

一种基于 XML 的数据交换中间件技术*)

李双庆 游 莲 古 平 程代杰

(重庆大学计算机科学与工程学院 重庆400044)

A Data Exchange Middleware Technology based on XML

LI Shuang-Qing YOU Lian GU Ping CHENG Dai-Jie

(College of Computer Science and Engineering, Chongqing University, Chongqing 400044)

Abstract With the development of electronic commerce, the requirements of the data exchanging and interoperability among different EC systems keep growing. This paper analyzes the data exchange modes among different systems, proposes a XML-based data exchange middleware model, which focuses on solving the problems in data exchanging among heterogeneous EC systems. The key technologies in data exchange middleware model are studied, which include exchange service interface, ontology, data exchange engine and template. Lastly, the XSLT is extended according to the data exchange requirements.

Keywords Middleware, XML, Data exchange, Electronic commerce

1 引言

电子商务作为一种新的商务模式在近几年得到迅速的发展。早期的电子商务模式通常以一个孤立的企业网站为中心,不能支持商务参与者之间对等的信息交流。随着企业信息系统和电子商务系统的广泛应用,电子商务正向企业间供应链整合的模式发展。在新的模式下,若干商务实体共同构成一条完整的供应链,代表每个商务实体的企业信息系统或电子商务系统成为供应链上的某个节点,各个系统之间构成对等的协作关系,能够相互交换商务数据,并直接触发对方系统的相关操作。

要实现这种多个异构系统之间的信息交互,已不是传统意义上的动态页面所能达到的,系统需要实现更加复杂的商务逻辑,并解决一些关键性的技术问题。目前,应用服务器平台提供了运行复杂商务逻辑所需的环境,并通过 CORBA 和 J2EE 等分布式对象技术实现了分布环境下对象互操作;XML 语言^[1]通过良好的数据描述能力,能够支持不同系统之间数据的传递。但是由于系统的多样性,不同系统所交互的数据尽管存在很大的相关性,但也存在不少差异。如果针对每一对系统来实现数据交互功能,开发工作量将十分巨大,并且缺乏可维护性。因此,需要一种能整合不同应用系统数据并提供数据方便交互能力的中间系统来解决该问题,这正是本文研究的动因。

本文将分析异构应用系统间数据交换的模式和需要解决的关键问题,然后提出一种基于 XML 的电子商务数据交换中间件模型,研究电子商务应用中异构系统间数据交互的关键技术和实现方法。

2 数据交换模式

不同应用系统对系统数据格式的定义通常是不同的,数

据所针对的目标也不相同,同时,因为没有统一的数据交互标准,因此不同应用系统的数据存在很大的差异。当异构的应用系统之间需要进行数据交换时,必须定制相互的数据转换操作才能支持实现数据交换。这种操作可以由其中一方完成,或者由数据交换的双方共同完成。在电子商务的供应链整合应用中,存在若干系统之间需要相互交换数据共同完成某个商业目标,这种情况就变得十分复杂。下面是异构系统之间数据交换模式的形式化描述。

定义1 $\Psi(\lambda \rightarrow \xi)$ 表示为数据 λ 经过变换 Ψ 转换为数据 ξ 。这是一种一对一的变换关系。

定义2 $\Psi(\lambda_1, \lambda_2, \dots, \lambda_n \rightarrow \xi)$ 表示为数据 $\lambda_1, \lambda_2, \dots, \lambda_n$ 经过变换 Ψ 转换为数据 ξ 。这是一种多对一的变换关系。

定义3 $\Psi(\lambda \rightarrow \xi_1, \xi_2, \dots, \xi_n)$ 表示为数据 λ 经过变换 Ψ 转换为数据 $\xi_1, \xi_2, \dots, \xi_n$ 。这是一种一对多的变换关系。

定义4 $\Psi(\lambda_1, \lambda_2, \dots, \lambda_m \rightarrow \xi_1, \xi_2, \dots, \xi_n)$ 表示为数据 $\lambda_1, \lambda_2, \dots, \lambda_m$ 经过变换 Ψ 转换为数据 $\xi_1, \xi_2, \dots, \xi_n$ 。这是一种多对多的变换关系。

