Objekti- ja komponentti-väliohjelmistot

Teemat

- Järjestelmämallin tarpeet
- Palvelut ja rajapinnat
- Sisärakenteet
  - networking and interoperability
  - server control
  - system administration infrastructure
- Sovellustilanteen arkkitehtuurimallikkeita
- Realisaatioita
- Laajennustrendit

Järjestelmätarpeet

- Objekti- ja komponentti-vo:lla yhteiset
- Arkkitehtuurimalli
  - Hajautetut objektit (distributed objects)
  - Asiakas-palvelin
- Kehityskaari
  - Käyttöjärjestelmät
    - Laitteistoabstraktio, suoritusaikainen ympstö, peruspalvelut sovelluksille
  - Objektijärjestelmät
  - Komponenttijärjestelmät

Objektijärjestelmät

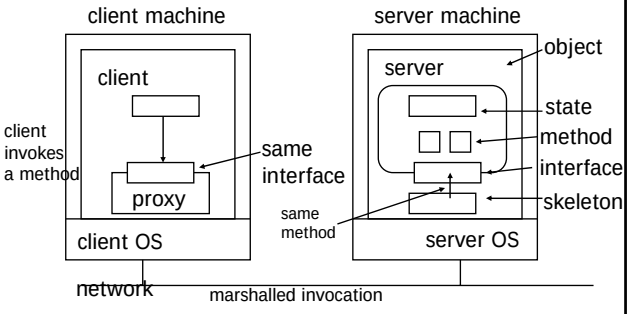
- Objektialustat
  - Objektimalli
    - osoittaminen, rajapinnat, vuorovaikutustavat, objektien implementointi
  - Objektimallin toteutus ja suoritusaikainen ympäristö
    - Objektien luonti ja elinkaaripalvelut
  - Esimerkkejä: CORBA
  - Vertailu objektikielen ja CORBA-mallin mukaisesta ohjelmointitavasta
    - Schmidt, Vinoski: Object Interconnection. SIGS C++ Report Magazine, May 1995.

Objekti- ja komponenttimallit

- Objektimallit keskittyvät operaatioiden käynnistämiseen

  "Objects are discrete software units – they contain data, and can manipulate it. There are four keys: encapsulation, inheritance, polymorphism, and instantiation."
- Komponenttimallit kuvaavat komponentin toimintaympäristöä
- Eroihin johtaneita seikkoja

Distributed object model

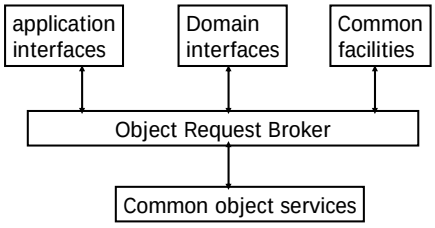


Distributed object model

- CORBA defines *interfaces*, not *implementations*
- It simplifies development of distributed applications by automating/encapsulating
  - Object location
  - Connection & memory mgmt.
  - Parameter (de)marshaling
  - Event & request demultiplexing
  - Error handling & fault tolerance
  - Object/server activation
  - Concurrency
  - Security

Objektijärjestelmä -- esimerkki

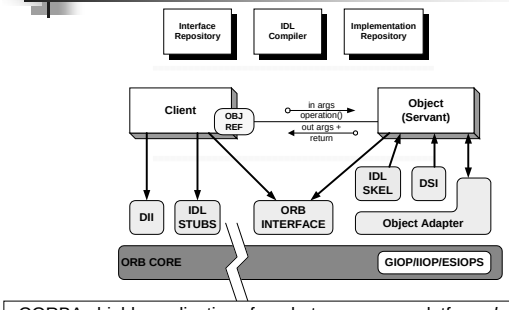
- CORBA (Common Object Request Broker Architecture)



CORBA object services

- Object naming service
- Trading service
- Event service and Notification (async comm)
- Persistence
- Transaction service
- Lifecycle services (factory)
- Externalisation
- Systems management services
- Relationship service (schema)
- Licensing, time, security, property, ...

