面向协同系统集成的数据同步模型*

申利民，李卫东
（燕山大学 信息科学与工程学院，河北 秦皇岛 066004）

摘 要：为了在系统协同工作过程中使异构数据库的数据同步，基于 agent 技术给出了一个数据同步模型。在该模型中，agent 的局部代理利用数据同步监听器对数据的变化进行实时监听，全局代理利用数据同步检测器对数据是否同步进行定期检测，并生成相应的同步命令由全局代理分发执行，解决了协同系统中异构数据库的数据同步问题。最后，该模型应用到一个“项目控制管理系统”中，说明了该模型的可用性和有效性以及数据同步机制的正确性和实时性。

关键词：协同系统；系统集成；异构数据库；数据同步；agent
中图分类号：TP311 **文献标志码**：A **文章编号**：1001-3695(2012)04-1384-03
doi:10.3969/j.issn.1001-3695.2012.04.051

Data synchronization model for collaborative system integration

SHEN Li-min, LI Wei-dong
(*College of Information Science & Engineering, Yanshan University, Qinhuangdao Hebei* 066004, *China*)

Abstract: In order to carry out data synchronization of heterogeneous database in collaboration process, the paper presented a new data synchronization model based on agent technology. In the model, the local agent monitored the changes of the data using data synchronization listener in real time. And the global agent tested system using data synchronization detector periodically and generated corresponding synchronization commands that were distributed execution by the global agent, and solved the data synchronization problems of heterogeneous database in collaborative system. Finally, the data synchronization model was applied to a "project control and management system". The result shows the availability and effectiveness of the model and the accuracy and real-time of the data synchronization mechanism.

Key words: collaborative system; system integration; heterogeneous database; data synchronization; agent

在计算机支持的协同工作(computer supported cooperative work,CSCW)环境下，协同系统一方面需要协同项目的多方参与者，另一方面需要协同来自不同时间或厂家的协同工具和管理工具，使它们在统一和透明的环境和界面下进行协同工作和人机交互[1]。系统集成是根据应用需要将不同的系统有机地组合为一个一体化的、功能更加强大的新型系统的过程和方法。系统集成是对现有的成熟系统进行组合，而不是针对功能再次开发，所以集成的系统是低成本的、高效率的、性能匀称的、可扩充的、可维护的系统[2]。在协同系统集成过程中，所有子系统对应的数据库构成异构数据库。异构数据库可以实现数据的共享和透明访问，每个数据库在加入异构数据库之前本身就已经存在，拥有自己的 DBMS。异构数据库的各个组成部分具有分布性、自治性和异构性，即每个数据库均有独立自治能力，可分布在多个场地，并且编码和运行平台也各不相同[3]。Agent 是一个自主的并具有一定智能的实体，能够对多个异构数据库进行虚拟集成。Agent 通过接受操作命令，并依据很强的控制规则逻辑完成对数据库的操作，从而屏蔽了多个数据库的异构性[4]。数据同步是维持多个数据库间数据一致性的一种手段，在多个数据库中部分数据保持一致的前提下，各个数据库可以任意扩展其他数据信息[5]。通过实现异构数据库的数据同步使得系统在数据驱动的作用下将这些协同的工具进行集成，最后构成一个统一的协同系统。从用户的角度看，协同系统的内部结构是透明的、统一的[6]。

目前已有很多模型解决了异构数据库间的数据同步问题。基于 XML 与 Web 服务的数据同步模型是面向服务的，不需要本地保存的数据便可直接进行更改[7]。基于触发器技术的增量更新日志表数据变化的数据同步模型能够有效地解决数据同步，但是在数据安全方面存在一定的局限性[8]。异构数据同步模型(DMS)将当今最流行的对等网络引入到了模型中，通过 XML 技术与 P2P 技术相结合，为异构数据模式提供了一个标准的数据模式，解决了数据差异更新困难和传输效率低等难题[9]。这些模型虽然使异构数据库的数据同步得到实现，但是在数据同步过程中存在着扩展性、可靠性、实时同步性能差等问题。

针对上述模型的不足，本文提出了一个新的 agent，并在该 agent 的基础上给出了面向协同系统集成的数据同步模型。

1 基于 agent 的数据同步模型

1.1 模型描述

本模型由应用层、代理层、数据层三层构成。应用层由多

收稿日期：2011-09-12；**修回日期**：2011-10-19 **基金项目**：国家自然科学基金资助项目(60873008)；河北省自然科学基金资助项目(F2011203234)

作者简介：申利民(1962-)，男，黑龙江佳木斯人，教授，CCF 会员，博士，主要研究方向为柔性软件工程、协同计算(shenllmm@sina.com)；李卫东(1986-)，男，硕士，主要研究方向为协同系统研究与设计.

