基于XML的电子商务应用体系构建

廖俊松　汤宏斌　张金隆　苏春园　（武汉华中科技大学管理学院　430074）

摘要　本文对XML语言及其应用于电子商务的技术特点进行了分析，在此基础上以银行和证券公司之间的转账系统为例对XML技术在具体电子商务应用体系的构建进行了研究和探讨。

关键词　XML　HTML　电子商务

1　XML技术

XML即扩展标识语言(eXtensible Markup Language)，是互联网联合组织（W3C）创建的一组规范，以便与软件开发人员和内容创作者在网页上组织信息，其目的不仅在于满足不断增长的网络需求，同时还希望借此能够确保在通过网络进行交换合作时，具有良好的可靠性和互操作性。

XML是SGML（Standard Generalize Markup Language）标准通用标识语言的子集，保存着SGML可扩展性、结构化和有效性等特点。与HTML不同，XML用来描绘结构化数据，而HTML用来显示内容。XML包括以下几个方面内容：DTD（Document Type Definition,文档类型定义），XSL（eXtensible Stylesheet Language,可扩展样式语言）和XLL（eXtensible Link Language,可扩展链接语言）等。

DTD规定了XML文件的逻辑结构，定义了XML文件中的元素、元素的属性以及元素与元素之间的关系。它可以帮助XML的分析程序校验XML文件标识的合法性；XSL是用于规定XML文档样式的语言，它能在客户端使Web浏览器改变文档的表示法，从而不需要再与服务器进行交互；XLL将进一步扩展目前Web已有的简单链接。

XML提供在应用程序和系统之间传输结构化数据的方法。像客户信息、信用卡交易、定单和完成请求这类数据能够转换成XML可以用来在Web服务器、浏览器之间、贸易伙伴之间传输数据。因此XML非常适合于电子商务。

2　XML应用于电子商务的技术特点

XML的优势主要体现在其描述数据的特点上。当HTML已无法担当领军电子商务的重任时，XML出现就在于其是用来描述数据而不是来显示数据的，这是XML和HTML的显著区别。XML的文件结构嵌套可以复杂到任意程度，能表示面向对象的等级层次，XML对于数据的描述性，以及其语言数据结构化的特点，在电子商务应用中具有巨大的优势。

（1）易于与数据库相连接：在互联网上，在基于Internet商务应用上，存在大量数据需要处理分析，最好的方法是把这些数据放到数据库中，所以几乎所有大型的商业应用系统都是和数据库相关联的，所以如果XML需要在商业领域大展宏图的话，也必须要和数据库相联系。而HTML主要是用来数据的显示，它不利于数据的格式化、结构化。我们可以把整个XML看成是一个数据库系统，XML文本本身可以看成是数据库中的数据区，DTD或者Schemas可以看成是数据库模式设计，XQL可以看成是数据库查询语言，SAX或DOM可以看成是数据库处理工具。XML本身不能和数据库挂上钩，但是加上一些其他的辅助工具，我们是能够做到这一点的。XML的这一优点是构建基于XML电子商务应用系统的关键。

（2）可扩展性方面：HTML不允许用户自行定义他们自己的标识或属性，而在XML中，用户能够根据需要，自行定义新的标识及属性名，以便更好地从语义上修饰数据。XML有利于数据交换和传递的特性将为电子商务，尤其是B2B带来冲击。各种专门的行业可以自行定义行业内信息存储的格式，从而有利于行业内各企业的内容定义与信息交换。

(3) 对EDI的支持：XML所采用的标准技术最适合Web开发，应用于Internet EDI，则可以得到真正Web风格的EDI——XML / EDI。XML支持结构化的数据，可以更详细地定义某个数据对象的数据结构，例如，描述产品或者详细定义该产品的生产厂、产品名、产品号、产地等信息。这种定义不仅为标记该产品提供方便，而且这种XML数据很容易按生产厂、产品名等排序，使用户的查询变得更加方便。如果出现某些商业规则例外，例如销售商想在订购单中增加注释，只要采用XML，销售商就可以在指定的数据放入文档后加入注释，解决了以前固定格式EDI的困难。