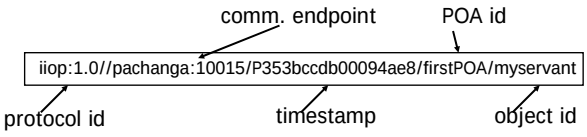
Overview of CORBA



- CORBA shields applications from heterogeneous platform *dependencies*
  - *e.g.,* languages, operating systems, networking protocols, hardware

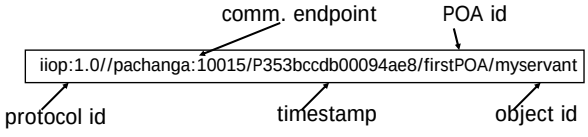
Some CORBA concepts – runtime

- Client
  - The program entity that invokes an operation on an object implementation; ideally, identical call mechanism than with object method call within the programming language
- Object
  - CORBA programming entity that consists of an identity and an implementation, which is known as a servant



Some CORBA concepts

- Servant
  - An implementation programming language entity that defines the operations that support a CORBA IDL interface.
- Object reference
  - Immutable, opaque reference to object (instance)



Some CORBA concepts

- ORB interface and ORB core
  - Abstract interface, standard for communication and helper services
    - manipulating object references
    - binding to common services (naming etc)
    - generic communication primitives
  - Hides object location, implementation, execution state, communication mechanisms
  - IIOP, GIOP
    - request, reply, locaterequest, locatereply, cancelrequest, closeconnection, messageerror, fragment

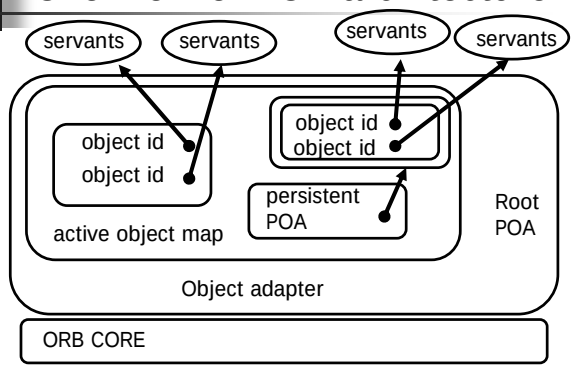
Some CORBA concepts

- stubs and skeletons
  - proxies for marshalling and unmarshalling messages for transfer over ORB
  - ORB is generic, no appl. interface standards but IDL compilation for producing stubs and skeletons
- DII and DSI
  - dynamic invocation interface, dynamic skeleton interface
  - when server interface is introduced ”too late” for static stub selection

Some CORBA concepts

- object adapter
  - has one or more objects under its control
    - unaweare of the specific interfaces of the objects controlled, just a dispatcher
  - Object registration, Object reference generation
  - Request multiplexing
  - Server process activation, Object activation with activation policy
    - threading/concurrency policy
    - persistency policy
    - termination policy
  - Object upcalls
  - Support for transactions, security, threading, etc.

Overview of POA architecture



Some CORBA concepts

- Implementation repository
  - all that is needed to implement and activate objects
  - nonstandard, specific to
    - operating system
    - ORB vendor
    - object adapter
  - table specifying what server should be started, which port number, which file to load and excecute; if already running, where

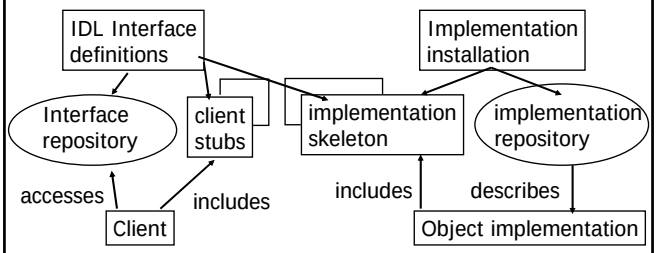
CORBA communication modes

- Object invocation models
  - syncronous (blocks caller) with at-most-once
  - one-way with best effort delivery
  - deferred syncronous with at-most-once (unblocked till syncronises for reply)
- Signaling events
  - event service for basic event channel model
  - notification service for more advanced event channels with filtering etc.
- Message queues
  - messaging service
  - callback model

Some CORBA concepts - production

- CORBA IDL
  - IDL compiler
  - Language mappings
  - Interface inheritance
- Interface repository
  - stores interface definitions
  - relationships – replaceability