个局部系统组成,在逻辑上是一个整体,但在物理上是分布的、异构的。数据层由多个局部数据库组成,在逻辑上是一个异构数据库整体,在物理上各个局部数据库具有分布性、自治性、异构性。应用层中的局部系统与相应的数据层中的局部数据库构成自制的独立系统。为了使多个独立的系统集成为一个一体的协同系统平台,在系统集成过程中引进了 agent 代理层,如图 1 所示。

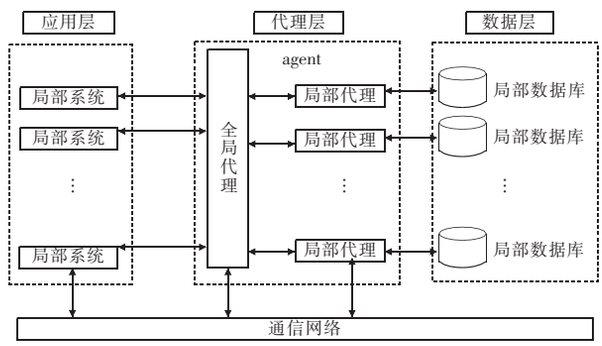

图 1　基于 agent 的数据同步模型系统结构

Agent 由全局代理和局部代理组成,通过将多个局部数据库虚拟集成为一个全局数据库,从而屏蔽各数据库的异构性。全局代理是应用层与数据层交互的桥梁,主要实现向局部代理分发命令。另外,全局代理中的同步检测器具有数据同步检测和同步命令生成的功能。局部代理主要担任全局事务在局部系统中的事务代理,同时监听相应局部数据库中同步数据的变化,并生成同步命令发送到全局代理以实现异构数据库的数据同步。

1.2　局部代理

局部代理分为命令执行模块和同步处理模块。命令执行模块由解析器、编译器、命令执行器、关系处理器组成,主要担任接收全局代理发送的命令并解析执行。同步处理模块由数据同步监听器和同步命令生成器组成。同步监听器实现对局部数据库中同步数据变化的监听,并由同步命令生成器生成同步命令发送给全局代理以实现异构数据库同步。数据库链接池利用缓存技术可提高数据库的访问效率。如图 2 所示。

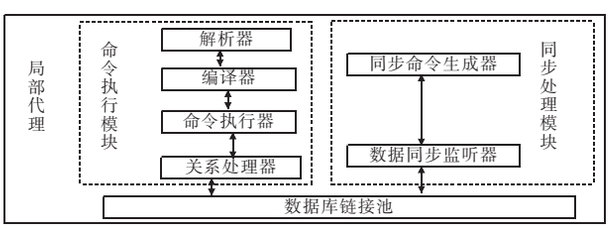

图 2　局部代理内部结构

1.2.1　命令执行模块

命令执行模块担任接收和执行全局代理所发送的命令。全局代理向局部代理发送的命令可能是来自应用层的请求(应用层通过 agent 操作局部数据库)、由同步检测器生成(系统中存在不同步数据)、同步处理模块的发送(数据同步监听器监听到同步数据的变化)三个方面。当局部代理接收到全局代理所发送的命令时,命令执行模块首先由解析器对命令进行功能解析并传递给编译器;编译器把命令编译为当前局部代理可识别的命令并传给命令执行器执行。关系处理器实现对当前局部系统自身业务逻辑关系的处理。关系处理器首先要判断命令的操作对象是否与局部系统的其他信息有关联,如果有关联,需要对其他信息进行相应的处理,如数据表关联,文件关联、业务关联等。

1.2.2　同步处理模块

处理数据同步的过程就是对多个局部数据库同时进行添加、修改、删除操作以实现异构数据库的数据同步。当局部系统操作与之对应的局部数据库时,如果被操作的是同步数据,数据同步监听器会触发 agent 对其他局部数据库做相应的操作以实现数据同步。数据同步监听器主要监听局部系统对相应局部数据库操作所引起同步数据的改变,但是由 agent 操作局部数据库所引起的同步数据的变化则不会触发数据同步监听器的执行。当数据同步监听器监听到同步数据的变化时,会把数据变化的记录传递给同步命令生成器。同步命令生成器将生成一条或多条对其他局部数据库的操作命令,最后把同步命令发送到全局代理,再由全局代理把命令分发到其他局部代理完成对同步命令的执行,实现多个局部数据库的数据一致。C#算法如下:

```
Listion L = new listion( );/ * 创建新的监听器对象 * /
if( L == null) return;/ * 监听器如果没有监听到同步数据的变化则返回 * /
string cmd = Listion. getListionCMD( );/ * 获取对同步数据操作的方式 * /
List A = Listion. keys; / * 所有被操作的同步数据关键字列表 * /
switch( cmd)
{
case " Delete" :postCommand( createDelCMD( A) ) ;break;
/ * createDelCMD 根据同步数据关键字(或列表)生成删除同步操作命令 * /
case " Insert" : postCommand ( createAddCMD ( getnowInfor ( A ) ) ) ; break;
/ * getnowInfor 根据关键字从当前局部数据库获取同步数据; createAddCMD根据同步数据(或列表)生成添加同步操作命令; postCommand 把同步操作命令传回全局代理 * /
case " Update" : postCommand ( createEditCMD ( getnowInfor ( A ) ) ) ; break;
//createEditCMD 根据同步数据(或列表)生成修改同步操作命令
}
```

