文章编号:1001 - 9081(2008)02 - 0528 - 03

ESB 环境下可视化数据映射技术的研究

马中杰[1], 周兴社[2], 侯志刚[1], 杨 刚[2], 符 宁[2], 张海辉[2]
(1. 西北工业大学 软件与微电子学院,西安 710065; 2. 西北工业大学 计算机学院,西安 710072)
(mazhongjie@ gmail. com)

摘 要:ESB 系统作为 SOA 的核心部分,如何使企业之间可以实现透明的数据交换,已成为其需要解决的问题之一。针对这一问题,在研究了 XSLT 和相关标准的基础上,提出了一种基于 XSLT 的可视化数据交换模型,并对模型中映射关系从图形化表示转化为 XSLT 表示的关键算法进行了详细地研究,并提出基于 ESB 的可视化数据映射工具的设计。

关键词:ESB;XSLT;数据映射;可视化
中图分类号:TP39 **文献标志码**:A

Research of visualized data mapping technology in ESB environment

MA Zhong-jie[1], ZHOU Xing-she[2], HOU Zhi-gang[1], YANG Gang[2], FU Ning[2], Zhang Hai-hui[2]
(1. *College of Software and Microeletronics, Northwestern Polytechnical University, Xi'an Shaanxi* 710065, *China*;
2. *College of Computer, Northwestern Polytechnical University, Xi'an Shaanxi* 7100721, *China*)

Abstract: As a core component of Service-Oriented Architecture(SOA), Enterprise Service Bus(ESB) systems need to resolve the problem of the transparent data exchange between enterprises. Based on the research of eXtensible Stylesheet Language Transformations(XSLT) and related standards, the paper proposed the visual XSLT data exchange model and analyzed the key algorithm from the graphical expression into XSLT expression of this model in detail. At last, the paper proposed a design of the data visualization tool based on ESB.

Key words: Enterprise Service Bus(ESB); eXtensible Stylesheet Language Transformations(XSLT); data mapping; visual

0 引言

企业服务总线(ESB),它是传统中间件技术与 XML、Web 服务等技术结合的产物,提供了网络中最基本的连接中枢。ESB 的出现改变了传统的软件架构,可以提供比传统中间件产品更为廉价的解决方案,同时它还可以消除不同应用之间的技术差异,让不同的应用服务器协调运作,实现了不同服务之间的通信和整合,ESB 提供了事件驱动和文档导向的处理模式,以及分布式的运行管理机制,它支持基于内容的路由和过滤,具备了复杂数据的传输能力,并可以提供一系列的标准接口。

随着 XML 及其相关技术和应用的发展,XML 逐渐成为了应用间交换数据的一种标准。目前,XML 已有多方支持,并且 XML 的强适应性,使其可以实现对资源的快速包装和集成发布。所以,通过引入了 XML 技术,将 XML 技术与全局数据模式相结合可以使异构数据源集成中间件系统能更好地适应于开放、发展环境中的数据集成。

然而,采用 XML 技术来为消息系统提供统一的全局数据模型仅仅是解决了数据表示的问题,并不能很好的解决异构数据间映射的问题。针对这一问题,常用的解决方式是采用 XML 来描述和存储数据,然后再通过映射规则来进行数据转换,常用的转换规则表示有 XSLT 和用户子自定义规则等,在生成表示转换规则的过程中,目前在 ESB 系统中一般是通过手工编写 XSLT 规则文件来完成规则的制订工作,这种工作方式既繁琐,又对开发人员有很高的技术要求,且容易出错。如果可以提供一种可视化的数据映射技术来生成消息映射规则文件将大大简化制定数据转换规则的工作。可视化数据映射技术为用户提供了完全图像化的界面操作来进行数据映射关系的配置。

