Itkonen

Figure 20.2
Java interfaces generated by *idlj* from CORBA interface *ShapeList*

```

public interface ShapeListOperations {
    Shape newShape(GraphicalObject g) throws ShapeListPackage.FullException;
    Shape[] allShapes();
    int getVersion();
}

public interface ShapeList extends ShapeListOperations, org.omg.CORBA.Object,
    org.omg.CORBA.portable.IDLEntity { }

```

Instructor's Guide for Coulouris, Dollimore and Kindberg Distributed Systems: Concepts and Design Edn. 4
© Pearson Education 2005

Figure 20.3
ShapeListServant class of the Java server program for CORBA interface *ShapeList*

```
import org.omg.CORBA.*;
import org.omg.PortableServer.POA;
class ShapeListServant extends ShapeListPOA {
    private POA theRootpoa;
    private Shape theList[];
    private int version;
    private static int n=0;
    public ShapeListServant(POA rootpoa){
        theRootpoa = rootpoa;
        // initialize the other instance variables
    }
}
// continued on the next slide
```

Instructor's Guide for Coulouris, Dollimore and Kindberg Distributed Systems: Concepts and Design Edn. 4
 © Pearson Education 2005

Figure 20.3 continued

```
public Shape newShape(GraphicalObject g)
    throws ShapeListPackage.FullException {
    version++;
    Shape s = null;
    ShapeServant shapeRef = new ShapeServant( g, version);
    try {
        org.omg.CORBA.Object ref =
            theRoopoa.servant_to_reference(shapeRef);
        s = ShapeHelper.narrow(ref);
    } catch (Exception e) {}
    if(n >=100) throw new ShapeListPackage.FullException();
    theList[n++] = s;
    return s;
}
public Shape[] allShapes() { ... }
public int getVersion() { ... }
}
```

Jos palautettava lista on suuri,
 palauta ennemmin *iteraattori*.

Instructor's Guide for Coulouris, Dollimore and Kindberg Distributed Systems: Concepts and Design Edn. 4
 © Pearson Education 2005

Figure 20.4 Java class *ShapeListServer*

```

import org.omg.CosNaming.*; import org.omg.CosNaming.NamingContextPackage.*;
import org.omg.CORBA.*; import org.omg.PortableServer.*;
public class ShapeListServer {
    public static void main(String args[]) {
        try{
            ORB orb = ORB.init(args, null);
            POA rootpoa = POAHelper.narrow(orb.resolve_initial_references("RootPOA"));
            rootpoa.the_POAManager().activate();
            ShapeListServant SLSRef = new ShapeListServant(rootpoa);
            org.omg.CORBA.Object ref = rootpoa.servant_to_reference(SLSRef);
            ShapeList SLRef = ShapeListHelper.narrow(ref);
            org.omg.CORBA.Object objRef =orb.resolve_initial_references("NameService");
            NamingContext ncRef = NamingContextHelper.narrow(objRef);
            NameComponent nc = new NameComponent("ShapeList", "");
            NameComponent path[] = {nc};
            ncRef.rebind(path, SLRef);
            orb.run();
        } catch (Exception e) {
        }
    }
}

```

Instructor's Guide for Coulouris, Dollimore and Kindberg Distributed Systems: Concepts and Design Edn. 4
© Pearson Education 2005

Figure 20.5
Java client program for CORBA interfaces *Shape* and *ShapeList*

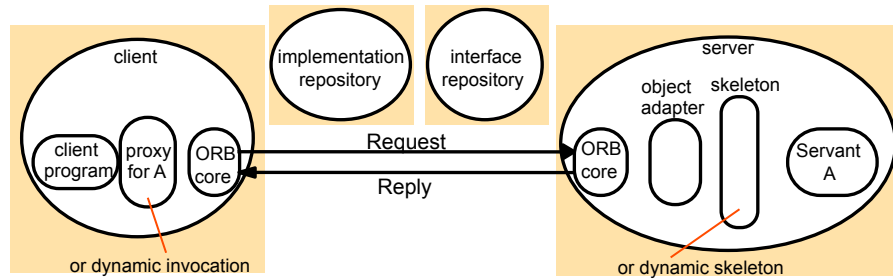
```

import org.omg.CosNaming.*;
import org.omg.CosNaming.NamingContextPackage.*;
import org.omg.CORBA.*;
public class ShapeListClient{
    public static void main(String args[]) {
        try{
            ORB orb = ORB.init(args, null);
            org.omg.CORBA.Object objRef =
                orb.resolve_initial_references("NameService");
            NamingContext ncRef = NamingContextHelper.narrow(objRef);
            NameComponent nc = new NameComponent("ShapeList", "");
            NameComponent path [] = { nc };
            ShapeList shapeListRef =
                ShapeListHelper.narrow(ncRef.resolve(path));
            Shape[] sList = shapeListRef.allShapes();
            GraphicalObject g = sList[0].getAllState();
        } catch(org.omg.CORBA.SystemException e) { ... }
    }
}

```

Instructor's Guide for Coulouris, Dollimore and Kindberg Distributed Systems: Concepts and Design Edn. 4
© Pearson Education 2005

Figure 20.6
The main components of the CORBA architecture



Instructor's Guide for Coulouris, Dollimore and Kindberg Distributed Systems: Concepts and Design Edn. 4
© Pearson Education 2005

