O Adaptador de Objetos CORBA

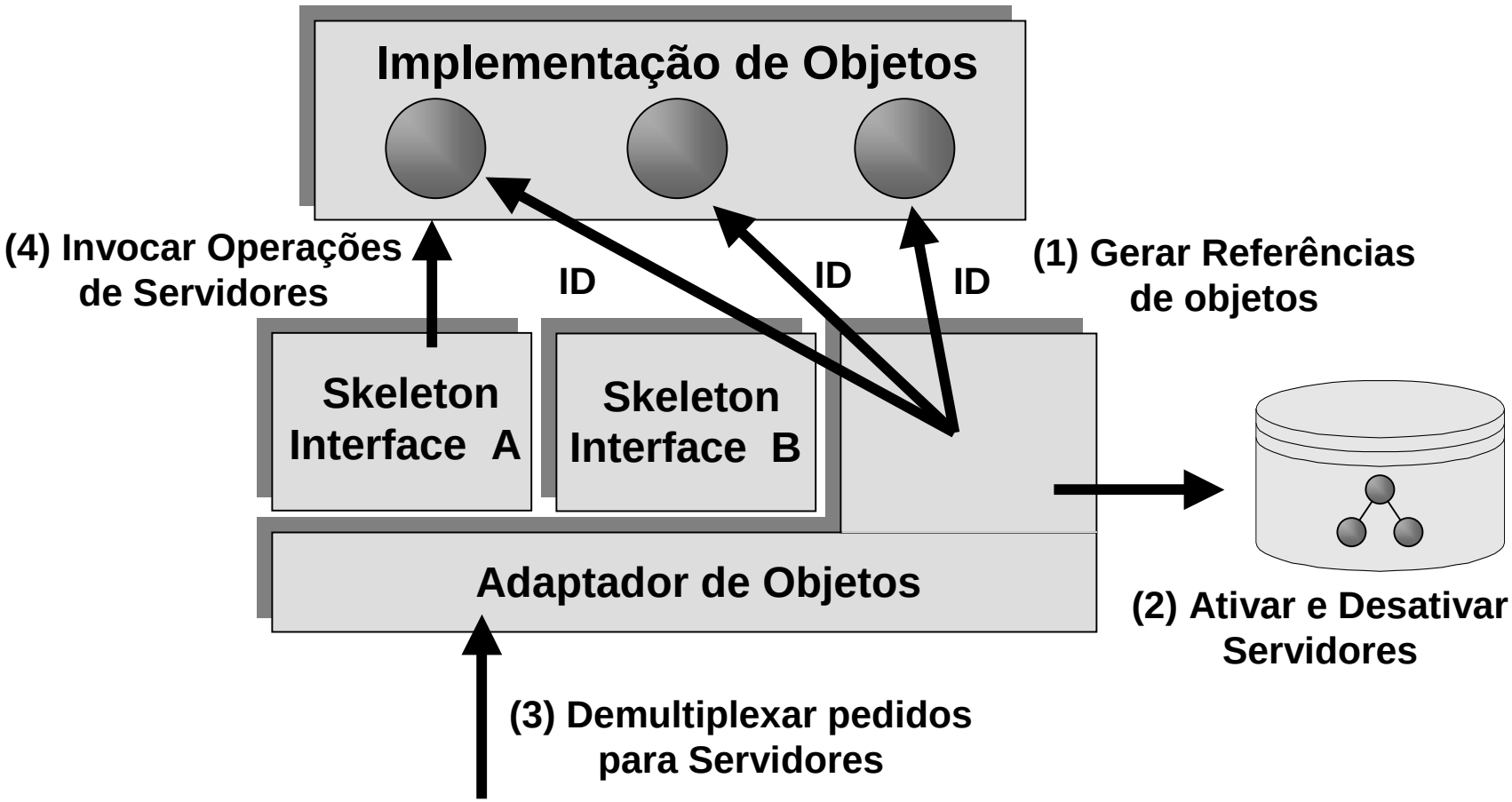
© 2002-2003 Carlos A. G. Ferraz

O Adaptador de Objetos

- O Adaptador de Objetos de CORBA é responsável por:
  - Gerar Referências de Objetos
  - Ativar e desativar servidores (Servants)
  - Demultiplexar pedidos para servidores
  - Coloborar com esqueletos IDL para invocar operações em servidores

© 2002-2003 Carlos A. G. Ferraz

O Adaptador de Objetos



© 2002-2003 Carlos A. G. Ferraz

Portable Object Adapter (POA)

- O POA (Portable Object Adapter) é a especificação atual para o adaptador de objetos CORBA
- Substitui a versão anterior denominada BOA (Basic Object Adapter)

