Uma Introdução à Arquitetura CORBA

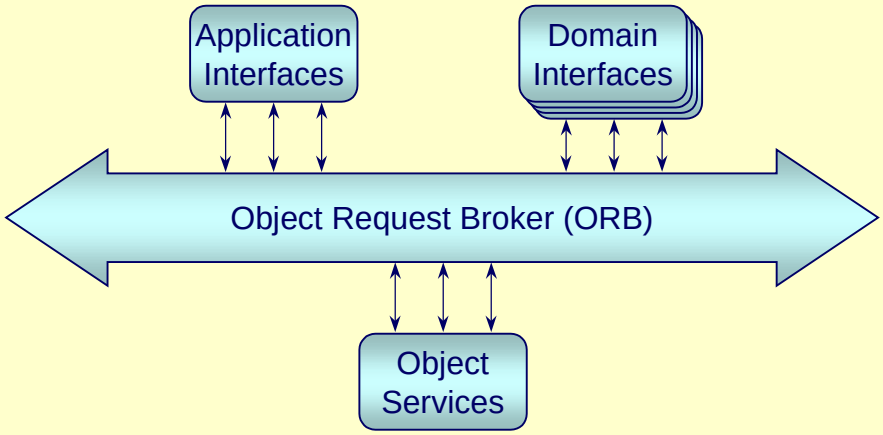
Francisco C. R. Reverbel

O Object Request Broker (ORB)

- Via de comunicação entre objetos (*object bus*), na arquitetura do OMG
- Definido pela especificação CORBA (*Common Object Request Broker Architecture*)
  - 1991: CORBA 1.0
  - 1995: CORBA 2.0
  - revisão atual: CORBA 2.3
  - próxima grande revisão: CORBA 3.0

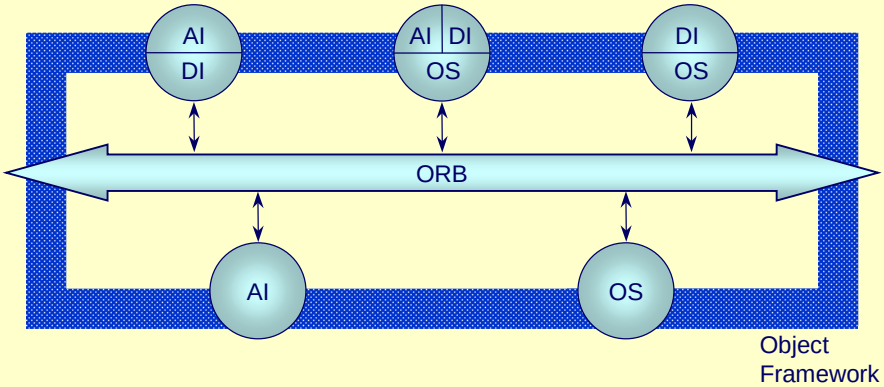
Arquitetura de Gerenciamento de Objetos

OMA (Object Management Arquitecture):



Utilização da OMA

Arcabouço (framework) para aplicações distribuídas:



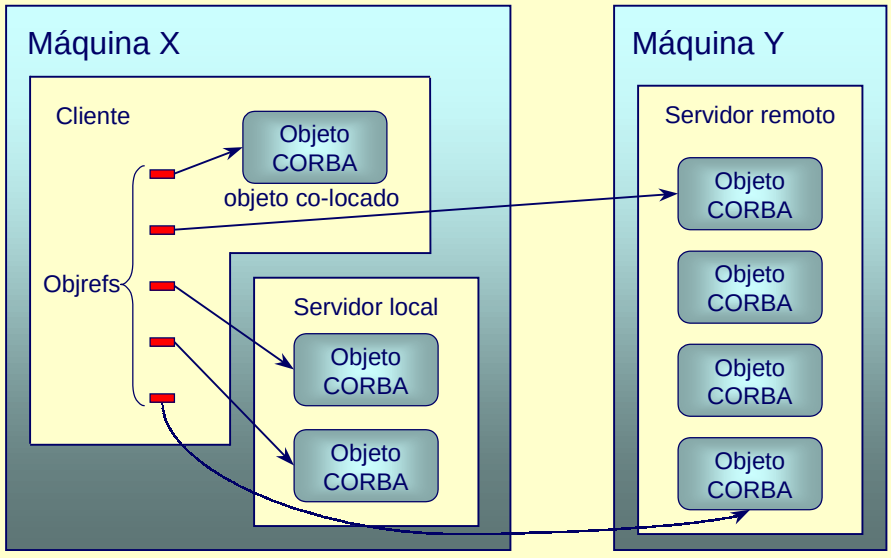
AI = Application Interfaces

DI = Domain Interfaces
OS = Object Services

CORBA

- Arquitetura cliente/servidor orientada a objetos
- Serviço fundamental do ORB: invocação remota de métodos
- Clientes podem chamar métodos de objetos remotos do mesmo modo que chamam métodos de objetos locais (transparência de localização)
- Para chamar métodos de um objeto o cliente precisa ter uma referência para o objeto

