

# 基于 SDO 的虚拟视图多源数据集成研究

温志萍,刘爱华,程 初

(南京工程学院 计算机工程学院,江苏 南京 211167)

**摘要:**随着网络的发展和企业信息化建设的推进,企业内部的“信息孤岛”问题日益突出。为更好地实现信息共享,在分析 SDO 架构的基础上,提出了一种基于 SDO 的虚拟视图多源数据集成框架。该系统集成框架利用 XSD 描述多重数据源信息,采用 LAV 方法映射全局视图,然后利用 DAS 实现对底层不同数据源的访问。文中详细介绍了该框架体系的构成及工作流程,并给出了具体实现过程。实例表明该方案有效地屏蔽了底层多源数据的差异性,较好地实现了多数据源的集成,解决了“信息孤岛”问题。

**关键词:**SDO;虚拟视图;数据集成;XSD;LAV;DAS

中图分类号:TP31

文献标识码:A

文章编号:1673-629X(2010)07-0089-04

## Multi-Source Data Integration Research of Virtual View Based on SDO

WEN Zhi-ping, LIU Ai-hua, CHENG Chu

(School of Computer Engineering, Nanjing Institute of Technology, Nanjing 211167, China)

**Abstract:** With the network's development and the construction of enterprise information, the problem of information islands is increasingly prominent within the enterprise. In order to better share information, a multi-source data integration framework of virtual view based on SDO is proposed by analyzing the SDO architecture. The framework uses XSD to describe the multiple data source information, uses LAV mapping global view, and then uses DAS to realize the underlying access of the different data sources. Describe in detail the composition of the frame system and work flow, and give a concrete realization of the process. Examples show that the schema is effective in masking the underlying differences in multi-source data, and achieving the integration of multiple data sources to solve the information island problem.

**Key words:** SDO; virtual view; data integration; XSD; LAV; DAS

## 0 引言

随着企业信息化的不断推进,在企业内部各种信息以几何的速度在增长,然而,由于各种历史条件的限制,以至在一个企业中各种类型平台的、各种数据管理系统同时并存,它们构成了企业中的多种异构数据源<sup>[1]</sup>。如无结构化的文本文件、Excel 文件、XML 文件,以及结构化的 Access、Oracle、DB2 数据库等。这些不同的数据源,彼此之间不能互访和互操作,造成企业有很多重复信息存储,投资浪费,进而严重影响了企业信息化进程。

为了解决上述问题,国内外学者提出了许多数据

源集成方案,主要有联邦数据库集成法、数据仓库集成法和中间件模式法集成法<sup>[2,3]</sup>。其中,联邦数据库法只支持对数据库的集成,数据源类型单一;数据仓库法则需较高的成本且数据不能更新;中间件模式法则在数据源和应用程序之间提供一个统一的全局模型,向下对各数据源起协调作用,向上为应用程序提供全局视图和数据访问接口。

SDO 是一种针对在不同的数据源之间使用统一的数据编程模型的规范说明,基于 SDO 进行多源数据集成,可以屏蔽底层不同数据源的异构性,向用户提供统一的访问接口。

## 1 SDO 规范

SOA(面向服务的体系结构)是一个组件模型,它通过定义独立于硬件平台、操作系统和编程语言的接口,来将应用程序的各种功能单元(即服务)联系起来,

收稿日期:2009-11-11;修回日期:2010-02-21

基金项目:国家自然科学基金(50809025);江苏省“青蓝工程”资助项目(苏教师[2007]2号);南京工程学院科研基金(KXJ08072)