以上定义中的数据通常属于不同的应用系统。为简单起见,仅考虑定义1的变换关系。假设有 N 个参加交换的应用系统且每个系统之间均需要交换数据,则数据交换实现的复杂性为 $N * (N-1)$,即每个系统必须实现到其它所有系统的数据变换。如果针对每一种交换分别进行开发,将会有很大的工作量。而且当某系统的数据格式发生变化时,系统维护也相当困难。况且一个系统与其它系统交互的数据往往不止一种,因此系统的复杂性还会大大增加。此外,当系统变换关系扩展到上述的其它定义时,复杂性还会进一步增加。因此,需要有一种有效的方法来降低数据交换的复杂性,解决多个应用系统之间相互交换数据的问题。

3 XDDX 中间件模型

针对上述问题,本文提出一种基于 XML 的分布式数据

*)项目资助:重庆市科技攻关项目(2000 6065)。李双庆 博士研究生,主要研究方向为电子商务,多代理技术和分布式计算。游 莲 高工,主要研究方向为电子商务和多代理技术。古 平 助教,主要研究方向为电子商务和计算机网络。程代杰 教授,博士生导师,主要研究方向为电子商务,计算机网络,并行处理。

交换中间件 XDDX(XML-based Distributed Data eXchange)模型。中间件 XDDX 与不同的应用系统实现数据交换时,首先将源应用系统 XML 数据格式转换为一种中间交换数据格式 XDF(eXchange Data Format),再由 XDF 格式转换为目标应用系统的 XML 数据格式。因此,在有 N 个参加交换的应用系统且每个系统之间均需要一对一交换数据的情况下,数据交换的复杂性为 $2 * N$,比上述的 $N * (N - 1)$ 复杂性明显降低。

XDDX 以 XML 为数据交换语言。XML 是由 W3C 提出的一种扩展的标记语言,由于它具有很强的数据描述能力,已成为分布环境下数据交换的重要语言和工具。XDDX 模型的结构如图1所示,它由交换服务接口 XSI(eXchange Service Interface)、数据交换引擎和本体及转换规则库等三部分组成。

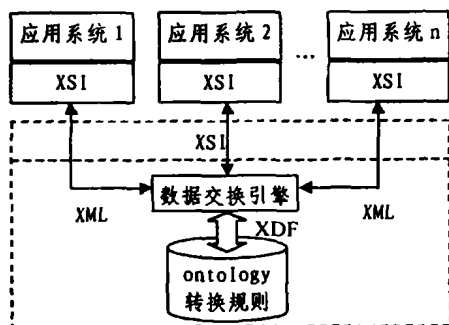


图1 XDDX 数据交换中间件模型

3.1 交换服务接口

XSI 提供 XDDX 与应用系统之间交换数据的接口。从功能上,XSI 可以分为报文服务接口、报文服务和传输接口等三个部分,他们的结构和相互关系如图2所示。

报文服务接口提供与应用系统或 XDDX 的接口,检查所传递数据和参数的合法性;报文服务基于 SOAP 协议,完成报文封装与分解、报文头处理和解析,并能根据应用的需求提供电子商务应用的安全服务,包括数据加密/解密、数字签名和身份认证等。此外报文服务还支持可靠数据传输,负责处理数据传输的逻辑连接、数据确认、差错控制、超时控制和重复数据处理;传输接口提供封装后的报文与承载协议间的映射,所支持的承载协议包括 HTTP、SMTP、FTP 和 IIOP 等。

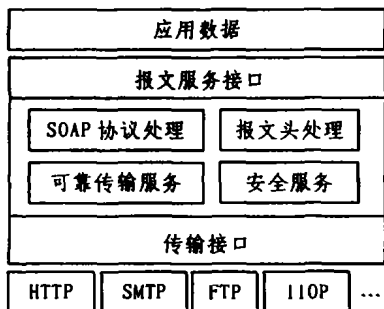


图2 XSI 结构示意图

XSI 报文包括报文信封和报文体两大部分。报文信封由信封头和信封体组成,其中信封头包括报文头信息,而信封体包含承载数据内容的清单。XSI 允许一个报文同时承载多个逻辑独立的数据。以下是用 XML 表示的 XSI 信封:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xml-
```