Transparência de Localização

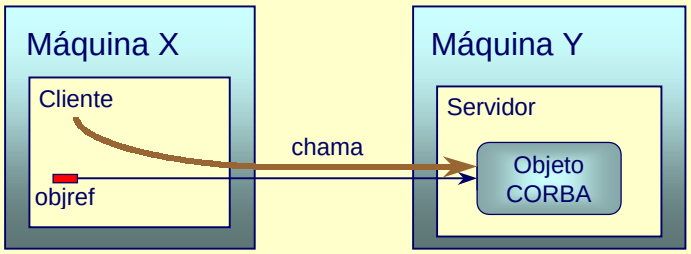


Invocação Remota de Métodos

- O cliente simplesmente chama um método sobre uma object reference
  - O ORB se encarrega de mandar uma mensagem de requisição, esperar a mensagem de resposta e retornar os resultados para o cliente
  - Para o cliente parece uma chamada de método normal
- Independência de linguagens de programação, sistemas operacionais e plataformas de hardware

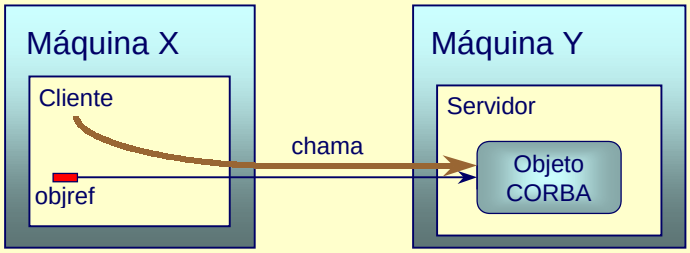
Invocação Remota de Métodos (cont.)

- O que parece acontecer

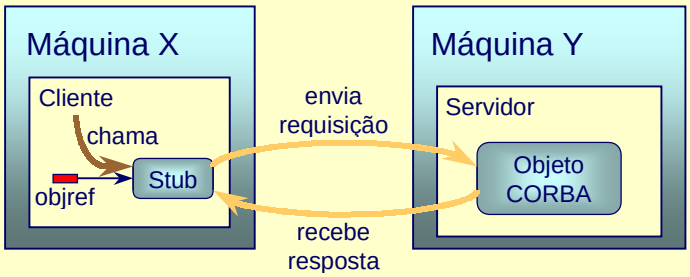


Invocação Remota de Métodos (cont.)

- O que parece acontecer



- O que acontece na verdade

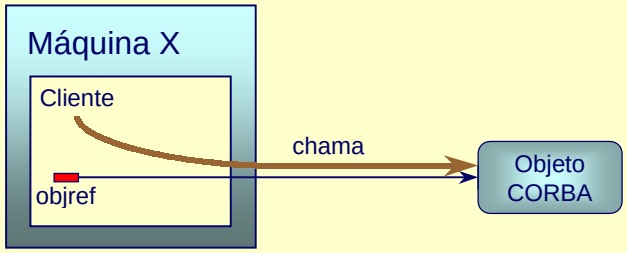


Serventes

- Objetos CORBA são encarnados por entidades da linguagem de programação na qual o servidor foi escrito
- Essas entidades são denominadas serventes
  - Ao longo de sua vida um objeto CORBA pode ser representado por diferentes serventes
- Em C++ ou Java um servente é uma instância de uma classe