XML的另一个好处是大大降低了数据管理和交换的成本。XML的强大之处就在于它具有一套统一的数据格式，这种统一的数据格式可以使数据管理和交换的成本更低，也更易于管理。

3 XML的电子商务应用设计

本节将通过证券和银行之间转帐系统的设计和实现这个具体的实例来展现XML在构建电子商务体系中的应用。

为了方便股民在股票资金帐户和银行帐户之间方便的进行资金的划拨，需要设计一个应用层协议来实现这个功能。一般都是基于TCP/IP协议设计高层的应用协议，并通过特定的应用程序实现这个功能。这里设计了一个通过WWW服务器之间的通信实现转帐的协议体系结构，能够保证一个证券公司同时可以和多个银行进行交互，而同时，一个银行也可以与多个证券公司进行交互。协议的设计是基于XML的标记模型（元素/属性）而建立的。目的之一是建立一个和实现无关的通信协议。每一个消息由一个有效的XML文档组成，在通信的两端（银行和证券）通过发送和接收文档来进行交互。实际上该协议是建立在HTTP基础上，是通过HTTP来进行传输的。

3.1 协议设计总体说明

设计的协议允许用户既可以在银行，也可以在证券公司办理转帐功能。只要这两个机构已经连网并且用户在这两个机构都拥有自己的帐号。实际上，比如在银行端,银行会保存和该用户活期存款帐号相关的所有信息，如果用户需要在银行端办理转帐功能，只要用户能够提供银行活期存款帐号的号码和正确的密码，同时又能够提供证券公司的资金帐号和密码，系统首先会在本地查找该活期存款帐号是否存在，密码是否相符，如果是的话，会判断该活期存款帐号是否已经和用户提供的证券公司的资金帐号相关联，如果已经相关联的话，就返回一个错误信息，表示用户已经办理过该业务。如果还没有相关联的话，银行通信服务器就会和证券公司的相关的通信服务器相联系，如果证券公司返回信息认为用户提供的资金帐号和密码都是正确的话，这样就在银行和证券公司的相关的数据库中分别加上标志，来表明某个特定的资金帐号已经和某个特定的活期存款帐号相关联。这样，转帐功能就建立起来了。当用户已经办理了转帐功能以后，事实上，用户可以自己通过浏览器（需要支持XML格式）自行进行资金帐户的划拨。在划拨的时候需要注意的是，只有帐户中的余额大于要划拨的款项的时候才能进行划拨。在满足这个条件的情况下，用户既可以把活期存款中的部分款项划拨到资金帐户中，也可以把资金帐户中的款项划拨到活期存款中。大大方便了用户资金的调度和操作。

3.2 协议设计细节描述

下面具体描述协议的设计和相关的实现的细节：

首先我们来看整个通信协议的总的体系结构：

```
< !ELEMENT 信息负载 （信息头，(请求信息|响应信息)) >
< !ATTLIST 信息负载
版本号 CDATA #REQUIRED
时间标签 CDATA #REQUIRED
>
```

一条消息主要有三个元素组成：信息负载，信息头和请求信息或者是响应信息。信息负载是对这个数据包属性的描述，信息头用来表示信息的发送者和接收者，请求信息或者是响应信息是消息的主体内容。

信息负载包括两个属性：

版本号：用来表示通信所使用的版本，必须保证通信双方使用相同的版本号。这里我们设置为1.0。

时间标签：用来表示该数据包发出的时间。实际时间标签的格式我们设定为：YYYY：MM：DDTHH:MI:SS,其中T是日期和时间的一个分隔标志符。比如：2000:03:01T10:22:33就是一个有效的时间标签。

下面是关于信息头的定义：

```
<!ELEMENT 信息头(信息发送者,信息接收者)>
```

信息头主要有两部分组成，信息的发送者和信息的接收者。用来表示消息的发出方和消息的接收方。消息发送发出方和接收方拥有相同的属性。它们的定义格式如下：

```
< !ELEMENT 信息发送者 EMPTY >
< !ATTLIST 信息发送者
名称 CDATA #REQUIRED
发送者ID CDATA #REQUIRED
```

