Interoperabilidade em CORBA
Protocolos GIOP e IIOP

Sumário

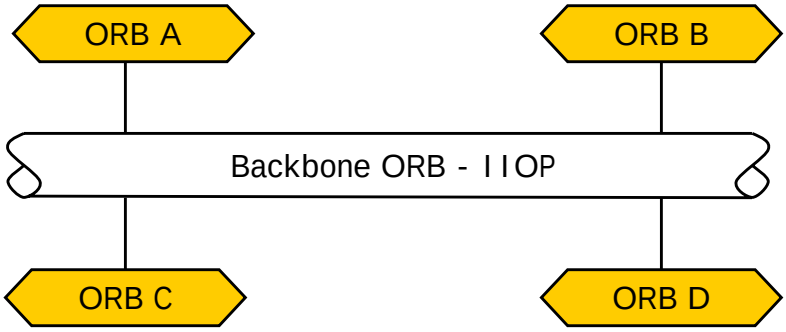
1 - Introdução
2 - Protocolo GIOP
3 - Protocolo IIOP
4 - Considerações Finais

1 - Introdução

- O CORBA é um padrão aberto que define a especificação da plataforma middleware, mas não se preocupa com aspectos tecnológicos, como por exemplo, detalhes de implementação.
  - ... desta forma, diferentes organizações podem implementar suas próprias versões do padrão CORBA, como o MICO;
  - ... desta possibilidade surge uma necessidade: as diferentes implementações do CORBA devem se comunicar!
- Vale a pena salientar que no desenvolvimento de aplicações distribuídas, usando CORBA, o desenvolvedor (programador) não se preocupa como os ORBs se comunicam.
- Contudo, é interessante conhecer os mecanismos quem possibilitam a comunicação entre aplicações desenvolvidas com diferentes ORBs.

... 1 - Introdução

- Os protocolos GIOP (General Inter-ORB Protocol) e IIOP (Internet Inter-ORB Protocol) são os responsáveis por fazerem a ponte entre as diferentes implementações do CORBA.



2 - Protocolo GIOP

- O GIOP especifica um conjunto de formatos de mensagens, representações de dados comuns para a comunicação entre os ORBs e realiza o mapeamento de requisições ORB para protocolos de transportes orientados à conexão.
  - independente de qualquer protocolo de comunicação, contudo, um candidato óbvio é o TCP (padrão de facto para a Internet)
- Na Especificação da OMG, três partes são definidas p/ o GIOP:
  - Representação de Dados Comum (CDR) - traduz tipos OMG-IDL para tipos de baixo-nível usados entre os agentes da redes.

... 2 - Protocolo GIOP

- Na Especificação da OMG, três partes são definidas p/ o GIOP:
  - Os formatos das mensagens GIOP - são sete as mensagens definidas pela OMG para o GIOP. Os formatos são os seguintes:

Mensagem	Origem	Valor
Request	Client	0
Replay	Server	1
CancelRequest	Client	2
LocateRequest	Client	3
LocateReplay	Server	4
CloseConnection	Server	5
MessageError	Both	6

... 2 - Protocolo GIOP

- As mensagens do GIOP são definidas em OMG-IDL como segue:

```
module GIOP{  // IDL
 enum MsgType{
  Request, Reply, CancelRequest,
  LocateRequest, LocateReply,
  CloseConnection, MessageError
 };

 struct MessageHeader{
  char magic[4];
  Version GIOP_version;
  boolean byte_order;
  octet message_type;
  unsigned long message_size;
 };
```

... 2 - Protocolo GIOP

- ... as mensagens do GIOP são definidas em OMG-IDL como segue:

```
struct RequestHeader {
  IOP::ServiceContextList service_context;
  unsigned long request_id;
  boolean response_expected;
  sequence object_key;
  string operation;
  Principal requesting_principal;
};

enum ReplyStatusType {
  NO_EXCEPTION,
  USER_EXCEPTION,
  SYSTEM_EXCEPTION,
 LOCATION_FORWARD
};
```

... 2 - Protocolo GIOP

- ... as mensagens do GIOP são definidas em OMG-IDL como segue:

```
struct ReplyHeader {
  IOP::ServiceContextList service_context;
  unsigned long request_id;
  ReplyStatusType reply_status;
};

struct CancelRequestHeader {
  unsigned long request_id;
};

struct LocateRequestHeader{
  unsigned long request_id;
  sequence object_key;
};
```

... 2 - Protocolo GIOP

- ... as mensagens do GIOP são definidas em OMG-IDL como segue:

```
enum LocateStatusType{
  UNKNOWN_OBJECT,
  OBJECT_HERE,
  OBJECT_FORWARD
};

struct LocateReplyHeader{
  unsigned long request_id;
  LocateStatusType locate_status;
};

};  // End of GIOP Module
```

... 2 - Protocolo GIOP

- Na Especificação da OMG, três partes são definidas p/ o GIOP:
  - Recomendações para o transporte da mensagem - o transporte de cada mensagem GIOP deve obedecer às seguintes recomendações:

1 - O transporte deve ser orientado à conexão;
2 - O transporte deve ser confiável;
3 - O transporte deve ser visto como uma *stream* de *bytes*;
4 - O transporte deve fornecer notificação de perda de conexão;
5 - O modelo de transporte para iniciar conexões deve ser mapeado no modelo de conexão geral do TCP/IP.