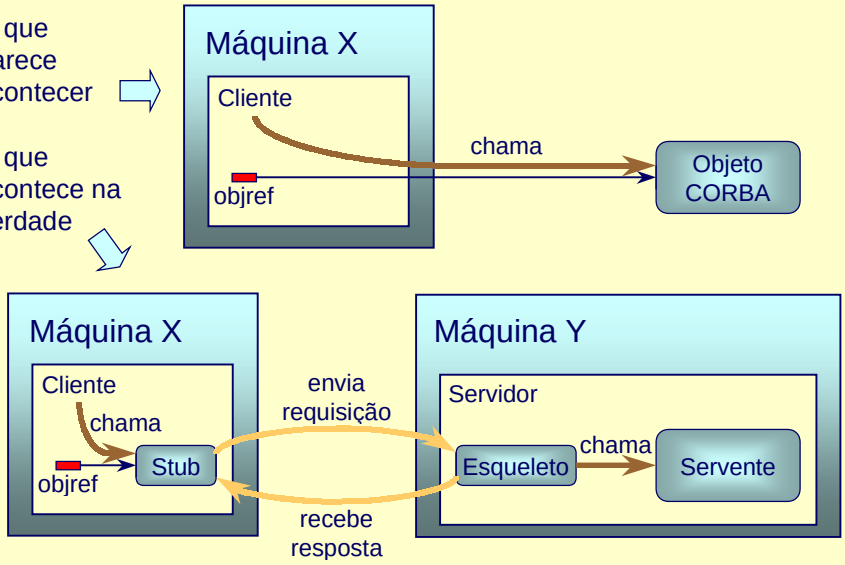
Refinando uma Figura Já Vista

- O que parece acontecer



Refinando uma Figura Já Vista

- O que parece acontecer
- O que acontece na verdade



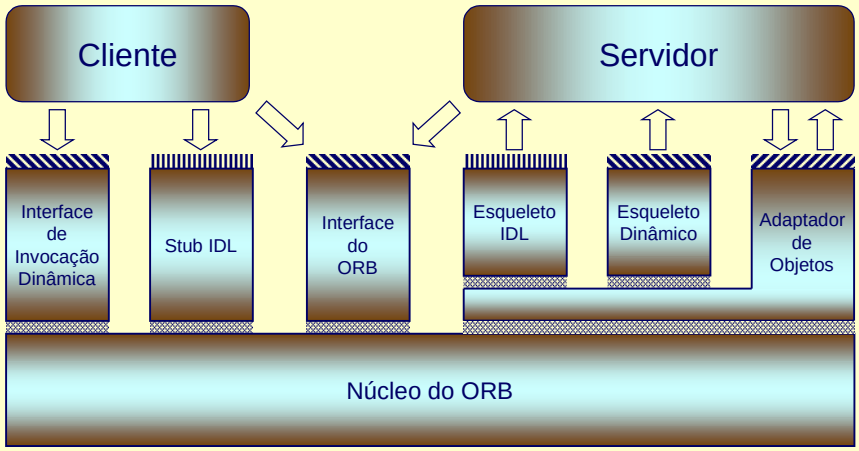
Stubs e Esqueletos

- O usuário não precisa escrever:
  - nem os stubs usado nos clientes
  - nem o esqueletos usados no servidores.
- O que faz o esqueleto:
  - Desempacota os parâmetros recebidos numa mensagem de requisição
  - Chama o método de um servente que implementa a operação requisitada
  - Empacota numa mensagem de resposta os resultados da operação

O Compilador IDL

- Os stubs e esqueletos são automaticamente gerados por um compilador IDL
- O compilador IDL recebe como entrada uma especificação das interfaces dos objetos
- Essa especificação é escrita numa linguagem neutra, a Interface Definition Language (IDL)
  - Independência de linguagem de programação

Componentes de CORBA



Partes da Especificação CORBA

- O núcleo (*core*) do ORB
- A linguagem de definição de interfaces (IDL)
- O repositório de interfaces
- Mapeamentos de IDL para linguagens de programação
- *Stubs* e esqueletos estáticos
- Interfaces de invocação dinâmica (DII) e de esqueleto dinâmico (DSI)
- Adaptadores de objetos (*Object Adapters*)
- O repositório de implementações
- Protocolos (GIOP e IIOP)

O Desenvolvimento de um Sistema

- Etapas Gerais:
  - Defina as interfaces dos objetos e expresse-as em OMG IDL (Interface Definition Language)
  - Gere stubs e esqueletos
  - Implemente objetos serventes
  - Implemente o programa servidor
  - Implemente aplicações clientes

OMG IDL

- Linguagem declarativa usada para especificar interfaces de objetos
  - Sintaxe baseada em C++
  - Suporta herança de interfaces
  - Todas as interfaces de objetos são derivadas de **`CORBA::Object`**
  - CORBA é especificada em IDL
    - de modo a ser independente de linguagem de programação
  - Oferece tipos básicos (pré-definidos) e permite definição de tipos pelos usuários