作者简介:温志萍(1975-),女,副教授,硕士,研究方向为数据库及网络技术。

从而使得企业可以按照模块化的方式来添加新服务或更新现有服务,保护了现有的 IT 基础建设投资,并解决新的业务需要<sup>[4]</sup>。

SDO(服务数据对象)和 SCA(服务组件架构)是针对 SOA 的重要编程规范<sup>[5]</sup>。SDO 旨在创建一个统一规范的数据接入层,将混杂的数据源整合到工具集和框架中,为通用的应用模式提供健壮的支持,方便应用程序、工具、框架等进行数据的增、删、查、改、约束、更新等操作。利用 SDO,可以简化和统一 SOA 中数据应用程序的开发,以一种可以服从工具和框架的易用方式为不同的数据源提供一种一致的数据访问解决方案,很好地满足企业多源数据集成的需要<sup>[6]</sup>。

### 1.1 SDO 体系结构

SDO 的基本组件<sup>[6,7]</sup>有:

(1)数据对象(Data Object):是 SDO 的基本组件,用于保存具体的数据,包括原始数据以及指向其他数据对象的引用。数据对象也包含了指向元数据的引用,这使得 SDO 的元数据能够被读取,包括数据的类型、关系和约束等。

(2)数据图(Data Graph):一个概念上的数据集合。具体的讲,数据图是一个有多个树根的数据对象的集合,包含一个根数据对象、与根关联的所有数据对象和变更摘要,可以记录所有对数据对象的操作,此外还可以包含表示不同数据源中数据的对象。它是系统中组件之间的传输单元,当进行信息传输时,数据图被序列化为 XML,供 SDO 客户机使用。

(3)元数据(Meta Data):元数据使得开发工具或运行环境能够动态地或者静态地查看数据的属性,包括数据类型、关系和约束等,同时提供了一组跨数据源类型的元数据 API 启用通用工具、框架。

(4)数据访问服务(Data Access Service):对于终端用户而言,访问数据图的方式是通过数据访问服务。数据访问服务负责与后台的数据源进行通信,完成从数据源中创建加载数据图和将对数据图的更改保存回数据源。

(5)更改摘要(Change Summery):更改摘要作为数据图所有活动的历史记录使用<sup>[8]</sup>。一个数据图包含一个更改摘要,当 DAS 返回一个数据图时,更改摘要是空的,随着用户修改数据对象的状态(如创建或删除),这些更改将被记录在更改摘要中。通过更改摘要,DAS 能够访问数据图中任何一个数据对象的更改历史。

SDO 的架构如图 1 所示,是基于断开连接的数据图的概念的。

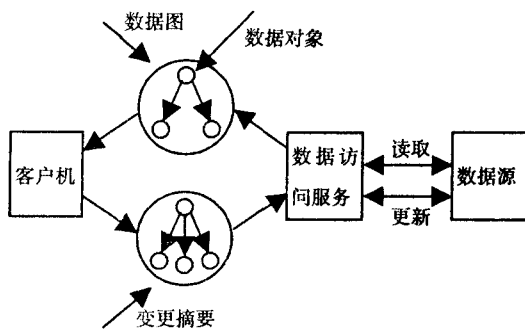


图 1 断开连接的数据图架构

在该模式下,SDO 的工作过程为:用户发送一个加载数据图的请求给 DAS,DAS 向数据源获取数据创建一个表达该数据的数据图,并将该数据图返回给用户,用户访问或更新数据图,然后基于更新后的数据图调用 DAS,DAS 启动事务更新数据源中的数据。

### 1.2 SDO 相关技术

SDO 相关技术有 XSD、EJB 等。

(1)XSD 技术<sup>[9]</sup>。XSD(XML Schemas Definition)描述了 XML<sup>[10]</sup>文档的结构,可以用来验证某个 XML 文档是否符合要求。XML Schema 本身是一个 XML 文档,它符合 XML 语法结构,支持一系列的数据类型,支持综合命名空间,可以用通用的 XML 解析器解析它。SDO 规范提供了序列化的 XML Schema。

(2)EJB 技术<sup>[11]</sup>。EJB(Enterprise Java Beans)是一种服务器端组件的体系结构,它使得应用 Java 语言可以开发分布的、健壮的、安全的企业级应用而不用预先开发复杂的分布式对象框架。EJB 从一开始就被设计成支持可移植和可重用于任何开发商的企业级中间件服务,它具有很高的灵活性、很好的兼容性和跨平台性、支持“即插即用”的特性。