```
soap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-in-
stance"
xsi:schemaLocation="http://schemas.xmlsoap.org/soap/en-
velope/
http://.../xddx/transport/envelope.xsd" ...
<SOAP-ENV:Header xmlns:xddx="http://.../namespaces/
messageHeader"
xsi:schemaLocation="http://.../namespaces/message-
Header
http://.../xddx/transport/messageHeaderV0-99.xsd"
...
<xddx:MessageHeader ...
</xddx:MessageHeader
</SOAP-ENV:Header
<SOAP-ENV:Body xmlns:xddx="http://.../namespaces/
messageHeader"
xsi:schemaLocation="http://.../namespaces/message-
Header
http://.../xddx/transport/messageHeaderV0-99.xsd"
...
<xddx:Manifest ...
</xddx:Manifest
</SOAP-ENV:Body
</SOAP-ENV:Envelope
```

3.2 本体库^[3,4]

在不同系统间共享知识时,本体具有非常重要的作用。本体是特定领域知识概念化的明确表示和描述,使不同人员或应用系统对某一特定领域的概念有共同的理解,提高交流和协作的效率以及不同系统间的互操作性。XDDX 中的本体库描述交换数据元素间的层次关系和相互联系,是源数据、目标数据和 DXF 间数据元素分解与合成的基础,同时也为开发人员定义数据交换模板提供指导。

概念化 C 可以表示为 $C = \langle D, W, R \rangle$,其中 D 为一个领域,其中的元素为域中的概念,W 为该领域相关状态的集合,R 是领域空间 $\langle D, W \rangle$ 上概念关系的集合。

XDDX 的概念关系主要定义为以下形式:

定义5 Attribute(c, a)表示概念 c 具有属性 a,其中 $c, a \in D$ 。

定义6 Value(a, v)表示属性 a 具有值 v,其中 $a \in D, v \in W$ 。

定义7 Synonym(c1, c2)表示 c1 与 c2 是同义概念,其中 $c1, c2 \in D$ 。

定义8 Map(c, D', r)表示概念 c 的属性可以通过概念集合 D' 中的元素和规则 r 运算来映射,其中 $D' \subset D, r$ 为运算规则表达式, $r \in R$ 。

XDDX 的数据交换引擎根据以上定义建立数据交换的本体和转换规则库,为数据转换样式表的定义提供参照。

本体描述了 DXF 中间数据的基础构成,以及应用系统数据元素与中间数据元素之间的映射关系。不同应用系统中即使名字和意义完全相同的数据,也被认为是不同数据,因为它们的名字空间(namespace)是不同的。一个应用数据元素由一个或多个中间数据元素变换产生,但一个中间数据元素最多只能变换成一个应用数据元素,而且这种转换是可逆的。根据这一原则,也可以把中间数据元素称为原子数据(atom)。

转换规则包含应用数据元素与原子数据的双向转换规则,分别描述如何将应用元素分解转换成原子数据以及如何将原子数据重构成应用数据元素。前者通常会提取部分或全部信息并经过适当的转换获得原子数据,例如日期2002-06-05转换得到三个原子数据,2002年、06月和05日。后者通常会进行一定的运算或函数转换,例如目标数据可能是四个原子数据的运算结果: $A1 * (B2 + C3) - D4$ 。

3.3 数据交换引擎

数据交换引擎的功能有三个方面。首先是根据应用系统

的定义,构建本体和转换规则库,然后将源应用系统的数据转换成 DXF 中间数据,最后是按转换的要求将 DXF 中间数据转换成目标系统的数据。

数据转换的关键技术是扩展的 XSLT^[2]。XSL 是可扩展样式表语言(Extensible Stylesheet Language),它能够改变及转换一种 XML 格式的数据,使其成为另一种 XML 格式。XSLT(XSL Transformations)是一种完全基于文本的转换过程。在 XSLT 过程中,根据一个 XSL 文本样式表,一个 XML 文本文档被转换成另外一种格式的数据,如 HTML 或 VRML 等。XSL 中提供了数据转换需要的基本要素,其中包括节点拷贝、模板选择、循环与选择处理、输出格式和编码等,但是数据项运算能力比较欠缺,无法满足 XDDX 中数据转换的需求。因此 XDDX 根据需要对 XSLT 进行了一定的功能扩充,增强了数据项的运算能力,且 XDDX 在描述运算规则时,保持了 XML 的良好性(wellformed)。

对 XSLT 的扩展体现在数学表达式运算和数据统计运算两个方面。首先扩展了一个或多个数据项构成的数学表达式运算功能,使数据转换过程中能进行必要的算术运算,实现正确的数据转换目标。扩展的数学符号有如下定义:

- 1) + 定义为<xsl:plus/>
- 2) - 定义为<xsl:minus/>
- 3) * 定义为<xsl:time/>
- 4) / 定义为<xsl:divide/>
- 5) (定义为<xsl:lbracket/>
- 6)) 定义为<xsl:rbracket/>

前面例子中的表达式 $A1 * (B2 + C3) - D4$,在转换样式表中表示为:

```
<xsl:value-of select="A1"/><xsl:time/><xsl:lbracket/><xsl:value-of select="B2"/><xsl:plus/>
<xsl:value-of select="C3"/><xsl:rbracket/><xsl:minus/><xsl:value-of select="D4"/>
```

另一方面的扩展体现在对一些数据统计运算功能的支持,包括求和、求平均值、计数、最大值和最小值等。扩展的函数定义如下:

- 1) 求和定义为<xsl:sum>
- 2) 求平均值定义为<xsl:average>
- 3) 计数定义为<xsl:count>
- 4) 最大值定义为<xsl:max>
- 5) 最小值定义为<xsl:min>

例如对运输货物表中所有的货物数量值求和可以表示为:

```
<xsl:sum>
  <xsl:for-each select="Shipment;cargo">
    <xsl:value-of select="Shipment:amount"/>
  </xsl:for-each>
</xsl:sum>
```

显然,以上两个方面的功能扩展是可以混合使用的。

3.4 数据交换模板

为实现正常的的数据交换,还必须定义数据交换模板,以确定数据交换模式、源数据、目标数据、可能的文档类型定义 DTD、对应的数据转换样式表 XSD 和数据间的同步关系。这样,在数据交换过程中,数据交换引擎才能引用对应的模板定义实施数据转换操作。数据交换模板也通过 XML 语言来描述,下面给出一个数据交换模板的实例,它描述了部件供应

商、承运商和制造商间的多对一数据交换关系:

```
<xddx:xddx-Template>
  <xddx:instance ID="1234-5678" Name="Part Provision and Shipment"/>
  <xddx:exchangemode type="Many-to-One"/>
  <xddx:security Protocol="SSL3.0"/>
  <xddx:timeOut>3D</xddx:timeOut>
  <xddx:sourcedata>
    <xddx:dataitem>
      <xddx:company>ABC COMPANY</xddx:company>
      <xddx:dataname>Part Provision Sheet</xddx:dataname>
      <xddx:dataid>ABC-PPS</xddx:dataid>
      <xddx:datatype>http://.../abc-pps.dtd</xddx:datatype>
      <xddx:datatemplate>http://.../abc-pps.xsd</xddx:datatemplate>
    </xddx:dataitem>
    <xddx:dataitem>
      <xddx:company>CDE COMPANY</xddx:company>
      <xddx:dataname>Part Shipment Sheet</xddx:dataname>
      <xddx:dataid>CDE-PSS</xddx:dataid>
      <xddx:datatemplate>http://.../cde-pps.xsd</xddx:datatemplate>
    </xddx:dataitem>
  </xddx:sourcedata>
  <xddx:destinationdata>
    <xddx:dataitem>
      <xddx:company>EFG COMPANY</xddx:company>
      <xddx:dataname>Part Supply Plan Sheet</xddx:dataname>
      <xddx:dataid>EFG-PPSP</xddx:dataid>
      <xddx:datatemplate>http://.../efg-ppsp.xsd</xddx:datatemplate>
    </xddx:dataitem>
  </xddx:destinationdata>
</xddx:xddx-template>
```

总结 随着电子商务的发展,电子商务应用日益强调企业信息系统间的协作和整合。但是由于系统之间存在很大差异,要实现这一目标往往开发难度和工作量都很大。中间件技术为信息系统间的整合提供了良好的基础平台,减少了开发难度和工作量,同时也提高了系统的稳定性和可维护性。本文针对电子商务应用,在分析不同系统间数据交换模式的基础上,提出了一种基于 XML 的电子商务数据交换中间件 XDDX 模型,论述了模型中解决异构电子商务系统间数据交换的关键性技术问题,并根据应用需要对 XSLT 技术进行了扩充。通过 XDDX 中间件模型,能够将异构系统间数据交换的复杂度降低为 $2 * N$,大大提高系统开发的效率,降低开发成本,体现出该中间件技术具有较高的实用价值。该中间件技术的提出源于电子商务的应用,但是其应用领域并不局限于此,该技术可应用于众多基于 Internet 的数据交换应用中。

参考文献

- 1 eXtensible Markup Language (XML) Version 1.0 [S]. http://www.w3.org/TR/REC-xml/, 1999
- 2 XSL Transformations (XSLT) Version 1.0 [S]. http://www.w3.org/TR/1999/REC-xslt/, 1999
- 3 Guarino N. Formal Ontology and Information Systems [C]. In: Proceedings of FOLS'98. 1998. 3~15
- 4 Uschold M, Gruninger M. Ontologies: Principles, Methods and Applications. The Knowledge Engineering Review, 1996, 11(2): 93~115