1.3　同步检测器

全局代理中的同步检测器用于检测整个系统中是否存在不同步的数据,并对不同步的数据给出相应的同步措施,所以同步检测器起着补救的作用。

1.3.1　不同步数据的产生

局部代理中的同步处理模块通过监听同步数据的变化并生成同步命令执行已经实现了异构数据库的数据同步。但是由于系统中存在不可预知的错误,并不能保证多个局部数据库的数据真正同步。产生数据不同步的原因主要有:人为修改数据库、数据同步监听器失效、同步处理模块执行时产生了错误、局部系统本身的漏洞等。数据不同步的主要体现有:a)多个需要同步的数据表中同步数据关键字(同步数据的主键)的内容不相同,如 A 表中有用户名为 x 而在 B 表中不存在用户名为 x 的记录;b)多个需要同步的数据表中同步数据关键字内容相同,但是其他同步数据内容不相同,如 A 表中有用户名为 x 的记录姓名为 a,而 B 表中用户名为 x 的记录姓名为 b。

1.3.2　不同步数据的处理

同步判定程序是同步检测器的一个组成模块,用于检测系统中是否存在不同步数据。当同步判定程序检测出数据存在不同步时,首先会从主体表(两个数据比较的参考主体)中把需要同步的数据提取出来,最后生成同步命令分发到局部代理并执行,完成对不同步数据的处理。以下为同步检测器的具体

执行过程。

a)针对多个需要同步的数据表中同步数据关键字不同,同步处理算法需要对多个表的同步数据关键字进行逐一比较,并以主体表的数据为标准生成添加或删除的同步操作命令,最后分发到局部代理执行。C#算法如下:

```
List A = null;/ * 利用 List 实现动态数组 * /
List B = null;
string [ ]mainK = array1; / * 主体表中的所有同步关键字 * /
string [ ]otherK = array2; / * 其他表中的所有同步关键字 * /
for( int i = 0;i < mainK. length;i ++ )
{
  if( compareKey( mainK[ i ], otherK) ) A. add( mainK[ i ]);
  / * compareKey 判断关键字在数组中是否存在,如果不存在返回真 * /
}
for( int i = 0;i < otherK. length;i ++ )
{
  if( compareKey( otherK[ i ],mainK) ) B. add( otherK[ i ]);
}
if( A! = null) sendCommand( createAddCMD( getmainSynInfor( A) ));
/ * getmainSynInfor 根据关键字(或列表)从主体表中获取同步数据信息;sendCommand 向相应的局部代理发送命令 * /
if( B! = null) sendCommand ( createDelCMD( B) );
```

b)针对多个需要同步的数据表中同步数据内容不同,同步处理算法需要对多个表中当前用户的同步数据内容进行比较(对系统中全部的同步数据比较工作量相当大,所以同步检测器只在用户登录和退出时检测当前用户的数据是否同步),并以主体表的数据为标准生成修改的同步操作命令,最后发送到局部代理执行。C#算法如下:

```
string userkey = key; / * 当前用户的同步关键字 * /
object mainInfor = getmainSynInfor( userkey);
object otherInfor = getotherSynInfor( userkey); / * 其他表中同步数据信息 * /
if( compareContent( mainInfor, otherInfor)
//compareContent 比较两组同步数据是否相同,如果不同返回真
sendCommand( createEditCMD( mainInfor) );
```

