

Metadados

Repositório de
Interfaces em CORBA

Metadados

Consiste do ingrediente que permite-nos criar sistemas cliente/servidor ágeis.

Um sistema ágil é auto-descritível.
dinâmico e reconfigurável.

Metadados

O sistema auxilia componentes descobrirem cada outro em tempo de execução. Ele provê informação que permite eles interoperarem.

Provê informação de metadados que ferramentas de aplicação podem usar e gerenciar componentes.

Metadados

Um sistema ágil diferencia ele próprio do sistema cliente/servidor convencional, pelo uso de metadados, que consistentemente descreve todos os serviços disponíveis, componentes e dados.

Metadados

Permitem componentes desenvolvidos independentemente, dinamicamente, descobrirem a existência de outros componentes.

A disseminação de metadados é um ingrediente chave em uma infraestrutura de componentes distribuídos.

Metadados

Sem eles, somente teremos um cliente/servidor convencional sem nenhuma flexibilidade.

CORBA é um exemplo de um sistema ágil.

Todo objeto ou serviço a nível de sistema de um barramento CORBA (ORB) deve ser auto-descritível.

Metadados

Mesmo, o barramento CORBA é auto-descritível.

CORBA é totalmente auto-descritível.

Para ser auto-descritível é necessário uma linguagem para se descrever os metadados.

A linguagem de metadados do CORBA é a IDL (Interface Definition Language).

Repositório de Interfaces

O repositório de metadados do CORBA é o Repositório de Interfaces.

É uma base de dados em tempo de execução que contém as especificações de interfaces de cada objeto que o ORB reconhece.

Repositório de Interfaces

Pense nele como um consultável e atualizável base de dados que contém informação gerada em IDL.

É uma base de dados on-line de definições de objetos.

Pode-se capturar essas definições diretamente de um compilador IDL ou através do Repositório de Interfaces do CORBA.

Repositório de Interfaces

CORBA não cuida de como a informação em um repositório de interfaces é colocada lá.

CORBA detalha como a informação é organizada e recuperada do repositório.

Repositório de Interfaces

CORBA especifica um conjunto de classes cujas instâncias representam a informação que está lá no repositório.

É uma base de dados de objetos altamente flexível que guarda pista de coleções de objetos organizados na mesma linha como em IDL.

Repositório de Interfaces

Todos os objetos no repositório são versões compiladas da informação que está em um arquivo-fonte IDL.

Porque um IR é necessário

Um ORB necessita entender a definição dos objetos com os quais ele está trabalhando.

Um modo para obter estas definições é incorporar a informação dentro de *stubs*.

O outro modo é que esta informação seja dinamicamente acessível através do IR.

O que faz o ORB com a informação do IR ?

Ele pode usar as definições de objetos para fazer o seguinte:

1. Provê a verificação de tipos das assinaturas dos métodos.
2. Auxilia conectar ORBs, formando um federação multi-ORB.
3. Provê informação metadados para clientes e ferramentas.
4. Provê objetos auto-descritíveis. Invocando-se *get_interface()* sobre qualquer objeto CORBA, obtém-se sua definição de interface.

Classes do IR

O IR é implementado como um conjunto de objetos que representam a informação nele.

Estes objetos devem ser persistentes.

CORBA agrupa os metadados em módulos que representam espaços de nomes.

Classes do IR

Os nomes dos objetos no repositório são únicos. Dentro de um módulo.

CORBA define uma interface para cada uma de suas estruturas IDL.

Classes do IR

ModuleDef define um agrupamento local de interfaces.

InterfaceDef define as interfaces de objetos.

OperationDef define um método sobre uma interface de objeto.

ParameterDef define um argumento de um método.

Classes de IR

AttributeDef define os atributos de uma interface.

ConstantDef define uma constante nomeada.

ExceptioDef define as exceções que podem surgir por uma operação.

TypeDef define os tipos nomeados que são parte de uma definição IDL.

Interface Raiz

Em adição a estas oito interfaces que representam estruturas IDL, CORBA especifica uma interface **Repository** que serve como raiz de todos os módulos contidos em uma espaço de nomes do repositório.

Hierarquia de *Containment* para as classes do IR.