## 2 基于 SDO 的数据集成

基于 SDO 的虚拟视图多源数据集成,利用 SDO 所提供的统一的 SDO API,即可以同一方式处理来自不同数据源的数据,包括关系数据库、实体 EJB 组件、XML 页面、Web 服务、Java Connector Architecture、Java Server Pages 页面等,有效地屏蔽了底层不同数据源的异构性。

### 2.1 体系结构

基于 SDO 的多源数据集成体系,可分为用户接口层、数据适配层和数据源层,如图 2 所示。

(1)用户接口层。

用户接口层主要负责系统与用户的交互。系统提供访问界面,用户根据虚拟数据视图构造访问请求,并将访问请求提交给数据适配层进行处理,当数据适配

层将操作结果返回时,系统将处理结果以友好的形式展现给用户。

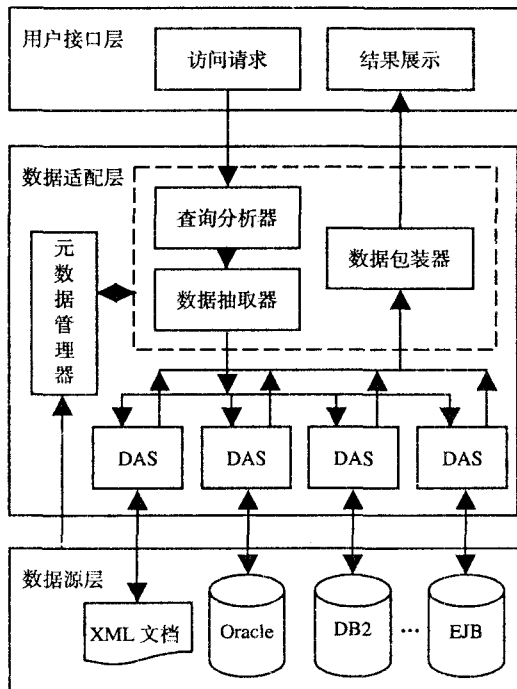


图2 集成体系结构

## (2) 数据适配层。

数据适配层是集成系统的核心,它负责访问请求的处理,主要包括元数据管理器、查询分析器、数据抽取器、数据包装器和 DAS 模块。

**元数据管理器:**主要负责数据源配置信息的管理,数据对象的自动生成、更新和发布,以及数据对象映射构建虚拟视图。

**查询分析器:**负责接收用户接口层的访问请求,对其进行语法解析,验证其访问请求的合法性,并进行优化整理。

**数据抽取器:**接收优化后的请求,将用户对虚拟视图的全局访问请求拆分转换为对各底层数据源的子操作,提交给相应的 DAS。

**数据包装器:**负责在本地访问结束后,将 DAS 的处理结果整理合并,创建相应的数据图,返回给用户接口层进行结果展示。

**DAS 模块:**对数据源的访问必须通过 DAS 进行,不同的数据源对应不同的 DAS。DAS 负责查询相应数据源,创建包含数据图的数据对象,并将处理信息以数据图的形式予以返回,同时还负责将修改后的数据图存回相应数据源。

## (3) 数据源层。

包括多种形式的分布式异构数据源,如结构化的 Oracle、DB2 数据库等,半结构化的 XML 文档、实体

EJB 等。

## 2.2 工作流程

系统主要工作流程为:

(1)首先,用户登录数据集成平台,系统对用户进行身份鉴别,通过鉴别的用户通过访问接口提交访问请求,并将访问请求提交给查询分析器。

(2)查询分析器接收到用户的访问要求后,进行词法、语法分析,向元数据管理器验证其操作的合法性。

(3)验证通过后,数据抽取器根据全局模式和局部模式的映射关系,对全局访问请求进行解析,将其分解为对各局部数据源的子操作,由数据抽取器将子操作分发给各 DAS。

(4)各 DAS 根据各数据源的配置、数据对象映射信息和数据访问内容等,对相应数据源进行增加、删除、修改和查询操作,并将访问结果以标准形式返回给数据包装器。

(5)数据包装器接收到各 DAS 返回的处理结果,将各个数据图进行结果合并,最后形成标准数据图,返回给用户接口进行发布。

(6)发布引擎将处理结果以统一友好的形式(网页或表格等)呈现给用户。

## 3 集成系统的实现

利用 SDO 进行多源数据集成访问,主要包括:数据源局部模式定义、全局模式映射、数据库连接、数据图的构建,以及对数据源的操作。

### (1) 数据源局部模式定义。

数据源局部模式,是描述各数据源基本信息的局部模式,它采用 XSD 文件格式进行存储,包括数据源的名称、地址、访问权限,以及数据对象与数据库模式的对应关系等。其模式结构如图 3 所示。

### (2) 全局模式映射。

局部数据源的相关信息由元数据管理器负责注册、管理,在利用 XSD 进行各数据源局部模式的定义后,系统采用 LAV<sup>[12]</sup>方法通过模式映射将各数据源的 XSD 文档统一映射到一个具有一致的数据结构的全局 XSD 文档中。

构建全局模式时,可由各数据源局部模式为其结构的子元素构成。若局部数据源中存在部分数据对象重叠,可在全局模式中相应子元素的定义处增加相应约束。通过该方法,即可实现全局模式与局部模式之间的映射。

### (3) 创建 DAS。

用户对系统的访问,通过查询全局模式定义的 XSD 文档,最终转化为对局部数据源的访问。在多源

数据集成系统中,对局部数据源的访问,均通过 DAS 进行。

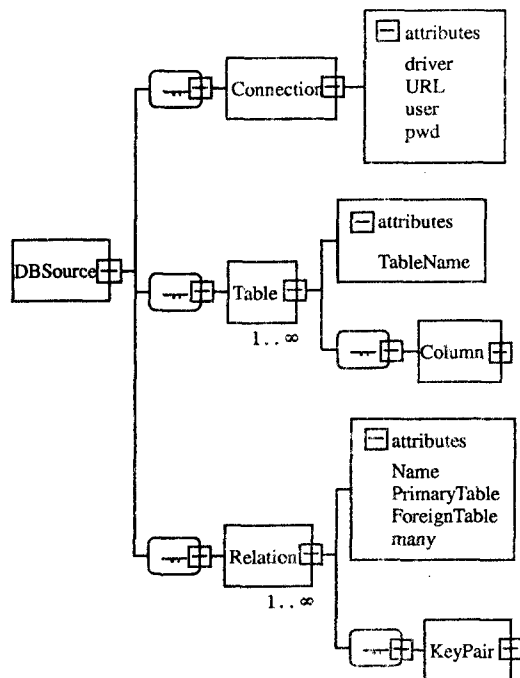


图 3 数据源局部模式结构图

在 SDO 规范中,可通过 DAS.FACTORY 的 createDAS 方法创建 DAS,方法如下:

```
InputStream input = getClass().getClassLoader().getResourceAsStream(数据源 XSD 文件名);
das = DAS.FACTORY.createDAS(input);
```

#### (4) 数据库连接。

创建 DAS 后,即可通过 DAS 与不同的数据源进行连接。而进行数据库连接所需的数据源连接信息保存在数据源局部模式定义的文件中。

获取数据源连接信息、进行数据库连接的方法如下:

```
Config config = ConfigUtil.loadConfig(getConfig(数据源 XSD 文件名));
String url = config.getConnection().getURL();
String user = config.getConnection().getUser();
String pwd = config.getConnection().getPwd();
Connection con = DriverManager.getConnection(url + "? user = " + user + "&password=" + pwd);
```