... 2 - Protocolo GIOP

- O GIOP também define um formato para IORs:
  - IOR (Interoperable Object Reference) é uma extensão do conceito de referência de objeto;
  - ... extrapola os limites de processos, máquinas e linguagens de programação e possibilita a identificação um objeto independente do ORB no qual ele fora criado.
  - ... uma vez que um componente do sistema obtém a IOR de um objeto, um proxy deste objeto pode ser criado.
  - No CORBA, o cliente interage diretamente com o proxy do objeto CORBA e toda a complexidade de comunicação é encapsulado pelo CORBA, e portanto, transparente para o desenvolvedor .

... 2 - Protocolo GIOP

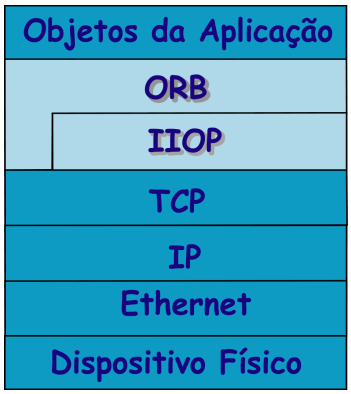
- A IOR de qualquer objeto CORBA pode ser convertida de uma referência de objeto para uma representação de string de caracteres (útil para transmissão pela rede) e vice-versa (recuperação da referência para definir um proxy para o objeto CORBA - cliente).

3 - Protocolo IIOP

- O IIOP especifica como as mensagens GIOP são trocadas sobre uma rede TCP/IP, ou seja, o IIOP mapea o GIOP para TCP/IP.
- O IIOP é uma especialização do GIOP, pois toda a sintaxe definida no GIOP é também aplicada no IIOP, ao passo que IIOP adiciona elementos específicos do TCP (endereços de redes e estabelecimentos de conexões).
- Desta forma, o IIOP possibilita utilizarmos a Internet como um *backbone* ORB, através do qual diferentes ORBs são conectados.

... 3 - Protocolo IIOP

- Para melhor entendermos o que é o IIOP e onde ele se posiciona na rede e em relação ao ORB core, tomemos o modelo OSI como referência e ilustramos o ORB e o IIOP nele.



... 3 - Protocolo IIOP

- O IIOP é construído sobre o TCP e acima do IIOP estará o ORB, que realiza o *marshaling* e o *unmarshaling* das requisições IIOP.
- Na camada de aplicação temos as implementações de objetos e outros objetos CORBA, tais quais serviços de nomes, eventos, etc.
- No entanto, salientamos que em ORBs não-CORBA, o IIOP não necessariamente está presente e que em algumas soluções não-padrão adotadas em implementações CORBA não se utilizam do IIOP. Ambos situações evidentemente comprometem a interoperabilidade.
- Usando IIOP, as referência de objetos são publicadas através de IIOP/IOR. Os IIOP/IOR tem a seguinte descrição em IDL:

... 3 - Protocolo IIOP

- Usando IIOP, as referência de objetos são publicadas através de IIOP/IOR. Os IIOP/IOR tem a seguinte descrição em IDL:

```
module IIOP{   // IDL
 struct Version {
  char major;
  char minor;
 };

 struct ProfileBody {
  Version iiop_version;
  string host;
  unsigned short port;
  sequence object_key;
 };
};  // End of IIOP Module
```

... 3 - Protocolo IIOP

- Descrição dos campos presentes na struct "ProfileBody"
  - onde *iiop_version* é a versão IIOP;
  - *host* é a máquina Internet alvo (as mensagens devem ser enviadas no formato IP - exemplo 143.106.6.1);
  - parâmetro *port* representa o número do porte onde o objeto destino escuta as requisições;
  - parâmetro *object_key* é fornecido pelo mesmo objeto que instanciou aquele IOR, e contém a referência do objeto para o qual a mensagem está para ser enviada.

... 3 - Protocolo IIOP

- Toda implementação CORBA-*compliant* (a partir da especificação CORBA 2.0) deve implementar IIOP nativo ou fornecer um *half-bridge* para ele.
- Desta forma, torna-se muito prático a criação de topologias flexíveis e/ou federações de ORBs, segmentando os ORBs em domínios de acordo com as necessidades administrativas:
  - ORBs de diferentes fabricantes, protocolos de redes, tráfego, tipos de serviços e segurança podem ser empregados em conjunto.

4 - Considerações Finais

- A área de objetos distribuídos está expandindo-se rapidamente, e os padrões abertos estão ocupando um papel importante nos dias atuais, vide o CORBA.
- A forma dos objetos distribuídos se comunicarem é igualmente importante: objetos que falam a mesma língua (IIOP) têm mais chances de sobrevivência.