

基于 SOAP 协议 MISC 与 SP 彩信业务订购接口的实现

MISC & SP MMS Subscription Interface Implementing Based on SOAP Protocol

刘咏亭 (中国科学院研究生院 北京 100049)

摘要:彩信是目前无线增值业务中增长迅速的一项业务,无线增值服务提供商(SP)通过与无线运营商(如中国移动)的合作,为用户提供彩信服务,即图片、声音等多媒体信息服务。用户订购彩信业务,需要在 SP 和中国移动之间提供数据、信息交互,完成业务订购、退订等操作。而 SP 与中国移动之间的接口,基于 SOAP 协议进行数据传输。本文分析双方 SOAP 通信接口定义,并探讨在数据交互过程中,如何进行 SOAP 消息封装、传输及解析,从而实现用户订购数据在 SP 和中国移动之间的数据同步。

关键词:SOAP XML 彩信

1 引言

MMS (Multimedia Messaging Service), 即多媒体信息业务,简称彩信,是按照国际上 3GPP 标准 (3GPP TS 23.140) 和 WAP 论坛标准 (WAP-206 和 WAP-209) 基础上开发的无线增值业务。SP (无线增值服务提供商) 通过无线网络运营商,为移动终端用户提供文本、图片、动画、声音等多媒体信息服务。

用户订购彩信业务,需要在 SP 和无线运营商之间提供数据、信息交互,完成业务订购、退订等操作。以中国移动为例,其 MISC (Mobile Information Service Center 移动信息服务中心) 平台负责支撑彩信数据业务模式,完成与 SP 订购关系同步等功能。MISC 遵循移动数据业务管理平台 (DSMP) 规范。

MISC 与 SP 的订购数据同步,基于 SOAP 协议,所有请求和响应消息均应采用 SOAP 格式的消息封装。接口通讯层与 HTTP 1.1 协议的 POST 和 GET 绑定在一起。按 DSMP 规范规定,服务器和客户端之间均通过 HTTP 的 POST 方法来进行交互。与此相关,HTTP 报文内容格式与 MIME 格式绑定,例如 Content-Type 为 text/xml 等均遵循 MIME 标准。

2 接口交互流程

一般情况下,用户发送短信 (SMS, Short Message Service) 指令订购 SP 的包月彩信业务,移动 MISC 接收

用户的订购请求后,利用接口通知 SP,SP 返回确认信息给 MISC,于是用户成功订购该业务。订购流程如图 1 所示。其中 ISMG (Internet Short Message Gateway) 是短信网关。MISC 与 SP 之间即是传递基于 SOAP 协议的 SOAP 消息 (Message), 如图中的 SyncOrderRelationReq 和 SyncOrderRelationResp。

其中,SyncOrderRelationReq 是 MISC 向 SP 发送的订购关系同步请求,SP 收到同步请求后,对这个订购请求做相应的订购关系处理,并返回订购关系同步应答 SyncOrderRelationResp。

3 SOAP 接口分析

根据 W3C 组织的定义,SOAP Version 1.2 (SOAP) 是一种轻量级的协议,用于在分散和分布环境下进行结构化信息的交换。它利用 XML 技术来定义可扩展的消息框架,以在多种承载协议上进行消息结构的交换。这种框架设计成对于任何软、硬件环境都是无关的,因此 SOAP 可以在不同的软硬件平台上实现。

3.1 SOAP 消息结构

SOAP 消息是一个 XML 文档,包括一个必需的 SOAP 封装,一个可选的 SOAP 头和一个必需的 SOAP 体。

- SOAP 封装 (Envelope) 是表示消息的顶层容器。
- SOAP 头 (Header): 应用 SOAP 交换信息的各

方是分散的且没有预先协定, SOAP 头提供了向 SOAP 消息中添加关于这条 SOAP 消息的某些要素 (feature) 的机制。SOAP 定义了少量的属性用来表明这项要素 (feature) 是否可选以及由谁来处理。

DSMP 规范相关定义的命名空间, 而所有无前缀的标记均处于此作用域内。同时, SOAP 封装标记均处于 SOAP 标准的命名空间作用域内。

在 DSMP 平台规范中, 以“订购关系同步请求包”

