



CORBA

Common Object Request Broker Architecture

Carlos Ferraz
cagf@cin.ufpe.br



© 2002-2003 Carlos A. G. Ferraz 2



Tópicos

- O Modelo OMA
- A estrutura do ORB
- IDL



© 2002-2003 Carlos A. G. Ferraz 2



Objetos Distribuídos (1/3)

- Características das aplicações atuais
 - Complexas
 - Multilínguas
- Características dos ambientes atuais
 - Distribuídos
 - Heterogêneos



© 2002-2003 Carlos A. G. Ferraz 3



Objetos Distribuídos (2/3)

- Particionar e distribuir dados e funções é uma tarefa bastante difícil
 - O Paradigma de Orientação a Objetos trata desse problema
- Também é bastante difícil produzir aplicações complexas em ambiente de rede
 - Os Sistemas Distribuídos abstraem as características da rede → transparências



© 2002-2003 Carlos A. G. Ferraz 4



Objetos Distribuídos (3/3)

Orientação a Objetos + Sistemas Distribuídos

↓

Objetos Distribuídos



© 2002-2003 Carlos A. G. Ferraz 5



CORBA

- Common Object Request Broker Architecture
 - Padrão para desenvolvimento de aplicações distribuídas para sistemas heterogêneos usando orientação a objetos
- OMG: Object Management Group (<http://www.omg.org>)
 - Consórcio de empresas responsável pela proposição e manutenção do padrão CORBA
 - Criado em 1989



© 2002-2003 Carlos A. G. Ferraz 6

CORBA

■ **Objetivos dos padrões OMG**

- Permitir especificação de serviços separados da implementação
 - Interfaces de Objetos
 - Implementação de Interfaces
- Projetar componentes que se descubram e interoperem através de um "barramento" de objetos

© 2002-2003 Carlos A. G. Ferraz 7

Separação de Conceitos

© 2002-2003 Carlos A. G. Ferraz 8

O Modelo de Referência OMA

Object Management Architecture

© 2002-2003 Carlos A. G. Ferraz 9

O Modelo de Referência OMA

■ **Visão do OMG de como deve ser a arquitetura básica de um sistema distribuído**

■ **Dois modelos:**

- **Modelo de Objetos**
 - ◆ Define conceitos tradicionais de orientação a objetos (objetos, herança, interface etc)
- **Modelo de Referência**
 - ◆ Relaciona serviços de um sistema distribuído passíveis de serem padronizados e, posteriormente, implementados por diferentes empresas

© 2002-2003 Carlos A. G. Ferraz 10

O Modelo de Referência OMA

Object Management Architecture

© 2002-2003 Carlos A. G. Ferraz 11

O Modelo de Referência OMA

■ **Serviços CORBA: serviços para manipulação de objetos**

- Exemplo: criação de objetos, serviço de nomes, segurança, persistência, etc

■ **Objetos de Aplicação: serviços específicos de uma determinada aplicação**

- Não são objeto de padronização

© 2002-2003 Carlos A. G. Ferraz 12

O Modelo de Referência OMA

■ Facilidades CORBA: serviços para aplicações

■ Facilidades Horizontais: Genéricas para diversas aplicações

- Exemplo: Interface de Usuário, Gerenciamento de Tarefas, etc.

■ Facilidades Verticais: Específicas de uma área de conhecimento

- Exemplo: telecomunicações, financeira, indústria etc.

ORB: Object Request Broker

■ Conceito central da arquitetura OMA

■ “Barramento lógico de software” que controla toda a comunicação entre os componentes da arquitetura

■ Facilitador da comunicação entre objetos distribuídos

ORB: Object Request Broker

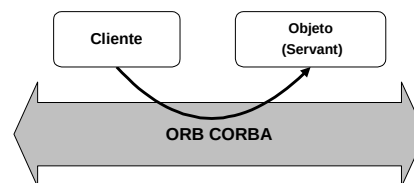
■ Objetivo:

- Transparência de localização
- Transparência de acesso
- Independência de sistema operacional e linguagem

