

Plataformas de Distribuição de Objetos

■ Denominações Equivalentes:

- Modelos de Componentes
- Modelos de Integração de Objetos

■ Motivação:

- Reuso de objetos



- Redução do Tempo e do Custo de Desenvolvimento de Software

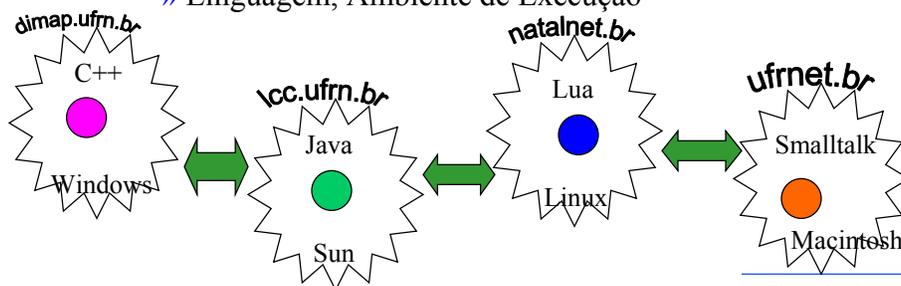
Thais Batista

Objetivo

■ TRANSPARÊNCIAS:

- Distribuição
- Heterogeneidade

» Linguagem, Ambiente de Execução



Thais Batista

Plataformas de Distribuição de Objetos

■ Exemplos:

- CORBA (Common Object Request Broker Architecture): OMG
- COM (Component Object Model) e .NET: Microsoft
- JavaBeans & Enterprise JavaBeans: Sun

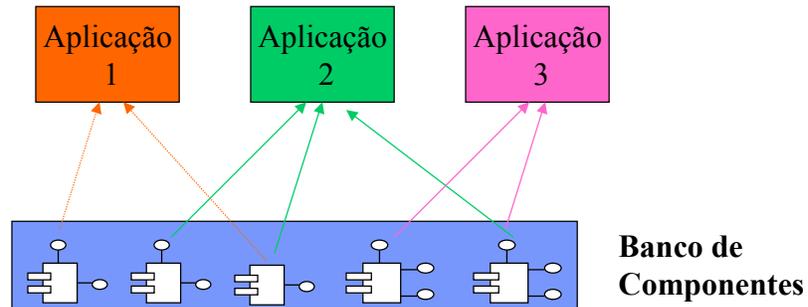
Thais Batista

Componentes

- Entidades de Software compiladas separadamente que interagem com outras entidades apenas através das suas *interfaces*.
- Componentes visam *Composição*
- Composição é a junção de componentes pré-fabricados

Thais Batista

Desenvolvimento Baseado em Componentes



Thais Batista

Interface

- Conjunto de operações (métodos) que podem ser chamadas pelos clientes

```
Objeto pessoa
{
  string nome;
  short identidade;
  long data_nascimento;

  void cadastro(in string nome, in short id, in long data)
  void modifica(in string nome, in short id, in long data)
  void imprime_todas()
}
```

Thais Batista

CORBA



■ CORBA (Common Object Request Broker Architecture)

- padrão para interoperabilidade de objetos distribuídos
 - » modelo cliente-servidor
 - » heterogeneidade de linguagens
 - » distribuição
- produto de um consórcio chamado OMG (Object Management Group)

Thais Batista

CORBA



■ Programação Modular + Programação Orientada a Objetos

- » encapsulamento de dados
- » objetos em interação que encapsulam a implementação
- » definição de interfaces dos objetos
 - substituição da implementação sem afetar o uso do objeto
 - desenvolvimento de software *plug-and-play*

Thais Batista

CORBA

- ORB (Object Request Broker): barramento através do qual objetos trocam mensagens
- Interface de Objetos escritas em IDL (Interface Definition Language)
- Implementação dos Objetos em qualquer linguagem que possua o mapeamento (binding) para CORBA