Building a CORBA application



CORBA limitations

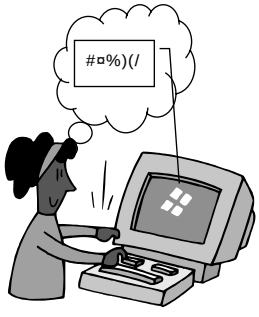
- No standard way to deploy object implementations
- limited standard support for common CORBA server programming patterns
- limited extension of object functionality
  - Inheritance, 1-1 relationship between implementations and interfaces
- Availablity of CORBA object services is not defined in advance
- No object/system management services defined
  - No standard object life cycle management

Object Models and Components

- Issues that lead to different models
  - Communication vs portability
  - Business functionality vs management
  - Advanced functionality vs guaranteed reuse
- Object models describe invocation
- Component models describe component execution environment

Tuumaustauko ...

- Tiedätkö miltä CORBA-objektit näyttävät? CORBA-asiakkaat? IDL-määritykset?
- Etsi itsellesi esimerkki oppikirjoista tai web-tutoriaaleista!
  - pakolliset kehysosat
  - kytkeytyminen ORBiin
  - objektin löytäminen ja sitominen
  - metodikutsujen tekeminen
  - asiakkaan päätöstoimet



Hajautettu CORBA-sovellus

- IDL

```
module HelloApp{
        interface Hello {
                string sayHello();
        };
};
```

Hajautettu CORBA-sovellus

- Asiakas

```
Import HelloApp.*;
//contains generated stubs
Import org.omg.Cosnaming.*;
Import org.omg.CORBA*;
//must be in all CORBA applications

Public class HelloClient {
    public static void main(String args[]) {
    try {
            // create and initialize the ORB
            ORB orb = ORB.init(args, null);
            // get the root naming context
            org.omg.CORBA.object objRef =
              = orb.resolve_initial_references("NameService");
```

```
NamingContext ncRef =
     NamingContextHelper.narrow(objRef);
// Resolve the object reference in naming
NameComponent nc = new NameComponent ("Hello", "");
NameComponent path[] = {nc};
// get the stub
Hello HelloRef =
     HelloHelper.narrow(ncRef.resolve(path));
// Call the Hello server object and print results
   String hello = helloRef.sayHello();
   system.out.println(hello);
} catch (Exception e) {
    System.out.println("Error : " + e);
    e.printStachTrace(System.out);
}
}
```

Hajautettu CORBA-sovellus

- Palvelupuoli

```
Import HelloApp.*;
// contains generated stubs
Import org.omg.Cosnaming.*;
Import omg.org.cORBA.*;

Public class HelloServer {
   public static void main(String args[]);
   //registration, entering server loop
   try{
       ORB orb = ORB.init(args, null);
```

```
// creae the servant and register with orb
HelloServant helloRef = new HelloServant();
orb.connect(helloRef);
//get root naming context
omg.org.CORBA.object objRef =
  orb.resolve_initial_references("NameService"):
NamingContext ncRef =
  NamingContextHelper.narrow(objRef);
//bind the object reference to name
NameComponent nc = new NameComponent("Hello", "");
NameComponent path[] = {nc};
ncRef.rebind(path, helloRef);
```

```
//wait for invocations from clients
java.lang.object sync = new java.lang.Object ();
syncronized (sync)  {
   sync(wait);
}
catch (Exection e) {
   system.err.println("Error: " + s);
   s.printStackTrace (System.out);}}


//the actual service
Class HelloServant extends _HelloImplBase {
  public string sayHello () {
  return "\nHello world!\n";}}
```

Muuta materiaalia

- POA-kalvoja
  - Schmidt, D., Developing Distributed Object Computing Applications with CORBA. Slide set from http://www.eng.uci.edu/~schmidt/PDF/corba4.pdf
- Ohjelmointitutoriaaleja
  - http://www.cs.wustl.edu/~schmidt/ACE_wrappers/TAO/docs/tutorials/Quoter/
  - useimmilla toimittajilla omansa (Sun+Java+CORBA, etc)
- CORBA-sivustoja
  - http://www.cs.wustl.edu/~schmidt/corba.html
  - http://www.puder.org/corba/