Capítulo VI – CORBA

Common Object Request Broker Architecture

OMG: Object Management Group

. Organização internacional, sem fins lucrativos, fundada em 1989

. Mais de 800 membros (incluindo fabricantes de sistemas, produtores de software e utilizadores)

. Objectivos:

- desenvolver a especificação de uma infra-estrutura para computação distribuída

- promover o uso de tecnologias baseadas em objectos

(promover reutilização, portabilidade e inter-operabilidade de sw em sistemas distribuídos heterogéneos)

From: Fintan Bolton
Pure CORBA
SAMS, 2001

From: Coulouris, Dollimore and Kindberg
Distributed Systems: Concepts and Design
Edition 3, © Addison-Wesley 2001

Paula Prata,

Departamento de Informática da UBI

http://www.di.ubi.pt/~pprata

Capítulo VI – CORBA

Common Object Request Broker Architecture

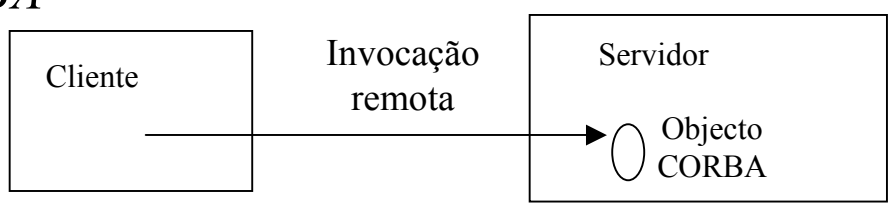
1ª especificação em 1991
CORBA 2.0 (1996)
CORBA 3.0

Características

. Paradigma orientado a objectos

Operações remotas são agrupadas em interfaces (tal como as funções (c++), ou os métodos (Java) são agrupadas em classes.

Uma instância de uma interface é um objecto em CORBA



. Transparência de localização

Sintaxe idêntica, qualquer que seja a localização do objecto

. Independente de qualquer linguagem de programação ou sistema operativo

(c, c++, Java, Ada, Cobol, ...)

. Suporte para ligação a tecnologias alternativas

(bridges para DCOM, DCE, ...)

IDL – Interface Definition Language

A interface de um objecto CORBA é definida através de uma linguagem de definição de interfaces (IDL):

- Linguagem puramente declarativa

(não contém construtores para avaliar expressões ou descrever algoritmos)

- Independente de qualquer linguagem, da plataforma, dos detalhes de implementação

- Uma interface em IDL é um modelo conceptual para descrever objectos

não é necessária em tempo de execução

- Vai ser necessário fazer a tradução para a linguagem que se pretende através de um compilador de interfaces

- (a linguagem pode não ser OO)

- Esse compilador, a partir da interface, gerará os stubs e skeletons da aplicação

CORBA IDL

Exemplo:

```
module BankExample {
  interface Account {
    string getName();
    long getAccountN ();
    boolean depositMoney (in float amount);
    boolean transferMoney ( in float amount,
                            in long destination_accountN,
                            out long confirmationN) ;
  };
};
```

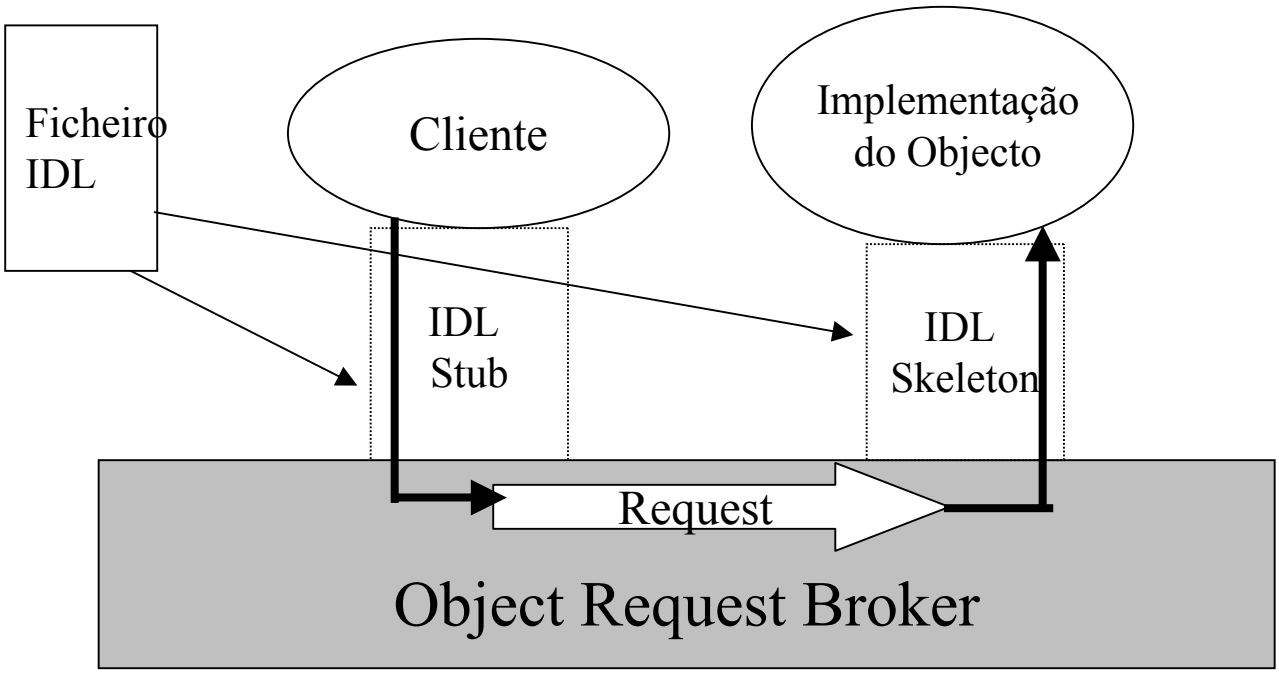
. Direcção da transmissão: parâmetros in, out, inout)

. Parâmetros passados por valor

Depois de definir as interfaces IDL, há que decidir que plataforma e linguagens a usar para a implementação do cliente e servidor.