Figure 20.7
IDL module Whiteboard

```

module Whiteboard {
    struct Rectangle {
        ... };
    struct GraphicalObject {
        ... };
    interface Shape {
        ... };
    typedef sequence <Shape, 100> All;
    interface ShapeList {
        ... };
};

```

Instructor's Guide for Coulouris, Dollimore and Kindberg Distributed Systems: Concepts and Design Edn. 4
© Pearson Education 2005

Figure 20.8
IDL constructed types – 1

<i>Type</i>	<i>Examples</i>	<i>Use</i>
<i>sequence</i>	<i>typedef sequence <Shape, 100> All;</i> <i>typedef sequence <Shape> All</i> bounded and unbounded sequences of <i>Shapes</i>	Defines a type for a variable-length sequence of elements of a specified IDL type. An upper bound on the length may be specified.
<i>string</i>	<i>String name;</i> <i>typedef string<8> SmallString;</i> unbounded and bounded sequences of characters	Defines a sequences of characters, terminated by the null character. An upper bound on the length may be specified.
<i>array</i>	<i>typedef octet uniqueId[12];</i> <i>typedef GraphicalObject GO[10][8]</i>	Defines a type for a multi-dimensional fixed-length sequence of elements of a specified IDL type.

this figure continues on the next slide

Instructor's Guide for Coulouris, Dollimore and Kindberg Distributed Systems: Concepts and Design Edn. 4
© Pearson Education 2005

Figure 20.8
IDL constructed types – 2

<i>Type</i>	<i>Examples</i>	<i>Use</i>
<i>record</i>	<i>struct GraphicalObject {</i> <i>string type;</i> <i>Rectangle enclosing;</i> <i>boolean isFilled;</i> <i>};</i>	Defines a type for a record containing a group of related entities. <i>Structs</i> are passed by value in arguments and results.
<i>enumerated</i>	<i>enum Rand</i> <i>(Exp, Number, Name);</i>	The enumerated type in IDL maps a type name onto a small set of integer values.
<i>union</i>	<i>union Exp switch (Rand) {</i> <i>case Exp: string vote;</i> <i>case Number: long n;</i> <i>case Name: string s;</i> <i>};</i>	The IDL discriminated union allows one of a given set of types to be passed as an argument. The header is parameterized by an <i>enum</i> , which specifies which member is in use.

Instructor's Guide for Coulouris, Dollimore and Kindberg Distributed Systems: Concepts and Design Edn. 4
© Pearson Education 2005

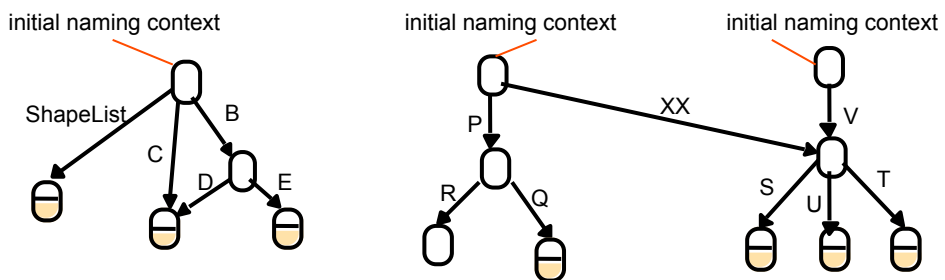
Page 842
CORBA interoperable object references

IOR format

IDL interface type name	Protocol and address details			Object key	
interface repository identifier	IIOP	host domain name	port number	adapter name	object name

Instructor's Guide for Coulouris, Dollimore and Kindberg Distributed Systems: Concepts and Design Edn. 4
© Pearson Education 2005

Figure 20.9
Naming graph in CORBA Naming Service



Instructor's Guide for Coulouris, Dollimore and Kindberg Distributed Systems: Concepts and Design Edn. 4
© Pearson Education 2005

Figure 20.10
Part of the CORBA Naming Service NamingContext interface in IDL

```

struct NameComponent { string id; string kind; };

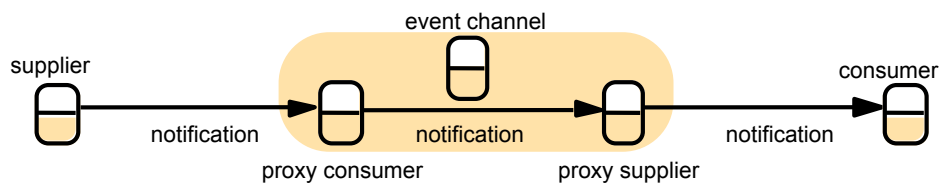
typedef sequence <NameComponent> Name;

interface NamingContext {
    void bind (in Name n, in Object obj);
        binds the given name and remote object reference in my context.
    void unbind (in Name n);
        removes an existing binding with the given name.
    void bind_new_context(in Name n);
        creates a new naming context and binds it to a given name in my context.
    Object resolve (in Name n);
        looks up the name in my context and returns its remote object reference.
    void list (in unsigned long how_many, out BindingList bl, out BindingIterator bi);
        returns the names in the bindings in my context.
};

```

Instructor's Guide for Coulouris, Dollimore and Kindberg Distributed Systems: Concepts and Design Edn. 4
© Pearson Education 2005

Figure 20.11
CORBA event channels



Instructor's Guide for Coulouris, Dollimore and Kindberg Distributed Systems: Concepts and Design Edn. 4
© Pearson Education 2005