1 基于 XSLT 的可视化数据转换模型

1.1 XSLT 简介

可扩展样式表语言转换(XSLT)不仅仅用于将 XML 转换为 HTML 或其他文本格式,而是一种用来转换 XML 文档结构的语言。

1.2 可视化数据转换模型

随着 XML 技术的发展以及 XSLT 规范的不断更新,现在的大多数 ESB 系统都引入了 XML 相关技术并把其与全局数据模式相结合,来完成异构数据源之间的数据交换问题。作者结合 XML 和 XSLT 的特点,提出了解决数据映射问题的可视化数据转换模型,如图 1 所示。

该模型主要由三部分组成:XML 数据服务接口、可视化数据映射工具和 XSLT Processor。XML 数据服务接口是组件和可视化数据映射工具进行交互的枢纽,负责提取组件的数据格式并交予可视化数据映射工具进行展现,组件必须实现 XML 数据服务接口。可视化数据映射工具为用户进行数据映射关系的配置提供了完全图像化的界面,用户可以以拖拽的方式配置映射关系,该工具根据用户配置生成数据映射规则文件,即 XSLT 文件。该文件是 XSLT Processor 工作的基础。XSLT Processor 按照用户配置的映射规则进行数据转换,

收稿日期:2007 - 08 - 13;**修回日期**:2007 - 10 - 27。
基金项目:国家发改委项目 (20052139);"十一五"国防预研项目(06004089);国家"863"计划(2006AA0121626)。
作者简介:马中杰(1982 -),男,安徽阜阳人,硕士研究生,主要研究方向:分布对象; 周兴社(1967 -),男,陕西葡城人,教授,博士生导师,主要研究方向:嵌入式计算、高性能计算; 侯志刚(1983 -),男,陕西富平人,硕士研究生,主要研究方向:分布计算; 杨刚(1974 -),陕西橙城人,男,博士研究生,主要研究方向:分布计算; 符宁(1976 -),男,陕西西安人,博士研究生,主要研究方向:分布计算、可信计算; 张海辉(1977 -),男,湖南娄底人,博士研究生,主要研究方向:分布计算、可信计算。

将源数据转化为目标数据。

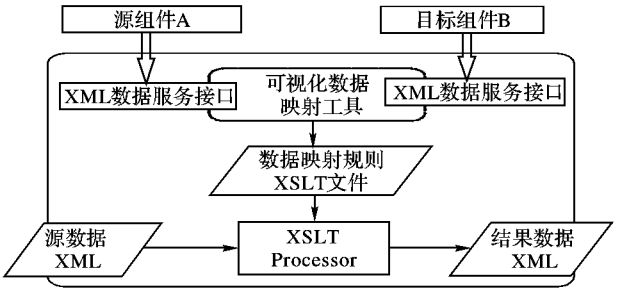

图 1　可视化数据转换模型

该模型的工作流程如下:

1)通过 XML 数据服务接口抽取源组件 A 和目标组件 B 的数据格式,并将组件的消息格式在可视化数据映射配置工具中进行图形化的展现。

2)用户可以根据自己的特定需求进行数据映射关系的配置。

3)可视化数据映射配置工具根据用户配置生成映射关系表示文件即 XSLT 规则文件。

4)XSLT 处理器根据 XSLT 规则文件将源 XML 数据转换成目标 XML 数据。

1.3　可视化数据映射工具

在可视化数据转换模型中,可视化数据映射工具处于核心地位,该工具为用户提供了完全图形化的配置数据映射界面,用户可以在界面上以拖拽的方式配置映射关系,工具根据用户配置生成结果映射规则文件,该工具的功能设计目标如下:

1)支持 xsl:value-of,xsl:template,xsl:for-each 等关键字。

2)支持多种数据源表示样式,包括:XML、XML schema 等消息源表示样式。

3)支持多个 XSLT 函数和操作符。

4)支持对已配置的映射关系保存和恢复的功能。

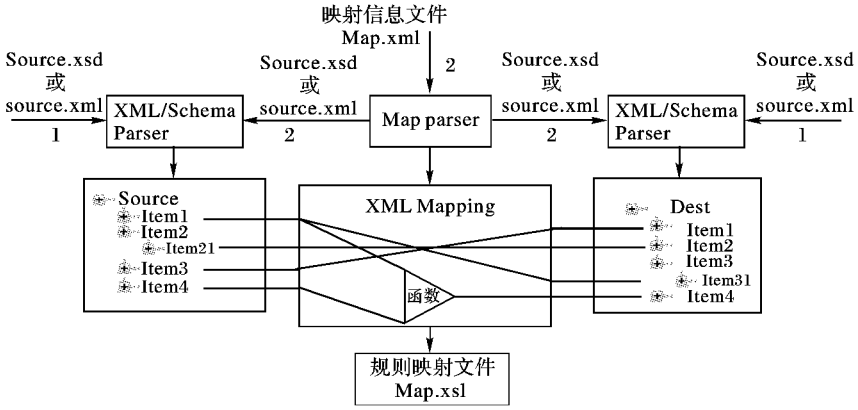

图 2　可视化数据映射工具的系统结构

用户首先指定源文件和目标文件,这两个文件分别用来描述源组件的数据格式和目标组件的数据格式,文件的格式可以为 XML 或者是 XML Schema。XML/ Schema 解析器分别对这两个文件进行解析,并以树形结构显示,用户根据自己的需求进行映射关系的配置,工具根据用户配置生成结果 XSLT 映射文件。

用户也可以指定该工具支持的包含映射信息的 XML 文件来加载已配置的映射关系,该 XML 文件中包含源组件消息格式信息和目标组件消息格式信息,并包含已配置的映射关系的配置信息。

可视化数据映射工具的实现关键是如何将映射关系从图形表示转化为 XSLT 表示。在描述该算法之前先介绍一下两个术语:

MapperTreeNode — 映射树的节点,通过该节点可以访问到该节点的所有子节点,并可以获得节点的连接属性,连接属性由 LinkLine 表示。

LinkLine — 节点之间映射关系的连接线。属性值主要包括:

1)连接线的类型,可以是 xsl:value - of,xsl:template,xsl:for - each 其中的一种。

2)连接线的两端连接点,分别为源节点和目标节点,连接点可以是树的节点也可以是代表函数或者操作符的点。

将图形表示转化为 XSLT 表示的映射关系的算法如下:

```
//按深度优先遍历的方式访问目标树
XSLT( MapperTreeNode root) {
//获得节点的连接线
LinkLine line = node. getLine( );
IF 目标节点是属性节点
THEN
   在XSLT 文件中目标节点的父节点对应的 XSLT 元素中生成
     xsl: attribute 子元素
   //设置 xsl: attribute 元素的 name 属性的值为源节点名称
   attributeElement. setAttribute( "select", "源节点名称");
ELSE IF
line. getType( ) == "xsl: value-of" //连接线类型为"xsl: value-of"
THEN
   在XSLT 文件中目标节点对应的 XSLT 元素中生成 xsl: value-of
     子元素
   //设置 xsl: value-of 元素的 select 属性的值为源节点名称
   valueOfElement. setAttribute( "select", "源节点名称");
   将目标节点对应的 XSLT 元素设置为其父节点对应的 XSLT
     元素的子元素
ELSE IF
line. getType( ) == "xsl: template" //连接线类型为"xsl: template"
                    THEN
      在XSLT 文件中目标节点的父节点对应的
        XSLT元素中生成 xsl: apply-templates 子元
        素;
      / * 设置 xsl: apply - templates 元素的 select
        属性的值为源节点名称 * /
        applyTemplatesElement. setAttribute( "select",
        "源节点名称");
      在XSLT 文件的根元素即 xsl: stylesheet 元素
        下生成一个新 xsl: template子元素
      / * 设置 xsl: template 元素的 match 属性的
        值为源节点的名称 * /
   templateElement. setAttribute( "match", "源节
     点名称");
   设置节点本身对应的元素为新 xsl: template
     元素的子元素
ELSE IF
line. getType( ) == "xsl: for - each"
                              //连接线类型为"xsl: for - each"
THEN
   在XSLT 文件中目标节点的父节点对应的元素中生成 xsl:
     for - each子元素
   //设置 xsl: for - each 元素的 select 属性的值为源节点名称
   forEachElement. setAttribute( "select", "源节点名称");
   设置目标节点对应的 XSLT 元素为 xsl: for - each 元素的子元素
ELSE IF 连接点为代表函数或者操作符的节点
THEN
   根据函数名称和其输入参数生成相应 XSLT 变量
   //设置 xsl: variable 元素的 name 属性的值
   variableElement. setAttribute( "name", "变量名称");
```

```
    string selectValue = "函数名称(函数参数列表)";
    //设置 xsl: variable 元素的 select 属性的值为 selectValue
    variableElement. setAttribute("selec", selectValue)
    设置 xsl: variable 为目标节点对应的 XSLT 元素的子元素
    将函数变量的值赋给目标节点设置目标节点对应的元素为其
      父节点对应的 XSLT 元素的子元素
END IF
//递归调用
For each childeNode of root {
  XSLT(node);
  }}
```

