

Introdução

Aspectos de comunicação do middleware

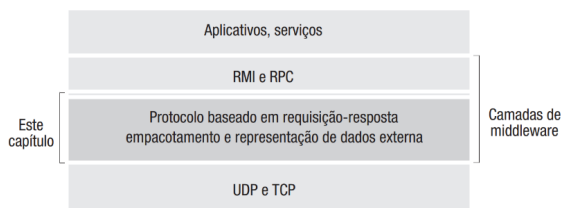


Figura 1 Camadas de middleware.

A API para protocolos Internet:

Características da comunicação

API que os protocolos de internet utilizam para comunicação entre os elementos de um SD.

- Comunicação usando send e receive;

O elemento que quer enviar uma mensagem executa o comando send e o elemento que quer receber executa o receive.

- Comunicação síncrona e assíncrona;
 - Síncrona: Send e receive causam bloqueio; Ambos são bloqueados para aguardar mensagens.
 - Assíncrona: Send não bloqueia. Receive normalmente bloqueia. Send não é bloqueado mas o receiver é bloqueado para aguardar a mensagem.
- Confiabilidade;

Fundamental nos SD:

validade: garantir que a mensagem enviada chegue ao destinatário.

integridade: garantir que chegue sem alterações.

- Ordenação;
Algumas aplicações exigem que a ordem das mensagens seja preservada no momento que são processadas pelo destinatário.
- Soquetes.
técnica de baixo nível para garantir a comunicação entre componentes de um sistema distribuído.

Representação externa de dados

Externa: dados que trafegam entre componentes completamente heterogêneos. Essa heterogeneidade é um problema corrigido com representação externa de dados.

O middleware pega as mensagens e converte para o padrão adequado para serem enviadas pela rede (**empacotamento**). Ao chegar no destinatário ocorre o processo inverso (**desempacotamento**).

- Heterogeneidade na apresentação de dados;
- Necessidade de um padrão comum;
- Empacotamento (marshaling);

- Desempacotamento (unmarshalling);
- Estratégias
Tecnologias que fazem representação externa de dados:
 - CDR de CORBA;
 - Serialização do Java;
 - XML em web services.

Representação comum de dados (CDR)

Mecanismo de representação do CORBA (middleware utilizado para garantir a implementação de sistemas de objetos distribuídos).

Permite representar todos os tipos de dados suportados.

Índice na sequência de bytes	← 4 bytes →	Observações sobre a representação
0–3	5	Comprimento do string
4–7	"Smit"	'Smith'
8–11	"h__"	
12–15	6	Comprimento do string
16–19	"Lond"	'London'
20–23	"on__"	
24–27	1984	unsigned long

Figura 2. Mensagem no CDR do CORBA.

Toda mensagem era dividida em grupos de 4 bytes. Para cada string antes era enviado o número da quantidade de string e em seguida a string dividida em grupos de quatro.

A mensagem convertida nesse formato é enviada pela rede e ao chegar é convertida novamente para

a forma como foi enviada inicialmente.

Empacotamento no CORBA

Linguagem usada para especificar ao middleware como esses dados devem ser convertidos para o padrão CDR.

IDL (Interface Definition Language) – Linguagem utilizada pelo CORBA para especificar os tipos de estruturas de dados e dos itens de dados básicos.

Ex: IDL da fig 2.

```
struct Person {  
    string name;  
    string place;  
    unsigned long year;  
}
```

Serialização de objeto Java

A serialização é o mecanismo de empacotamento de Java.

No Java RMI, tanto objetos como valores primitivos podem ser passados como argumento e resultados de invocações de um método.

Representação do struct person definido no IDL do CORBA em Java:

```
public class Person implements Serializable {  
    private String name;  
    private String place;  
    private int year;  
    public Person(String aName, String aPlace, int aYear) {  
        name = aName;  
        place = aPlace;  
        year = aYear;  
    }  
    // seguido dos métodos para acessar as variáveis de instância  
}
```

A classe implementa a interface serializable para converter os objetos

em um fluxo de bytes a serem enviados pela rede. Do outro lado o objeto é reconstruído.

XML (Extensible Markup Language)

Usada no webservice para elaborar documentos estruturados que consigam navegar pela web. Também usada em bancos de dados, definição de interfaces.

- Definida pelo W3C;
- Projetada para elaborar documentos estruturados para a WEB;
- É utilizada na comunicação com web services;
- Também utilizada no armazenamento, definição de interface e arquivos de configuração;
- É uma linguagem extensível. Novas tags podem ser definidas;
- SOAP é um protocolo que define tags XML que são publicadas para serem utilizadas pelos web services.

XML

```
<person id="123456789">
  <name>Smith</name>
  <place>London</place>
  <year>1984</year>
  <!-- a comment -->
</person >
```

XML usa uma estrutura de tags para delimitação.

XML – Components

- elementos
estruturas delimitadas por tags.
- atributos
valores que podem ser atribuídos para um elemento.
- nomes
nomes dados aos elementos (tags).
- dados binários
É possível inserir um dado binário dentro de uma tag representada por um texto.
- análise e documentos bem formados;
Padrão de tags com obrigatoriamente início e fim.
- CDATA
É utilizado dentro da tag entre colchetes os caracteres que poderiam ser processados como uma tag.
- prólogo XML
contém a versão do XML, a codificação utilizada nos caracteres, e o atributo standalone que define a vinculação do objeto XML.
- espaço de nomes
forma de definir uma tag e indicar escopo (onde a tag é válida.)

Aula 03 – Comunicação entre processos

- esquemas XML
Servem como documento onde é possível estruturar o formato de um documento bem formado XML.
- definições de tipo de documento;
Outra forma (linguagem) usada para escrever a estrutura de um XML. Na prática não tem sido utilizado.

Espaço de nomes

- exemplo:

```
<person pers:id="123456789" xmlns:pers = "http://www.cdk5.net/pers"
  <pers:name> Smith </pers:name>
  <pers:place> London </pers:place >
  <pers:year> 1984 </pers:year>
</person>
```

Antes de cada tag é definido o escopo ao qual aquele elemento pertence e atua.

Esquema XML

- exemplo:

```
<xsd:schema xmlns:xsd = URL das definições de esquema XML >
  <xsd:element name= "person" type ="personType"/>
  <xsd:complexType name="personType">
    <xsd:sequence>
      <xsd:element name = "name" type="xs:string"/>
      <xsd:element name = "place" type="xs:string"/>
      <xsd:element name = "year" type="xs:positiveInteger"/>
    </xsd:sequence>
    <xsd:attribute name= "id" type = "xs:positiveInteger"/>
  </xsd:complexType>
</xsd:schema>
```

define os nomes dos elementos e seus tipos, garantindo que a estrutura sempre seja obedecida por qualquer objeto XML que implemente

esse esquema. Serve de esqueleto para a análise do XML.

Referências a objetos remotos

Quando se trabalha com objetos remotos, para realizar a invocação desses objetos eles precisam ser identificados. Com essas identificações é feita a referência correta.

- A invocação de um método em um objeto remoto necessita de uma referência para identificar o objeto remoto;
- As referências de um objeto remoto devem garantir sua exclusividade no espaço tempo;
- Estratégia comumente utilizada.

32 bits	32 bits	32 bits	32 bits	
Endereço IP	Número de porta	Hora	Número do objeto	Interface do objeto remoto

Figura 3 Representação de uma referência de objeto remoto.

Para identificar um objeto remoto pode-se usar o endereço IP, a porta, a hora que a invocação está sendo realizada, o ID e a interface do objeto (métodos utilizados para invocação remota).

Essas informações unidas retornam o objeto com exclusividade.