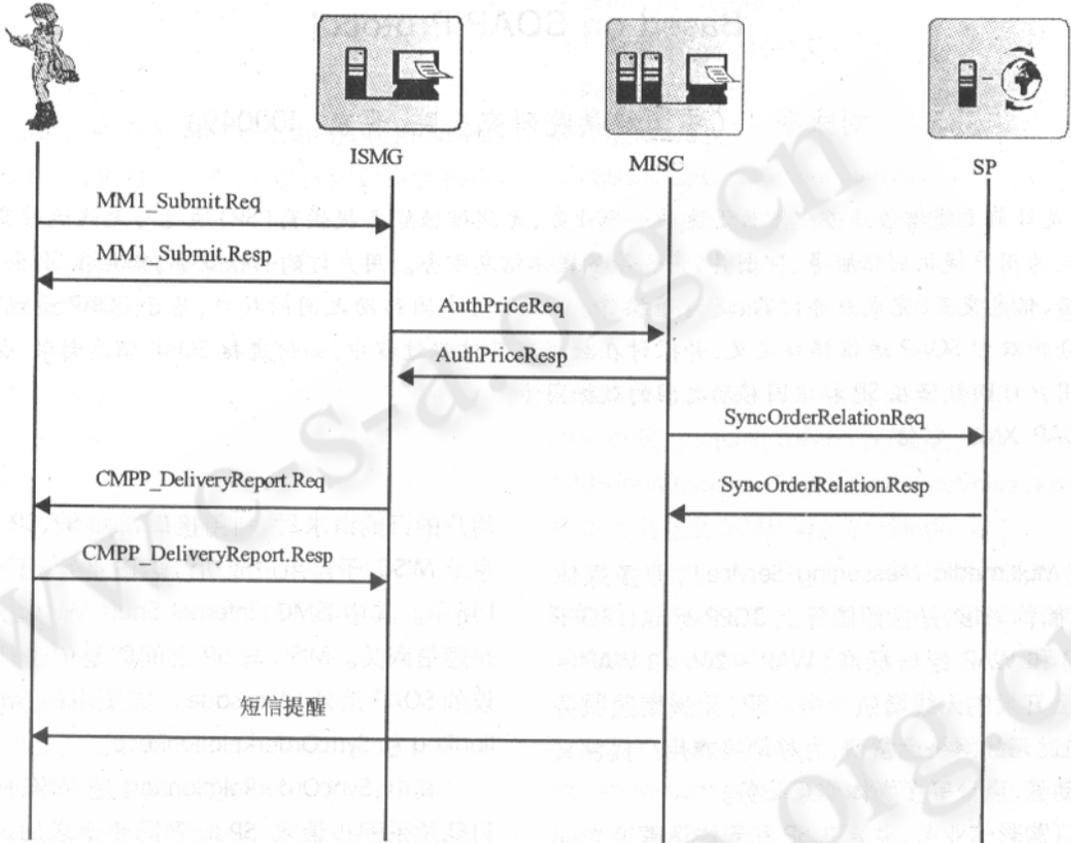


图 1 通过 SMS 通道订购流程

• SOAP 体 (Body) 是包含消息的最终接收者想要的信息的容器。

如下图所示, 一条 SOAP 消息可以在 Header 和 Body 容器中有多个应用程序定义块。

一个 SOAP 应用程序产生的消息中, 所有由 SOAP 定义的元素和属性中必须包括正确的命名空间 (Name Spaces)。SOAP 应用程序必须能够处理它接收到的消息中的 SOAP 命名空间。SOAP 定义了两个命名空间:

SOAP 封装的命名空间标志符是 "http://schemas.xmlsoap.org/soap/envelope/"

SOAP 的编码规则的命名空间标志符是 "http://schemas.xmlsoap.org/soap/encoding/"

在本接口中默认的命名空间为 URI: http://www.monternet.com/dsmp/schemas/, 即包含了中国移动

SyncOrderRelationReq 为例, 其数据类型定义如下:

```

< element name = " SyncOrderRelationReq" type = "
dsmp:SyncOrderRelationReqType" / >
    < complexType name = " SyncOrderRelationReq-
Type" >
        < sequence >
            < element name = " MsgType" type = "
string" / >
            < element name = " Version" type = "
string" / >
            .....
            < element name = " DestUser_ID" type = "
dsmp:user_id_schema" / > //订购用户手机号
            < element name = " ActionID" type = " Inte-
    
```

```

ger" / > //服务动作代码,其中 1 为开通;2 为停止
  < element name = " ActionReasonID" type
= " integer" / >