角色（银行|证券）#REQUIRED

>

< !ELEMENT 信息接收者 EMPTY >

< !ATTLIST 信息接收者

名称 CDATA #REQUIRED

接收者ID CDATA #REQUIRED

角色（银行|证券）#REQUIRED

>

它们都拥有三个属性：其中名称是指消息发送方或者是接收方的名称，比如：工商银行武汉市分行就是一个有效的发送方或者是接收方的名称。接收者ID是用来唯一的标识一个发送方或者是接收方，在实现的时候有两种机制。一种是简单的用URL作为对发送方或者是接收方的唯一标识。另外，可以对每一个发送方或者是接收方建立一个UUID，保证唯一的标识一个发送方或者接收方。角色是指消息发送或接收者的身份，是银行还是证券公司。

3.3 请求信息协议格式

下面我们来介绍信息的主体部分，首先我们介绍请求信息，以下是请求信息的定义：

<!ELEMENT 请求信息 %交易形式>

<!ATTLIST 请求信息

请求信息ID CDATA #REQUIRED

>

< !ENTITY %交易形式 "转帐开户+|转帐|交易查询+|交易管理|>"

请求信息主要有转帐开户，转帐，交易查询和交易管理这四大块，这是和转帐业务相关的主要功能。请求信息只包括一个属性，就是请求信息的ID号，请求信息ID号在服务器A到服务器B的所有的数据包中必须是唯一的，即每一个数据包必须要有一个唯一的ID号。在一个数据包里面可以同时递交几个转帐开户申请，这就给实现带来了一定的灵活性。服务器可以迅速处理每一个用户的申请，也可以根据具体情况批量的进行处理。同时，服务器也可以同时递交多个交易查询。

下面将以转帐开户和转帐这两块为例介绍请求信息协议格式：

转帐开户交易格式

<!ELEMENT 转帐开户 数据组 >

<!ATTLIST 转帐开户

开户消息ID CDATA #REQUIRED

交易代码 CDATA #REQUIRED

交易提出（股民|储户|银行系统|证券系统）#REQUIRED

>

转帐开户元素有三个属性：开户消息ID保证在所有一次递交的所有的开户申请业务中，每一个都有一个唯一的ID号。即必须保证该ID在一个包范围内的唯一性。交易代码是为了使机器能够更加方便的处理该业务。交易提出指明该业务是由谁触发的，是股民、储户、银行端的服务器系统还是证券端的服务器系统。

转帐业务包含了一个数据组，关于数据组的内容在下面还有详细的描述。这里，简单说明一下，数据组主要是用来表示在服务器之间进行和交易相关的业务数据的传递，比如资金帐号、交易金额等数据。

转帐交易格式

<!ELEMENT 转帐（资金帐户转出活期帐户转入+|

活期帐户转出资金帐户转入+|

交易结果查询+）>

<!ELEMENT 资金帐户转出活期帐户转入 数据组>

<!ATTLIST 资金帐户转出活期帐户转入

资金帐户转出活期帐户转入消息ID CDATA #REQUIRED

交易代码 CDATA #REQUIRED

交易提出（股民|储户）#REQUIRED

>

<!ELEMENT 活期帐户转出资金帐户转入 数据组>

<!ATTLIST 活期帐户转出资金帐户转入

活期帐户转出资金帐户转入消息ID CDATA #REQUIRED

交易代码 CDATA #REQUIRED

交易提出（股民|储户）#REQUIRED

>

<!ELEMENT 交易结果查询（相关消息状态+）>

<!ATTLIST 交易结果查询

交易结果查询消息ID CDATA #REQUIRED

交易代码 CDATA #REQUIRED

交易提出（银行系统|证券系统）#REQUIRED

>

<!ELEMENT 相关消息状态 数据组>

<!ATTLIST 相关消息状态

请求消息的ID CDATA #REQUIRED

要查询的消息的ID CDATA #REQUIRED

>

转帐交易是业务的核心，它实际上是处理在银行和证券之间的用户的资金的划拨，主要有两种模式，资金帐户转出活期帐户转入和活期帐户转

出资金帐户转入。

我们来分析资金帐户转出活期帐户转入这种情况，当股民或者是储户提出转帐请求时，考虑一般性，我们设定是股民提出转帐请求，证券公司端的服务器将分析股民要求转帐的金额是否小于资金帐户的余额，如果成立的话，减少该资金帐户的余额，然后该数据包就发送到银行的服务器，银行的服务器如果能正确处理该条消息的话，就增加相关的活期存款的余额，并把操作成功的结果返回给证券方的服务器。