可视化数据映射工具根据 XSLT 语法规则定义以下样式的 XSLT 文件，该文件作为可视化数据映射工具的生成的 XSLT 文件的模板文件。

```
<?xml version = "1.0" encoding = "UTF-8" standalone = "no"? >
< xsl: stylesheet
xmlns: xsl = "http: //www. w3. org/1999/XSL/Transform"
xmlns: fn = "http: //www. w3. org/2005/02/xpath-functions"
xmlns: var = "http: //www. synchroesb. org/2007/var" version = "2.0" >
< xsl: template match = "/" / >
</xsl: stylesheet >
```

该样式中包含了 xsl:stylesheet 元素，该元素是映射工具生成 XSLT 文件的根元素，xsl:stylesheet 元素包含了 XSLT 名字空间、函数名字空间、变量名字空间，以及版本信息。其中最重要的一句话是 < xsl:template match = "/" / >，它定义了一个从根节点来识别的模板，match 属性定义了识别模式，“/”表示从根开始。

2 基于 ESB 的可视化数据映射工具设计

基于 XSLT 的可视化数据转换模型虽然很好的解决了数据映射的问题，但是其并没有在数据转换工具和组件之间无缝交接，提出基于 ESB 的可视化数据映射工具的设计，解决 ESB 环境中组件数据转换的问题。图 3 是 ESB 环境下可视化数据映射工具结构图。

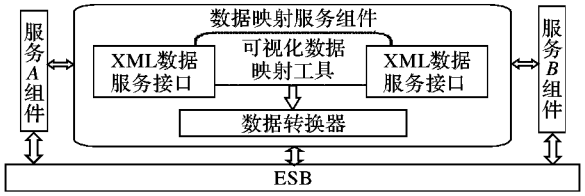

图 3 基于 ESB 的可视化数据映射工具结构

在该结构中，可视化数据映射工具和数据转换器共同组成了数据映射服务，各服务之间的数据通过 ESB 传输。ESB 环境中服务 *A* 组件和服务 *B* 组件进行数据交换的流程如下：

1）可视化数据映射工具通过 XML 数据服务接口抽取源服务 *A* 组件和服务 *B* 组件的数据格式，并将组件的消息格式在可视化数据映射配置工具中进行直观的展现。