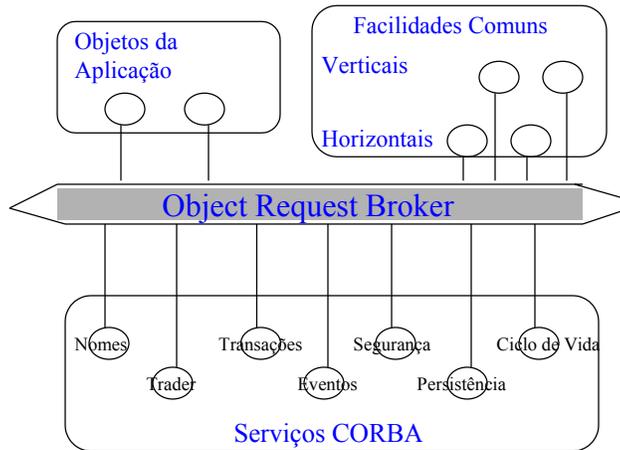
Thais Batista

Binding CORBA

- Mapeamento entre uma linguagem e a IDL que especifica como tipos IDL são convertidos em tipos e chamadas de funções da linguagem

Thais Batista

Arquitetura Global da OMG



Thais Batista

IDL

- Linguagem Declarativa
- Usada para expressar
 - atributos
 - operações que o objeto pode realizar
 - parâmetros de entrada e de saída
 - exceções
 - eventos que um objeto pode emitir

Thais Batista

Estrutura da Interface IDL

```
Module <identificador>
{
  <declarações_de_tipo>;
  <declarações_de_constantes>;
  <declarações_de_exceções>;

  interface <identificador> [[:<lista_interfaces_base>]]
  {
    <declarações_de_tipo>;
    <declarações_de_constantes>;
    <declarações_de_atributos>;
    <declarações_de_exceções>;

    [<tipo_operação>] <identificador> (<parâmetros>)
    [ raises exceção ] [contexto];
  }
}
```

Thais Batista

Exemplo IDL

```
Interface pessoa
{
  string nome;
  short identidade;
  long data_nascimento;

  void add_pessoa(in string nome, in short id, in long data)
    raises (JaCadastrado);
}

Interface professor: pessoa
{
  string disciplina;

  void add_prof(in string nome, in short id,
    in long datanasc, in string disc)
    raises (JaCadastrado);
}

module pessoas_da_escola
{
  interface estudante:pessoa
  {
    short matricula;
    string curso;
    void add_estud(in string nome, in short id, in long datanasc,
      in short mat, in string cur)
      raises(JaCadastrado);
  }
}
```

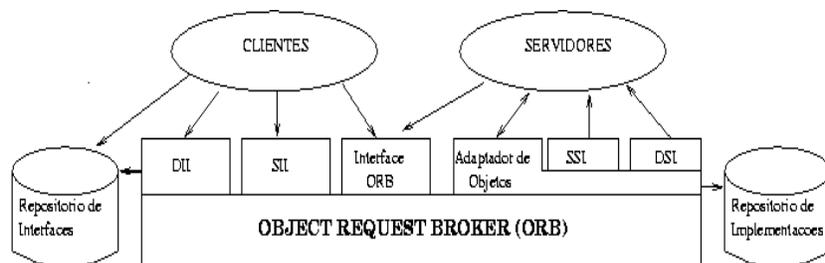
Thais Batista

Referência do Objeto

- Identificação do Objeto CORBA
- Atribuída pelo ORB na criação do objeto.
- Para usar o objeto, o cliente deve obter a referência do objeto pois em uma invocação de um método sobre o objeto, o ORB o identifica através da sua referência.
- Pode ser obtida através do serviço de Nomes ou do serviço de Trading

Thais Batista

Arquitetura CORBA



Thais Batista

ORB - Object Request Broker

- Realiza as tarefas de **Comunicação** e **Coordenação** entre objetos
- Intercepta as chamadas e direciona-as para o objeto apropriado

Thais Batista

Repositório de Interfaces (RI)

- O RI armazena todas as definições IDL de objetos CORBA disponíveis
- CORBA oferece um conjunto de métodos para leitura e navegação no RI
- O *RepositoryID* identifica globalmente os elementos do RI
 - Exemplo: o RepositoryID da interface *professor* no módulo *pessoas_da_escola* poderia ser:
IDL:Escola/pessoas_da_escola/professor/:1.0”