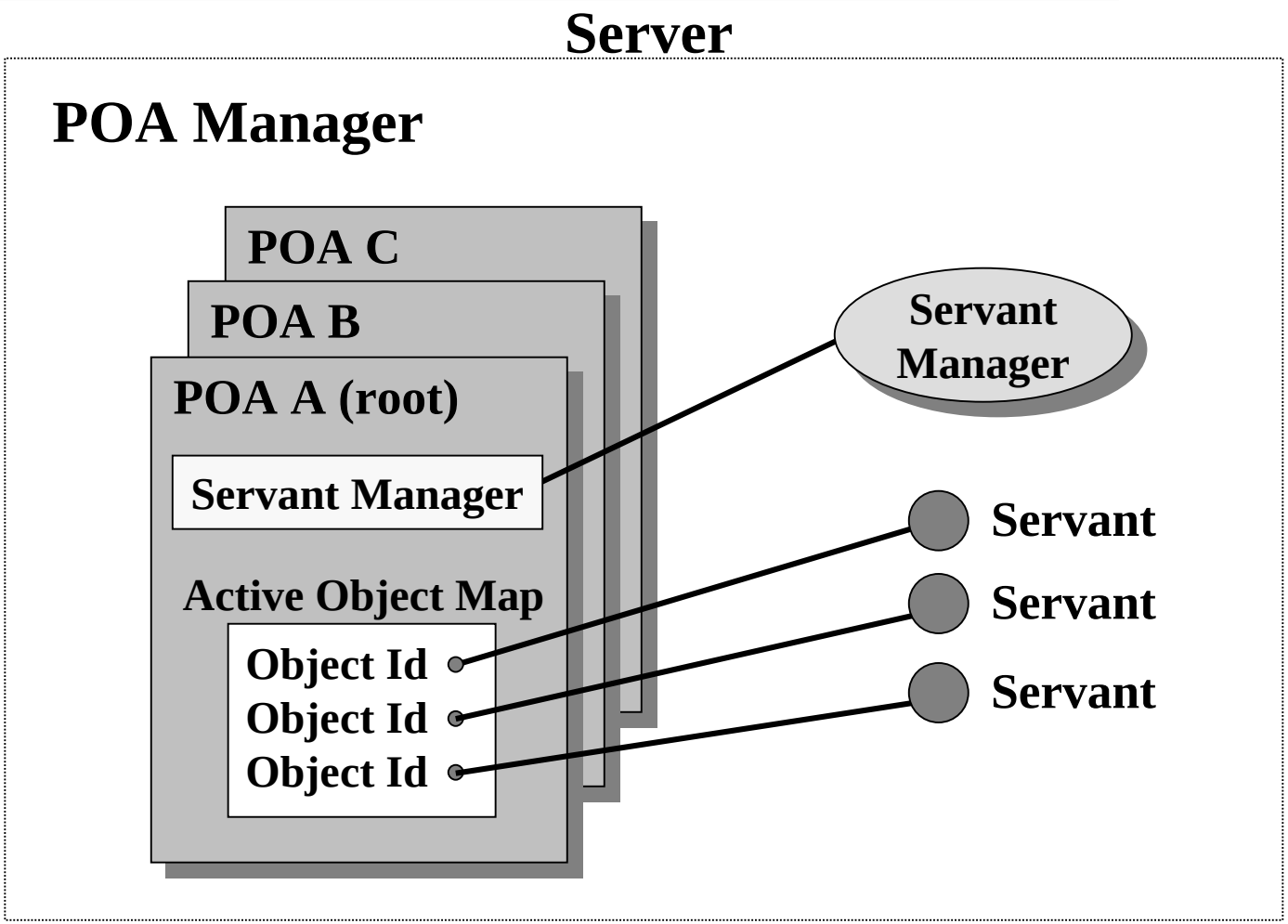
© 2002-2003 Carlos A. G. Ferraz

Portable Object Adapter (POA)

- Objetivos do POA
  - Portabilidade: permitir a construção de servidores realmente portáveis entre ORBs de diferentes fabricantes
  - Persistência: permitir a criação de objetos persistentes
  - Utilização de Políticas: permitir a associação de políticas aos servidores
  - Aninhamento de POAs: possibilitar a criação de múltiplas instâncias do POA no servidor

© 2002-2003 Carlos A. G. Ferraz

Arquitetura do POA



Componentes do POA

- Client
  - Um contexto computacional que realiza invocação de operações
- Server
  - Um contexto computacional que implementa um objeto ou mais objetos
- Servant
  - É um objeto de linguagem de programação que implementa pedidos em um ou mais objetos

© 2002-2003 Carlos A. G. Ferraz

Componentes do POA

- Object Id
  - Identifica um objeto CORBA dentro de um servidor. Pode ser fornecido pelo usuário ou pelo próprio POA
- Object reference
  - Estrutura de dados que identifica um objeto CORBA. Encapsula um Object Id e a identidade do POA que o controla

© 2002-2003 Carlos A. G. Ferraz

Componentes do POA

- POA
  - Entidade identificável no contexto de um servidor
  - Fornece um *namespace* para *object ids* para outros POAs
  - RootPOA pode ser obtido com resolve_initial_references
- Policy
  - É um objeto associado a um POA. Define características compartilhadas por todos os objetos implementados naquele POA