2 应用实例

河北省港务局一公司的“项目控制管理系统”是对多个现有成熟系统的集成,即是由多个独立自制的系统组合而成的协同系统。系统主要由权限管理、项目管理、进度控制、日志管理、实时聊天、个人空间、通知管理、系统消息等子系统组成。权限管理实现了给指定用户授予相应的权限;项目管理实现对项目信息的基本操作(创建、修改、删除等);进度控制实现了以甘特图的方式直观地向用户展示项目信息;日志管理主要记录员工每天对项目的执行情况并体现到项目进度控制图中;实时聊天实现用户在线实时交流;个人空间用于管理用户个人的私有信息(包括文件管理、虚拟目录、用户群组);通知管理实现向指定用户发送日常通知和维修通知;系统消息实现系统向用户进行必要的消息提示。这些子系统具有分布性、自治性、异构性的特性。其中能够充分体现协同系统集成过程中异构数据库数据同步的主要有权限管理、项目管理、实时聊天。

权限管理、项目管理、实时聊天分别对应的数据库存为 DB1(SQL Server 数据库)、DB2(SQL Server 数据库)、DB3(SQLite 数据库)。DB1、DB2、DB3 是分布的、自治的、异构的,分别包含各自的用户表 usersT1、usersT2、usersT3,并且三个用户表中的部分数据是保持一致的。其中,usersT1 是主体数据表,作为 usersT2 和 usersT3 的参考对象。同步数据主要有用户名、姓名、性别、部门等信息,其中用户名为同步数据关键字。当局部代理内的数据同步监听器监听到相应局部系统对同步数据直接操作时,会触发同步命令生成器生成同步命令并发送给全局代理,由全局代理分发到各个局部代理执行,从而实现三个局部数据库同步。例如项目管理修改 usersT2 的姓名信息时,与项目管理对应的局部代理会监听到同步数据的变化并生成对 usersT1 和 usersT3 的相应的同步操作命令发送到全局代理,由全局代理分发到与权限管理和实时聊天相对应的局部代理执行。当全局代理中的同步检测器检测到系统中用户数据存在不同步时,参考 usersT1 对 usersT2 和 usersT3 生成添加、修改、删除同步命令,并分发到局部代理执行,最终实现对系统不同步的数据实施同步处理。实现多个局部数据库的数据同步的前提下,在数据驱动的作用下完成“项目控制管理系统”的系统集成,展现给用户的是一个一体化的、透明的、稳定的、统一的协同系统控制平台。

最后,通过把基于 XML 与 Web 服务的数据同步模型[7]应用到该项目管理系统,由于该模型的同步处理操作并不是马上执行并且没有考虑到同步操作之后在系统中仍然存在不同步数据,数据同步的实时性和正确性方面不能得到很好的效果。但基于本文提出的数据同步模型,由于对数据同步处理是采用监听触发方式实现的,同时对系统中存在的不同步数据具有定期检测功能,数据同步的实时性和正确性在系统中得到很好的验证,证明本文提出的数据同步模型应用到协同系统集成过程中具有可用性和有效性。

3 结束语

本文在基于 agent 技术的基础上,提出了一种数据同步模型来解决协同系统集成过程中的异构数据库数据同步问题。由于采用该模型进行系统集成时是不需对现有子系统进行修改,从而保证了系统的稳定性和系统集成的速度。本模型已在河北港务局一公司的“项目控制管理系统”得到很好的验证。目前,该系统已持续稳定运行时长五个月,深受用户的好评。但是在本模型中同步监听器并没有真正地监听数据,而是通过修改程序使局部系统触发同步监听器的执行,实现同步监听器对同步数据变化的监听功能将是下一步要研究的问题。

参考文献:

[1] 杨沙洲,所光,王桂彬,等. 异构应用系统协同技术综述[J]. 计算机研究与发展,2007,44(z1):7-11.

[2] 李青山,王梅圣,赵晨光,等. 面向系统集成领域的多 agent 智能协作模型[J]. 西安电子科技大学学报,2010,37(1):62-67.

[3] 罗林球,孟琦,李晓,等. 异构数据库迁移的设计和实现[J]. 计算机应用研究,2006,23(12):233-238.

[4] 徐慧,杨永国. 多 agent 协同处理模型的研究与设计[J]. 计算机工程,2010,36(5):67-69.

[5] 张瑛,夏克俭,张法明,等. 分布式异构数据库数据同步系统的研究与实现[J]. 小型微型计算机系统,2007,28(10):1803-1806.

[6] 林源,陈志泊. 分布式异构数据库同步系统的研究与应用[J]. 计算机工程与设计,2010,31(24):5278-5281.

[7] 索红光,王雷全. 智能客户端系统中数据同步策略的研究与实现[J]. 计算机工程与设计,2007,28(2):351-353.

[8] 王玉标,饶锡如,何盼. 异构环境下数据库增量同步更新机制[J]. 计算机工程与设计,2011,32(3):948-951.

[9] 孟雷,孙彦杰. 基于 P2P 的异构数据库数据同步研究[J]. 山东大学学报,2008,43(11):61-66.