#### (5) 数据图的构建。

数据图是应用程序利用 SDO 进行数据操作的基础,通过 DAS 创建,DAS 返回一个包含数据对象的数据图或者返回一个空数据图。

在 SDO 规范中利用 DAS 的 getGraph() 方法可以生成数据图;使用数据对象的 getDataGraph() 方法也可以获得该对象所属的数据图。

#### (6) 数据对象的生成。

数据对象是应用程序利用 SDO 具体实现数据及其内容的单元。

通过 Command 对象的 executeQuery 方法可创建数据对象,如:

```
Command rd = das.getCommand("select * from AllSurveys");
DataObject root = rd.executeQuery();
```

在 SDO 规范中,还可利用空数据图的 createRootObject() 方法创建该数据图的根数据对象,或者利用数据对象的 getRootObject() 方法获得该数据对象所属的根数据对象。然后通过遍历数据图的树形结构,即可访问到所有其它数据对象。

利用数据对象的 createDataObject() 方法也可以创建新的数据对象。

#### (7) 对数据源的操作。

实例化数据对象后,通过遍历和访问数据对象即可实现 DAS 对数据源的查询操作,方法如下:

```
List surveys = root.getList("Survey");
for(int i=0; i<surveys.size(); i++){
    System.out.println("SID:" + ((DataObject) surveys.get(i)).getInt("SID"));
}
```

若要对数据源的更新操作,可由 DAS 对数据对象进行修改,然后利用 DAS 的 applyChanges() 方法实现数据对象的提交。如修改一条记录:

```
DataObject sv = root.getDataObject("survey");
sv.set("SID", "S2-2");
das.applyChanges(root);
```

## 4 结束语

随着信息技术广泛深入的运用,多源数据的集成已成为数据领域研究的热点问题。文中在借鉴中间件模式思想的基础上,提出了一种基于 SDO 的虚拟视图多源数据集成方案。该方案利用 SDO 在数据处理上的优势,简化和统一了数据应用程序的开发,为用户提供了一个统一的访问接口,使用户在无需关心底层异构数据源差异性的情况下,实现了对多源数据的访问和操作。

#### 参考文献:

- [1] 张晓星,唐朝晖.企业多源异构数据库数据集成的研究[J].可编程控制器与工厂自动化,2007(3):66-68.
- [2] 杨长辉.基于 XML/Web Services 的异构数据集成系统的研究和应用[D].重庆:重庆大学,2006.
- [3] 秦文文.基于 XML 的异构数据库集成研究[D].青岛:山东科技大学,2007.
- [4] 林怀恭,裴瑞华,罗辉琼,等.基于 SOA 架构的服务集成技

(下转第 97 页)

消息和 XQuery 模块发来的 HTTP POST 消息并发往服务器,同时,TCP/IP 模块接收服务器发来的消息并识别消息的协议类型分发给 SIP 模块或 XQuery 模块。

在 Server 软件结构图中,Overload Control 过载控制模块有助于保证系统安全可靠地运行。HTTP Server 解析收到的请求,分发给适当的协议适配层;同时,HTTP Server 还具备双向认证机制和鉴权作用,保证系统消息的安全传输。HTTP POST 请求消息将分发给 XQuery 适配器处理模块,SIP MESSAGE 的确认消息将分发给 SIP 适配器处理模块。XQuery Adapter 解析 HTTP POST 请求,得到 HTTP Request - URI 和 XQuery 查询表达式;XQuery Adapter 还负责将反向搜索结果封装成 HTTP POST 响应向客户端发送。SIP Adapter 用于接收客户端发来的 SIP MESSAGE 确认消息以及封装自动通知消息向客户端发送自动通知。Routing 模块向目标对象路由请求消息或响应消息。XDM Core 是整个服务器软件结果的核心,它根据 Routing 发来的消息执行相应处理,包括根据搜索表达式获取反向搜索的结果、自动发送通知消息等。Data 是一个存储模块,用于存储 PCC 信息和联系人信息。

反向搜索的软件结构的设计符合 OMA 定义的各项功能规范,加上对层管理(LM)和 Overload Control 等高可用性的架构设计,可以满足电信级业务运营的要求。