■ CORBA: padrão de ORB

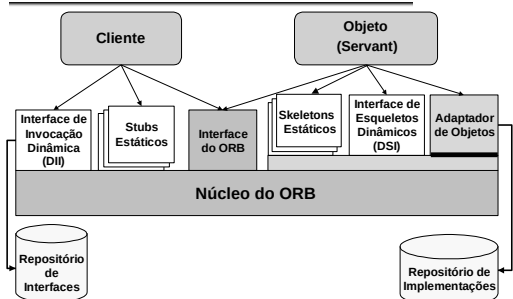
- Não é um produto, e sim uma especificação

ORB: Object Request Broker



Anatomia de um ORB CORBA

ORB CORBA



ORB CORBA

■ Cliente

- Acesso a objetos e operações
- Transparência de implementação dos objetos acessados

■ Objeto

- É uma instância de uma Interface IDL
- É definido por uma **object reference**
- Utiliza o conceito de **Objectid** para associar um objeto com suas implementações (servants)

■ Servant

- Implementa as operações das interfaces IDL

© 2002-2003 Carlos A. G. Ferraz

19

ORB CORBA

■ Núcleo do ORB

- Representação básica de objetos e comunicação
- Lida com referências a objetos
- Implementação específica

■ Interface do ORB

- Conjunto de operações para a manipulação de informações do ORB

© 2002-2003 Carlos A. G. Ferraz

20

ORB CORBA

■ Lado Cliente

■ Repositório de Interfaces

- Repositório de Descrições de Interfaces utilizado para a realização de invocações dinâmicas

■ Interface de Invocação Dinâmica (DII)

- Conjunto de operações utilizadas para a criação de invocações dinâmicas

■ Stubs Estáticos

- Stubs para invocações estáticas

© 2002-2003 Carlos A. G. Ferraz

21

ORB CORBA

■ Lado Servidor (Servant)

■ Repositório de Implementações

- Armazenamento de meta-dados sobre implementações de objetos

■ Esqueletos Estáticos

- Stubs utilizados para invocações estáticas

■ Interface de Esqueletos Dinâmicos (DSI)

- Conjunto de operações para a manipulação de invocações dinâmicas

■ Adaptador de Objetos

- Módulo responsável pelo controle de execução dos servants

© 2002-2003 Carlos A. G. Ferraz

22

Linguagem de Definição de Interfaces

IDL

IDL (Interface Definition Language)

■ Principal componente do padrão CORBA

■ A neutralidade de IDL define a heterogeneidade de CORBA

■ Linguagem para definir a interface de um objeto remoto

- Especifica a assinatura (cabeçalho) das operações que serão implementadas por um objeto remoto

© 2002-2003 Carlos A. G. Ferraz

24

IDL (Interface Definition Language)

■ Linguagem declarativa (sem código)

- Tipos pré-definidos: *long, short, float, double, char, boolean, enum*;
- Tipos estruturados: *struct, union, string, sequence*;
- Tratamento de exceções (*exception*);
- Modos de passagem de parâmetros: *in, out* e *inout*;

■ Idéia Básica: especificar serviços em uma linguagem neutra

© 2002-2003 Carlos A. G. Ferraz

25

IDL (Interface Definition Language)

■ Exemplo:

```
typedef sequence<string> extrato;  
struct Data {  
    short dia, mes, ano;  
};  
interface conta {  
    float obterSaldo ();  
    extrato obterExtrato (in Data inicio, in Data fim);  
    .....  
};
```

© 2002-2003 Carlos A. G. Ferraz

26

IDL (Interface Definition Language)

■ Compiladores de IDL geram automaticamente:

- stubs (clientes)
- esqueletos (servidores)
- Diversos outros objetos de suporte

■ Independência de linguagem:

- Compiladores IDL implementam geração de código para C, C++, Smalltalk, Ada, COBOL, Java, etc
- CORBA especifica como deve ser o mapeamento (binding) de IDL para cada linguagem

© 2002-2003 Carlos A. G. Ferraz

27

Implementações CORBA

■ Produtos que implementam o padrão CORBA:

- Visibroker (<http://www.inprise.com>)
- Orbix (<http://www.iona.com>)
- omniORB (<http://www.uk.research.att.com/omniORB>)
- Java IDL (<http://java.sun.com/j2se/1.4/docs/guide/idl/index.html>)
- ...

© 2002-2003 Carlos A. G. Ferraz

28