Thais Batista

Repositório de Implementações

- Armazena informações sobre a implementação dos objetos
 - as classes que um servidor implementa
 - os objetos que estão instanciados
 - as referências dos objetos

Thais Batista

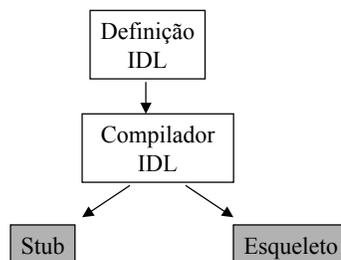
Solicitações a Objetos

- Podem ser elaboradas de duas maneiras:
 - Estaticamente: usando a Interface de Invocação Estática (SII)
 - Dinamicamente: usando a Interface de Invocação Dinâmica (DII)

Thais Batista

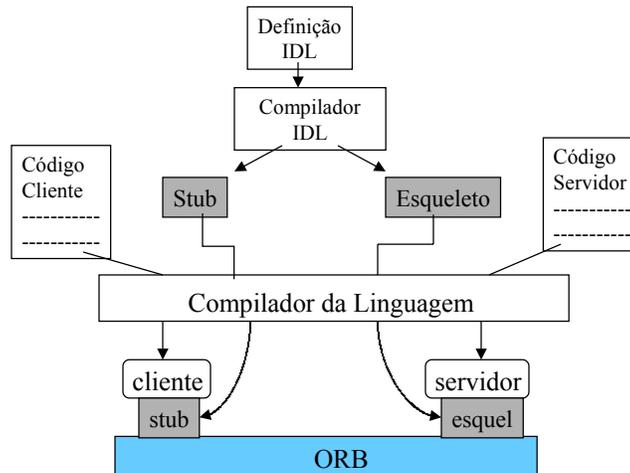
SII

- Especificações IDL são compiladas por um compilador IDL que gera stubs do cliente (simplesmente stubs) e stubs do servidor (esqueletos de programas)



Thais Batista

Programação CORBA Estática Visão Geral



Thais Batista

Problemas da Invocação Estática

- Cliente deve conhecer o stub IDL para cada interface usada
- Ligação estática entre programas clientes e as interfaces IDL dos objetos servidores
- Necessidade de modificação das interfaces implica em rever todos os programas clientes e servidores

Thais Batista

DII

- Permite a invocação de métodos sem necessidade de stubs.
 - Descobrir os novos objetos
 - Descobrir suas interfaces (operação `get_interface`)
 - Obter a definição do método
 - Criar uma lista de argumentos
 - Criar a solicitação
 - Fazer a chamada

Thais Batista

Interface de Esqueleto Estático (SSI)

- Para o servidor é transparente se a invocação do método é estática ou dinâmica (o ORB provê esta transparência)
- Funções da SSI:
 - desempacota as solicitações para transmitir para as implementações dos objetos
 - empacota os resultados para envia-los ao cliente

Thais Batista

Interface de Esqueleto Dinâmico (DSI)

- Trata chamadas sem stubs
- Funções da DSI:
 - para obter o objeto e o método destino da invocação, consulta os valores dos parâmetros que vêm na mensagem
 - Todas as operações são direcionadas a um esqueleto dinâmico que repassa a chamada

Thais Batista

Serviços CORBA

■ Serviço de Nomes

- mapeamento entre um nome simbólico e uma referência de objeto
- define interfaces para inserção e remoção de nomes, consulta e navegação pelo banco de nomes (repositório)
- permite que nomes sejam estruturados hierarquicamente em *Contextos* - estruturas similares a diretórios

Thais Batista

Continuação Serviço de Nomes

```
Interface NamingContext {  
    void bind(in Name n, in Object obj) // liga um nome a um objeto  
        raises(NotFound, AlreadyBound, ...)  
    ....  
    Object resolve(in Name n) // procurar um objeto com nome n  
    ...  
    void unbind(in Name n)  
        raises(NotFound, ...)  
    ...  
};
```

Thais Batista