Um compilador de IDL faz o mapeamento da interface na linguagem pretendida

Object Request Broker



Os objectos CORBA são ligados através de um Object Request Broker (ORB)

. ORB é o que está entre o cliente e a implementação de um objecto permitindo que o cliente invoque operações no objecto

. Responsável por: encontrar a implementação do objecto, marshalling / unmarshalling de argumentos, tratar da execução do código servidor

. Clientes e implementações podem interagir via ORBs diferentes

Mapeamento de IDL em Linguagens de Programação

"Language Mappings" especificam como as definições IDL são traduzidas para a linguagem alvo.

Em linguagens OO os objectos CORBA são vistos como objectos da linguagem

O mapeamento inclui definições para:

- tipos de dados da linguagem
- interfaces para aceder aos objectos através do ORB
- interfaces dos stubs e skeletons
- ...

Compiladores IDL são incluídos com os ORB's

Ex.lo:

. Java 2 ORB (SUN) http://www.sun.com

. VisiBroker for Java (Inprise) http://www.visigenic.com

. OrbixWeb (Iona Technologies) http://www.iona.com

. WebSphere (IBM) http://www.ibm.com

. ...

Referências para Objectos

Um cliente para fazer uma invocação sobre um objecto necessita da sua referência remota:

- Uma referência é um"handle" com a informação necessária para identificar e localizar o objecto

- Torna o acesso ao objecto transparente da localização

encapsula o endereço da máquina onde se encontra a implementação do objecto CORBA

IOR – Interoperable Object reference
(IP do servidor, nº porto onde o servidor escuta, objecto id., ..)

- A representação dada a um cliente só é válida durante o tempo de vida desse cliente

- Podem existir várias referências diferentes para o mesmo objecto.

- Referências são fortemente tipadas, permitem verificação de tipos em tempo de execução

- Referências podem ser tornadas persistentes

- Referências podem ser passadas como parâmetros entre diferentes processos.

- Serviço de Nomes: Interoperable Naming Service
- permite a um cliente aceder à referência de um objecto a partir do seu nome

Interface de invocação dinâmica

(DII - Dynamic Invocation Interface)

A invocação de um método num objecto pode ser feita de uma forma estática (o cliente sabe a referência para o objecto e a interface do objecto em tempo de compilação), invocando o código num stub apropriado para a interface do objecto

ou

pode construir o pedido dinamicamente, através da

Dynamic Invocaton Interface

- permite a construção dinâmica de invocações

- a informação que caracteriza uma operação é obtida pelo cliente em tempo de execução, a partir de um repositório de interfaces (Implementation Repository)

- a DII permite a invocação de operações em objectos cuja interface não era conhecida quando o cliente foi escrito

Desvantagens:

- complexo

- não tem verificação estática de tipos

- menos eficiente

Repositório de Interfaces

Permite guardar de forma persistente a informação das interfaces permitindo consultar em tempo de execução:

- nomes de interfaces

- assinaturas de métodos

...

Repositórios de implementações

Permite guardar de forma persistente implementações de objectos.

- permite ao ORB activar objectos remotos em resposta à invocação de clientes

Protocolos de comunicação

Inicialmente, cada ORB usava um protocolo próprio para comunicação entre diferentes máquinas.

Com o CORBA 2.0 foi definido o protocolo,

"General Inter-ORB protocol (GIOP), independente da camada de transporte, que passou a permitir comunicação entre ORB's diferentes

Finalmente, o "Internet Inter-ORB protocol" (IIOP) especifica como o GIOP pode ser implementado sobre TCP/IP.

Object Adapters

- Um “object adapter” é o meio pelo qual uma implementação de um objecto pode aceder a serviços fornecidos pelo ORB

Alguns desses serviços:

- criação e interpretação das referências para objectos

- activação / desactivação de implementações de objectos

Em CORBA 2.0 foi definido o “basic object adapter”

(BOA) – sendo ambíguo em alguns aspectos,originou implementações diferentes e pouco portáveis entre ORBs diferentes

CORBA 2.2 define o Portable Object Adapter (POA)

- permite implementações portáveis entre ORBs

- objectos com identidades persistentes

CORBA vs RMI

. Interfaces CORBA são definidas em IDL enquanto interfaces RMI são definidas em Java

. CORBA independente da linguagem, em CORBA podemos ter objectos implementados em linguagens diferentes, RMI foi concebido para um ambiente em que todos os objectos são definidos em Java

. Objectos CORBA não são reciclados, RMI tem sistema de “Garbage Collection”

. RMI não suporta parâmetros de saída (out) e de entrada e saída (inout)

...