2）用户可以根据自己的特定需求进行数据间映射关系的配置，可视化数据映射配置工具根据用户配置生成映射关系表示文件即 XSLT 规则文件。

3）服务 *A* 组件将数据通过 ESB 传送给数据映射服务组件，数据映射服务组件根据 XSLT 规则文件进行转换处理。

4）数据映射服务组件将转换后的数据通过 ESB 传送给服务 *B* 组件，由服务 *B* 组件进行处理。

该模型既很好的解决了数据映射的问题，同时也使得组件间的交互关系可配置，实现了透明化，在 ESB 系统中，这种模型具有普遍适用性。

3 试验

在试验过程中，使用可视化数据映射工具生成的 XSLT 规则，数据映射服务与其他服务组件通过 ESB 进行数据传输，具有高效、可靠、易配置等特点。图 4 是可视化数据映射工具主界面。

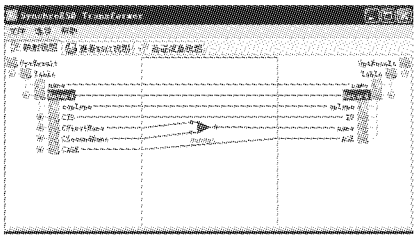

图 4 可视化数据映射工具主界面

图 5 是根据图 4 显示的映射配置关系生成的 XSLT 规则。

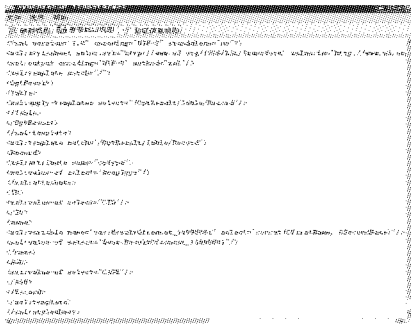

图 5 可视化数据工具 XSLT 结果

试验表明，使用可视化映射工具极大地简化 XSLT 规则文件制定的过程，即使没有掌握 XSLT 技术的人员也可以利用此工具来制定 XSLT 规则。

4 结语

这种可视化数据映射技术在异构数据集成方面，尤其是在异构的关系数据库的集成方面具有很好的效果，使用可视化映射工具简化了 XSLT 规则文件制定的过程，通过 ESB 进行数据的传输工作，可以使各组件间的信息转换更透明，能够有效地解决企业应用系统间的异构数据交换集成问题，而异构数据集成的问题将是是 ESB 系统需要解决的关键问题。

参考文献：

[1] MICHELSON B M. Enterprise service bus Q&A[EB/OL]. [2007 - 02 - 15]. http://www.ebizq.net/hot_topics/esb/features/6117.html.

[2] CHAPPELL D. Enterprise service bus[M]. NY: O'Reilly Publishing, 2004.

[3] KEEN M, ACHARYA A, BISHOP S. Patterns: implementing an SOA using an enterprise service bus[EB/OL]. [2007 - 01 - 25]. http://www.redbooks.ibm.com/ redbooks/pdfs/sg246346.pdf.

[4] CHERBAKOV L, GALAMBOS G, HARISHANKAR R, *et al*. Impact of service orientation at the business level[J]. IBM Systems Journal, 2005, 44(4): 653 - 668.

[5] SCHMIDT M T, HUTCHISON B, LAMBROS P, *et al*. The enterprise service bus: making service-oriented architecture real[J]. IBM Systems Journal, 2005, 44(4): 781 - 797.

[6] Sun Microsystems Inc, Java business integration (JBI) 1.0[S], 2005.

[7] 朱韵篪. 一种基于 XML 的分布式数据交换中间件(XDDX)研究[D]. 重庆: 重庆大学. 2002: 16 - 18.

[8] 李军怀, 周明天, 耿国华, 等. XML 在异构数据集成中的应用研究[J]. 计算机应用, 2002: 22(9): 10 - 12.

[9] 何中, 张申生. XML 映射器的实现[J]. 计算机工程与应用, 2004, 40(4): 137 - 139.

[10] 金峰, 李扬, 王瑜, 等. 一个可视化的多 XML 文档映射系统的设计与实现[J]. 计算机工程, 2004, 30(5): 63.

[11] 王炳清. 通用数据交换中心的设计与实现[J]. 计算机工程, 2004(增): 620.