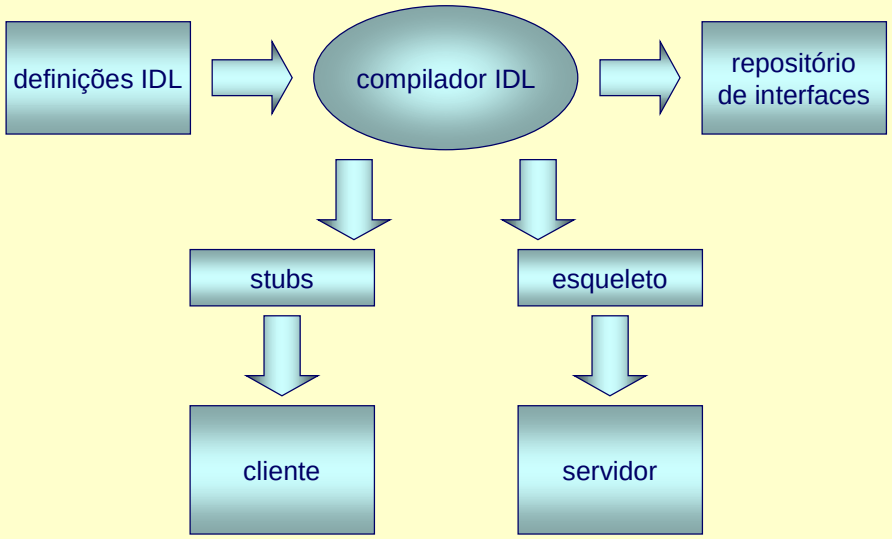
Mapeamentos de IDL

- Definem como IDL é traduzida para as diferentes linguagens de programação
- O OMG padronizou mapeamentos de IDL para C, C++, Smalltalk, COBOL, Ada e Java
  - Há mapeamentos para outras linguagens, mas ainda são proprietários (não padronizados)

Exemplo de Interface IDL

```
// IDL
module Stock {
    exception UnknownStock {
        string name;
    };
    interface Quoter {
        string name();
        long value(in string stock)
            raises(UnknownStock);
    };
};
```

Geração de Stubs e Esqueletos



Servente Java para a Interface Quoter

```
import java.util.Hashtable;
public class QuoterImpl extends Stock.QuoterPOA {
    QuoterImpl(String name) {...}
    public String name() { return myName; }
    public int value(String stock)
        throws Stock.UnknownStock {...}
    private String myName;
    private Hashtable myTable;
}
```

Implementação da Operação **value()**

```
public int value(String stock)
    throws Stock.UnknownStock {
    Integer result =
        (Integer)myTable.get(stock);
    if (result == null) {
        throw new Stock.UnknownStock(stock);
    }
    return result.intValue();
}
```


Construtor da Classe **QuoterImpl**

```
QuoterImpl(String name) {
    myName = name;
    myTable = new Hashtable();

    // inicializa myTable com informações
    // lidas de um arquivo ou banco de dados
    // ...

    //     (num quoter real, myTable deveria
    //      ser atualizado periodicamente)
}
```

Mainline do Servidor de Cotações

```
public static void main(String[] args)
    throws Exception {
    org.omg.CORBA.ORB orb =
        org.omg.CORBA.ORB.init(args, null);
    org.omg.CORBA.Object obj =
        orb.resolve_initial_references("RootPOA");
    POA poa = POAHelper.narrow(obj);
    QuoterImpl quoterServant =
        new QuoterImpl("Reuters");
```


Mainline do Servidor de Cotações

```
    org.omg.CORBA.Object quoter =
        quoterServant._this_object(orb);
    PrintWriter iorFile =
        new PrintWriter(
            new FileWriter("quoter.ref")
        );
    iorFile.println(orb.object_to_string(quoter));
    iorFile.close();
    poa.the_POAManager().activate();
    orb.run();
}
```

Obtenção de Object References

- Clientes precisam obter object references
- Uma object reference é obtida como resultado de alguma operação chamada pelo cliente
  - Clientes não podem “construir” ou “inventar” object references
- Uma object reference pode ser convertida em string e depois reconvertida em object reference
  - Você pode guardar essa string num arquivo
  - Pode divulgá-la como bem entender

Exemplo de Cliente Java

```
public static void main(String[] args) {
    try {
        quoter.ref BufferedReader iorFile =
            new BufferedReader(
                new FileReader("quoter.ref")
        );
        String s = iorFile.readLine();
        iorFile.close();
        org.omg.CORBA.ORB orb =
            org.omg.CORBA.ORB.init(args, null);
```

Exemplo de Cliente Java (cont.)

```
        org.omg.CORBA.Object obj =
            orb.string_to_object(s);
        Quoter quoter = QuoterHelper.narrow(obj);
        System.out.println(
            "Valor de " + args[0] + ": " +
            quoter.value(args[0]) +
            " (fonte: " + quoter.name() + ")”
        );
    }
```


Exemplo de Cliente Java (cont.)

```
    catch (Stock.UnknownStock e) {
        System.out.println(
            "Nao disponível cotação de " +
            args[0]
        );
    }
    catch (Exception e) {
        e.printStackTrace();
  }
}
```

Observações

- No exemplo visto o servidor e o cliente foram escritos na mesma linguagem
- Poderiam ter sido escritos em linguagens diferentes
  - Por exemplo: servidor em C++ e cliente em Java
- CORBA provê interoperabilidade entre
  - linguagens de programação
  - sistemas operacionais
  - arquiteturas de hardware