#### 4 结束语

文中主要对融合地址簿反向搜索在系统中的结构,实现技术及其消息流程做了详细的分析和研究,同时给出了反向搜索实现的软件结构。反向搜索的提出克服了搜索时双方信息不对称,不便于被搜索方进行联系人管理等问题。因此,反向搜索技术作为 CAB 系统功能的补充,能更充分地提升用户的体验,同时也使

CAB 系统在未来的通信系统,特别是 3G 通信中将会获得更广泛的应用。

#### 参考文献:

- [1] Open Mobile Alliance<sup>TM</sup>. Converged Address Book Requirements, OMA - RD - CAB - V1.0[M]. [s.l.]:[s.n.], 2008.
- [2] Open Mobile Alliance<sup>TM</sup>. Data Synchronization Syntax, OMA - TS - DS - Syntax - V2.0[M]. [s.l.]:[s.n.], 2008.
- [3] Boag S, IBM Research, Chamberlin D, IBM Almaden Research Center. W3C Recommendation. XQuery 1.0: An XML Query Language[S]. 2007.
- [4] Rosenberg J, Schulzrinne H, Columbia U, et al. IETF RFC 3261. SIP: Session Initiation Protocol[S]. 2002.
- [5] Rosenberg J. IETF RFC 4825. The Extensible Markup Language (XML) Configuration Access protocol (XCAP)[S]. 2007.
- [6] Open Mobile Alliance<sup>TM</sup>. Converged Address Book Architecture, OMA - AD - CAB - V1.0[M]. [s.l.]:[s.n.], 2008.
- [7] Open Mobile Alliance<sup>TM</sup>. XML Document Management (XDM) Specification, Version 2.0, OMA - TS - XDM - Core - V2.0[M]. [s.l.]:[s.n.], 2008.
- [8] Open Mobile Alliance<sup>TM</sup>. XML Document Management Architecture, Version 2.0, OMA - AD - XDM - V2.0[M]. [s.l.]:[s.n.], 2008.
- [9] 赵鑫,王纯,李陆,等. PoC XDMS 的设计与实现[J]. 计算机系统应用, 2008(10):107-110.
- [10] 程雷,朱茂盛. XQuery 的实现机制[J]. 计算机工程与应用, 2002(24):78-85.
- [11] 施振佳,曹渠江. 基于 XQuery 查询优化的研究[J]. 计算机应用与软件, 2006(11):86-88.
- [12] Campbell B, Rosenberg J, Schulzrinne H, et al. IETF RFC3428 Session Initiation Protocol (SIP) Extension for Instant Messaging[S]. 2002.

(上接第 92 页)

- 术的研究[J]. 计算机技术与发展, 2009, 19(7):141-144.
- [5] 刘壮业,姚郑. 面向服务架构若干关键问题研究[J]. 计算机工程与设计, 2009, 30(3):600-604.
- [6] BEA, IBM, Oracle, et al. Next - Generation Data Programming with Service Data Objects [EB/OL]. 2006. <http://www.ibm.com/developerworks/library/specification/ws-sdo/>.
- [7] 王紫瑶,南俊杰,段紫辉,等. SOA 核心技术及应用[M]. 北京:电子工业出版社, 2008.
- [8] 沈祥,方振宇. 面向服务架构的研究[J]. 计算机技术与发展, 2009, 19(2):74-76.
- [9] MedBiquitous. XML Schema Design Guide lines: version1.2 [EB/OL]. 2004. <http://www.medbiq.org/technology/tech-architecture/xml-design-guide-lines.pdf>.
- [10] Marchiori M. XQuery 1.0: An XML Query Language[EB/OL]. 2005-11-03. <http://www.w3.org/TR/2005/CR-xquery-20051103/>.
- [11] 陈立岩. EJB 组件技术及应用[J]. 计算机技术与发展, 2007, 17(3):62-64.
- [12] 杨雪梅,董逸生,王永利,等. 异构数据源集成中的模式映射技术[J]. 计算机科学, 2006, 33(7):87-91.