```

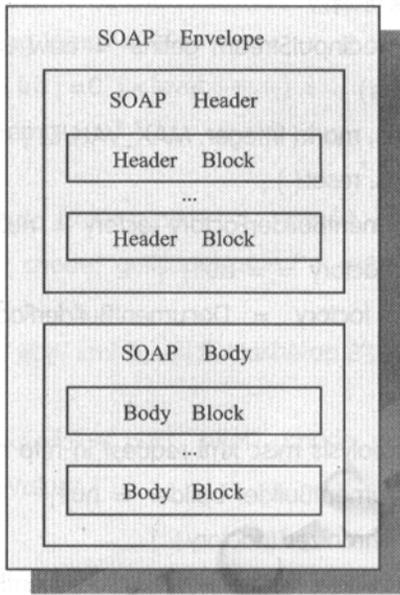


图 2 SOAP 消息结构

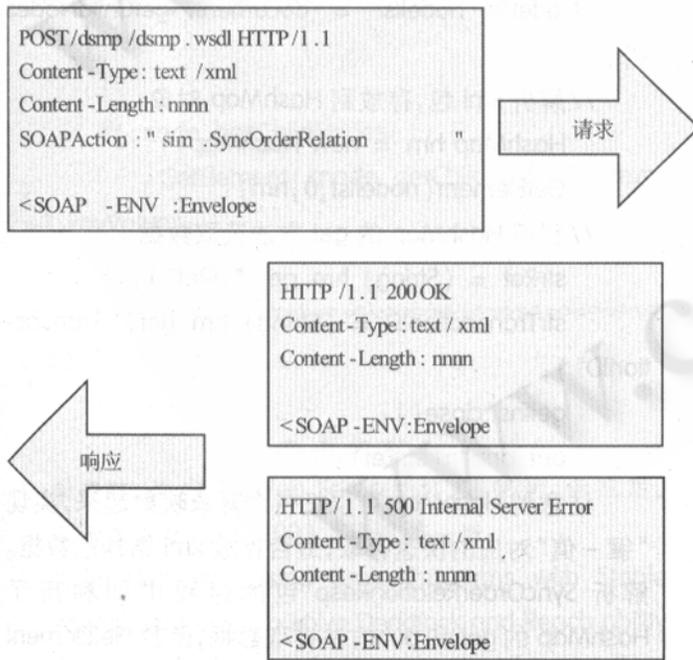


图 3 SOAP HTTP 绑定

```

< element name = " SPID" type = " string" mi-
nOccurs = " 0" / > //提供该服务的 SP 企业代码
  < element name = " SPServiceID" type = "

```

```

string" / > // SP 提供该服务的服务代码
.....
</sequence >
</complexType >

```

而“订购关系同步应答包”SyncOrderRelationResp, 其数据类型定义如下:

```

< element name = " SyncOrderRelationResp" type
= " dsmp:SyncOrderRelationResp" / >
< complexType name = " SyncOrderRelationResp-
Type" >
  < sequence >
    < element name = " MsgType" type = "
string" / >
    < element name = " Version" type = "
string" / >
    < element name = " hRet" type = " integer" /
> //返回值,其中 0 为成功,表示接收及解析 xml 包成
功
  </sequence >
</complexType >

```

注意 hRet 元素,由 SP 返回响应给 MISC,标志此次同步是否成功。

3.2 协议绑定

由于 HTTP 的广泛使用,SOAP 规范中规范化了一种用于 HTTP 的协议绑定。HTTP 协议绑定定义了 在 HTTP 上使用 SOAP 的规则。SOAP 请求/响应自然地映射到 HTTP 请求/响应模型。图 3 说明了 SOAP HTTP 绑定的很多细节。

HTTP 请求和响应消息的 Content - Type 标头都必须设为 text/xml (在 SOAP 1.2 中是 application/soap + xml)。对于请求消息,它必须使用 POST 作为动词,而 URI 应该识别 SOAP 处理器。对于 HTTP 响应,如果没有发生任何错误,它应该使用 200 状态码,如果包含 SOAP 错误,则应使用 500。

4 JAVA 环境实现过程

先介绍 MISC 端程序。即 MISC 向 SP 发送订购关系同步请求 SyncOrderRelationReq,接收响应包 SyncOrderRelationResp,并进行解析。具体实现如下:

(1) 向 SP 发送订购关系同步请求 SyncOrderRela-

tionReq, 并接收响应包。

//发送请求,接收响应

```
InputStream ins = getURLContent( ( new StringBuffer() ). append( strSOAP ). toString() );
```

其中, strSOAP 即是上述的 SOAP 包 SyncOrderRelationReq, 定义为:

```
String strSOAPENV = " <? xml version = 1.0 encoding = UTF-8? > " +
```

```
" < SOAP - ENV: Envelope xmlns: xsi = " http://www.w3.org/2001/XMLSchema-instance" +
```

.....

.....

