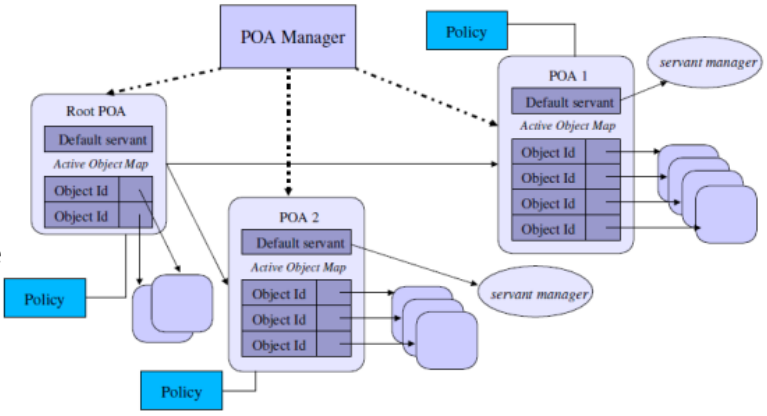
Corba en Java

InfoRep – TDM2

I. Définitions

- **ORB :** Bus sur lequel sont les objets partagés
- **IDL :** Langage de définition d'interface permettant de générer des squelettes
- **IIOP :** Internet Inter-Orb Protocol
- **SSI :** Static Skeleton Interface
- **DSI :** Dynamic Skeleton Interface
- **BOA / LOA / ODA / POA :** XXX Adapter
- **POA :** Portable Object Adapter
- **GIOP :** General Inter-Orb Protocol
- **IOR :** Interoperable Object Reference
- **CDR :** Common Data Representation



II. Spécifications IDL

```
module ServiceDate {
    typedef unsigned short Jour;

    enum Mois {
        Janvier, Fevrier, Mars, Avril, Mai,
        Juin, Juillet, Aout, Septembre,
        Octobre, Novembre, Decembre
    };

    typedef unsigned short Annee;

    struct Date {
        Jour leJour;
        Mois leMois;
        Annee lAnnee;
    };

    typedef sequence<Date> DesDates;

    interface Calendrier {
        attribute Annee anneeCourante;
        boolean verifierUneDate(in Date d);
        void leJourSuivant(inout Date d);
        exception DateErronnee {
            string raison;
        };
        Date convertirChaine(in string uneChaine)
           raises (DateErronnee);
        string convertirDate(in Date uneDate)
           raises (DateErronnee);
    };

    interface CalendrierFerie : Calendrier {
        void lesJoursFeries(in Annee deLAnnee,
                            out DesDates dates);
    };
};
```

III. Commandes

1. Compilation du IDL

```
idlj -f<all|client> -td <outDir>
<file.idl>
```

2. Compilation Java

```
javac -sourcepath ./src -d ./bin
$(find ./src/ -name *.java)
```

3. Lancement orbd

```
orbd -ORBInitialPort 1024
```

4. Lancement serveur

```
java -classpath bin HelloServer
-ORBInitialPort 1024
```

5. Lancement client

```
java -classpath bin HelloClient
-ORBInitialPort 1024
-ORBInitialHost 192.168.0.10
Thomas
```

IV. Creation d'un servant

```
import HelloWorld.*;
import HelloWorld.HelloPackage.*;

class HelloServant extends HelloPOA {
    public String sayHello(String name) throws ChaineVide {
    if (name.equals(""))
        throw new ChaineVide();
    System.out.println("Request from " + name);
    return "Hello "+name;
    }
}
```

V. Création d'un serveur

```
import HelloWorld.*;
import org.omg.CORBA.*;
import org.omg.PortableServer.*;
import org.omg.CosNaming.*;

public class HelloServer {
    public static void main(String args[]) {
        try {
            ORB orb = ORB.init(args, null); // init ORB
            POA rootpoa = POAHelper.narrow(orb.resolve_initial_references("RootPOA")); // Récup RootPOA
            rootpoa.the_POAManager().activate(); // Activation du manager
            HelloServant helloServant = new HelloServant(); // Instanciation d'un servant
            Hello helloCorbaObject =
                HelloHelper.narrow(rootpoa.servant_to_reference(helloServant)); // référencement
            org.omg.CORBA.Object corbaNamingServiceReference =
                orb.resolve_initial_references("NameService"); // Récup NS
            NamingContext corbaNamingServiceObject =
                NamingContextHelper.narrow(corbaNamingServiceReference); // Récup NC
            NameComponent helloNameComponent = new NameComponent("Hello", ""); // Création nom
            NameComponent helloPath[] = {helloNameComponent}; // Création Nnom
            corbaNamingServiceObject.rebind(helloPath, helloCorbaObject); // Bind du nom dans NS
            orb.run(); // Lancement ORB
        }
        catch (Exception e) {
            System.err.println("Error : "+e);
        }
    }
}
```

VI. Création client

```
import HelloWorld.*;
import org.omg.CORBA.*;
import org.omg.CosNaming.*;

public class HelloClient {
    public static void main(String args[]) {
        try {
            ORB orb = ORB.init(args, null); // init ORB
            NamingContext corbaNamingServiceObject =
             NamingContextHelper.narrow(orb.resolve_initial_references("NameService")); // Récup NS
            NameComponent helloNameComponent = new NameComponent("Hello", ""); // Création nom
            NameComponent helloPath[] = {helloNameComponent}; // Création nom
            Hello helloCorbaObject =
                 HelloHelper.narrow(corbaNamingServiceObject.resolve(helloPath)); // Récup objet
            System.out.println(helloCorbaObject.sayHello(args[4])); // Appel méthode
        }
        catch (Exception e) {
            System.err.println("Error : "+e);
        }
    }
}
```