© 2002-2003 Carlos A. G. Ferraz

Componentes do POA

- POA manager
  - É um objeto que encapsula o processamento de estados de um ou mais POAs
- Servant manager
  - Um objeto que pode ser definido pelo usuário para permitir que o POA possa ativar e desativar servants. Existem dois tipos de Servant Managers
    - ServantActivator: para objetos persistentes
    - ServantLocator: para objetos transientes

© 2002-2003 Carlos A. G. Ferraz

Políticas do POA

- Políticas são utilizadas para controlar o ambiente de execução dos servants
- Os seguintes comportamentos podem ser gerenciados através de políticas:
  - Thread policy
    - ORB_CTRL_MODEL: O POA é responsável pelo gerenciamento de threads (default)
    - SINGLE_THREAD_MODEL: Apenas uma thread por servant

© 2002-2003 Carlos A. G. Ferraz

Políticas do POA (cont)

- Lifespan policy
  - TRANSIENT: objetos não podem “sobreviver” fora do POA que os criou (Default)
  - PERSISTENT: objetos podem “sobreviver” fora do POA que os criou
- Id assignment policy
  - USER_ID: a aplicação fornece Object Ids
  - SYSTEM_ID: o POA fornece os Object Ids (Default)

© 2002-2003 Carlos A. G. Ferraz

Usando o POA

© 2002-2003 Carlos A. G. Ferraz

Usando o POA

1. Obtendo o rootPOA
2. Definindo Políticas do POA
3. Criando um Novo POA
4. Ativando o POAManager
5. Criando Referência de Objetos

© 2002-2003 Carlos A. G. Ferraz

Usando o POA

1. Obtendo o rootPOA

- O primeiro passo para se usar o POA é conseguir a referência do primeiro POA, denominado rootPOA
- O rootPOA é gerenciado pelo ORB

```
ORB orb = ORB.init(args, null);
org.omg.CORBA.Object objPOA =
      orb.resolve_initial_references("RootPOA");
POA rootPOA = POAHelper.narrow(objPOA);
```


© 2002-2003 Carlos A. G. Ferraz

2. Definindo Políticas do POA

- O POA permite um amplo controle sobre as características do ambiente de execução dos Servants através das políticas

```
Policy[] tpolicy = new Policy[2];

tpolicy[0] = rootPOA.create_lifespan_policy
   (LifespanPolicyValue.TRANSIENT);

tpolicy[1] = rootPOA.create_thread_policy
   (ThreadPolicyValue.SINGLE_THREAD_MODEL);
```


© 2002-2003 Carlos A. G. Ferraz

3. Criando um Novo POA

- O desenvolvedor pode aplicar políticas específicas através da criação de um POA que forneça diferentes Servant Managers
- Um POA é criado como um “filho” de um POA que já exista através da operação **`create_POA`**
- Criar um novo POA indica a definição de um novo espaço de nomes para objetos, onde Object Ids são relativos a esse POA

© 2002-2003 Carlos A. G. Ferraz

3. Criando um Novo POA

- As seguintes informações devem ser utilizadas na criação de um POA:
  - O nome do POA: esse nome deve ser único com relação aos nomes de outros POAs pertencentes a um mesmo pai
  - Um POA Manager: um POA Manager deve ser associado ao novo POA; se for passado null, um POA Manager padrão será criado
  - Uma Lista de Políticas: uma lista com as políticas para controlar o comportamento do POA

© 2002-2003 Carlos A. G. Ferraz

3. Criando um Novo POA

```
POA persistentPOA =
    rootPOA.create_POA("childPOA", null, tpolicy);
```


© 2002-2003 Carlos A. G. Ferraz

4. Ativando o POA Manager

- Cada objeto POA tem um objeto **`POAManager`** associado para controlar os estados de execução do POA
- Um **`POAManager`** pode ser associado a um ou mais POAs
- Um **`POAManager`** possui os seguintes estados
  - Holding: há o enfileiramento de pedidos
  - Active: inicia o processamento dos pedidos
  - Discarding: descarta os pedidos seguintes
  - Inactive: rejeita os pedidos atuais e os seguintes

© 2002-2003 Carlos A. G. Ferraz

4. Ativando o POA Manager

```
// Ativando o POAManager de um persistentPOA.
// Sem esse passo, todas as chamadas para o
// servidor persistente ficariam pendentes,
// pois o POAManager está no estado “HOLD”.

  persistentPOA.the_POAManager().activate();
```


© 2002-2003 Carlos A. G. Ferraz

5. Criando Referência de Objetos

- Referência de objetos são criadas no lado servidor e depois exportadas para os clientes
- Nelas são encapsuladas as informações relativas aos objetos e ao POA que os controla

```
XXX Objservant = new XXX();
...
org.omg.CORBA.Object objRef =
  persistentPOA.servant_to_reference(Objservant);
```


© 2002-2003 Carlos A. G. Ferraz

POA: Conclusão

- O POA permite o controle “total” do ambiente de ambiente de execução do servidor
- É um padrão aceito e bastante completo para para realizar as tarefas complexas do adaptador adaptador de objetos CORBA
- Introduz um esforço de programação maior, mas mas garante portabilidade das aplicações

© 2002-2003 Carlos A. G. Ferraz