而过程 getURLContent 定义如下:

```
public synchronized InputStream getURLContent( String strSoap ) {
```

```
try {
```

```
URL url = new URL( m_strURL );
```

```
URLConnection urlc = ( HttpURLConnection ) url.openConnection();
```

```
urlc.setRequestMethod( " POST " );
```

```
//设置 HttpURLConnection 对象
```

```
urlc.setDoOutput( true );
```

```
urlc.setRequestProperty( " Content - Type", " text/xml; charset = utf - 8 " );
```

```
//发送 xml 信息
```

```
PrintStream objURLOutput = new PrintStream( urlc.getOutputStream() );
```

```
objURLOutput.println( strSoap );
```

```
objURLOutput.flush();
```

```
objURLOutput.close();
```

```
//取得返回信息,返回 xml 对象
```

```
return urlc.getInputStream();
```

.....

.....

java.net 包中的类和接口提供了可用于高层网络编程的 API。高层的 API(如 URL,URLConnection 和 HttpURLConnection 等类)可以更快进行网络应用开发,代码实现简单。

使用 URL 类解析输入的 URL 并做一些处理底层工作,使用 openConnection 来建立到 URL 的连接。该方

法返回一个 HttpURLConnection 对象,与 URL 进行诸如读、写、查询等通信工作。

(2) 对响应包 SyncOrderRelationResp 进行解析。

//解析响应

```
BufferedInputStream getins = new BufferedInputStream( ins );
```

```
getins.mark( Integer.MAX_VALUE );
```

```
getins.reset();
```

```
DocumentBuilderFactory factory = null;
```

```
if( factory == null ) {
```

```
factory = DocumentBuilderFactory.newInstance();
```

```
}
```

```
//analysis misc xml request in http header
```

```
DocumentBuilder builder = null;
```

```
synchronized( factory ) {
```

```
builder = factory.newDocumentBuilder();
```

```
}
```

```
Document document = builder.parse( getins );
```

```
NodeList nodelist = document.getChildNodes
```

```
( );
```

```
//解析 xml 包,存放到 HashMap 对象
```

```
HashMap hm = new HashMap();
```

```
GetElement( nodelist, 0, hm );
```

```
//利用 HashMap 的 get 方法读取数据
```

```
strRet = (String) hm.get( " hRet " );
```

```
strTransactionID = (String) hm.get( " TransactionID " );
```

```
getins.close();
```

```
out.println( strRet );
```

注意到 HashMap 可以把各个对象映射起来,实现“键-值”对应的快速存取,适合存放 xml 结构的数据。解析 SyncOrderRelationResp 包的过程中即利用了 HashMap 的 get 和 put 方法存取数据,函数 GetElement 定义如下。其中, hRet 等元素名称可以在上文 SyncOrderRelationResp 包的数据类型定义中查到。

```
void GetElement( NodeList nodelist, int level, HashMap keyValue ) throws Exception {
```

```
Node cnode;
```

```
if( nodelist.getLength() == 0 ) {
```

```

return;
}
for( int i=0; i < nodelist.getLength(); i++ ) {
    cnode = nodelist.item(i);
    String temp = "";
    for( int j=0; j < level; j++ ) {
        temp += " \t" ;
    }
    if( cnode.getFirstChild() != null ) {
        if ( cnode.getNodeName().equalsIgnoreCase(" hRet" )) {
            //解析 xml 包,利用 HashMap 的 put 方法存放
            keyValue.put( " hRet" , cnode.getFirstChild().getNodeValue());
        }
        if( cnode.getNodeName().equalsIgnoreCase(" TransactionID" )) {
            keyValue.put( " TransactionID" , cnode.getFirstChild().getNodeValue());
        }
    }
    if( cnode.hasChildNodes() ) {
        GetElement( cnode.getChildNodes(), level+1, keyValue);
    }
}

```

再介绍 SP 端程序。即 SP 接收 MISC 发送的订购关系同步请求 SyncOrderRelationReq, 解析后发送接收响应包 SyncOrderRelationResp。由于程序实现过程与 MISC 端程序类似, 不再具体说明。

5 总结

在无线增值服务领域, SP 与中国移动之间提供数据接口, 基于 SOAP 协议进行数据传输。在结合实例分析了 SOAP 消息结构和 HTTP 协议绑定等原理后, 在 JAVA 平台上实现了双方数据传输的 SOAP 接口, 详细探讨在数据交互过程中, 如何进行 SOAP 消息封装、传输及解析, 从而实现用户订购数据在 SP 和中国移动之间的数据同步。

参考文献

- 1 3GPP TS 23.140 V6.0.0 Multimedia Messaging Service - Stage 2, (Release 5)
- 2 WAP-209-MMSEncapsulation-20010601-a WAP MMS Encapsulation Protocol
- 3 中国移动多媒体消息系统(MMS)设备规范, 版本号:1.0.2, 中国移动通信集团公司.
- 4 卓望数码技术(深圳)有限公司, MISC1.6 SP 订购通知接口要求, 2005 年 01 月.