但是有可能证券服务器在发出了这个交易命令后收不到应答信息（可能是包丢失的原因）。正是因为这个原因，所以需要一个交易结果的查询。交易结果的查询可以同时查询多条已经发出但没有及时应答的那些消息。在相关消息状态元素中的那个属性：要查询的消息的ID就是没有得到响应的消息的ID号。请求消息的ID这个属性表示和该请求相关的请求包的ID号。这两个属性的组合唯一的决定了一条消息。比如证券服务器方发出了三条资金帐户转出活期帐户转入消息，但很久没有响应，就可以发出交易结果查询这个指令来向银行端服务器查该三条消息的执行情况

3.4 响应信息协议格式

证券方和银行方都可以向对方的服务器发出请求，然后接收请求的服务器在处理了该请求以后，向发出请求的服务器返回响应消息。

协议对请求的响应的协议格式定义：

```
<!ELEMENT 响应信息(响应应答+)>
<!ATTLIST 响应信息
响应信息ID CDATA #REQUIRED
>
<!ELEMENT 响应应答 数据组*>
<!ATTLIST 响应应答
应答代码 CDATA #REQUIRED
应答代码描述 CDATA #REQUIRED
请求消息的ID CDATA #IMPLIED
相关消息的ID CDATA #IMPLIED
>
```

每一个响应消息包都有一个响应消息ID用来唯一的标识该响应数据包。一个响应消息中可以包含多个响应应答。因为在一个请求包中可能包含多个请求，比如在一个要求转帐开户的数据包中可能要求同时对几个用户进行转帐功能的建立。所以需要同时对这几个请求消息进行答复。其中在响应应答元素中的应答代码，我们采用三位的数字字符。比如200表示消息成功处理。301表示活期帐号不存在等等。请求消息的ID就是那个请求数据包的唯一的标识号，而相关消息的ID就是指该数据包中的某一个特定的请求内容，比如请求对一笔资金帐户转出活期帐户转入的查询。其中数据组是指对该该请求相关的需要返回的数据项的一个集合。

3.5 数据组描述

下面我们定义了数据组和数据项的格式：

```
<!-- 数据组描述 -->
<! ELEMENT 数据组 (数据项)+ >
<!-- 数据项描述 -->
< ! ELEMENT 数据项 EMPTY >
< ! ATTLIST 数据项
数据元名称 CDATA #REQUIRED
类型 CDATA #REQUIRED
长度 CDATA #REQUIRED
是否固定长度 （是 | 否）#REQUIRED
数据元内容 CDATA #REQUIRED
>
```

数据组由数据项组成，根据交易的特定，不同的交易请求包含不同的数据组，而对不同的交易的响应也包括不同的数据组。每一个数据项都定义了该数据项的名称，类型，长度，是否固定长度和数据项的实际的内容。这里常用到的数据项有资金帐号、活期帐号等。

4 结束语

XML技术易于掌握和容易扩展，它正在作为一种“较好的，具有可移植能力”的数据表示得到应用和推广。基于XML技术的电子商务解决方案降低了企业在IT基础设施上的开销，节约了经营成本。这种应用利用XML数据表示简化了应用互操作的复杂性，提高了数据交换的效率。■

参考文献

1 *肖菁、商卫东，XML——新一代Web标记语言，电脑与信息技术，1999(3)。*

2 *董 欣、陈晓鸥，XML文件的显示与浏览，计算机应用，2000,20(8):2–4。*

3 史彤毅、杨利，XML在电子商务中的应用，ebTimes.com。