第34卷 第19期 计 算 机 工 程 2008年10月
Vol.34 No.19 Computer Engineering October 2008
·软件技术与数据库· 文章编号：1000—3428(2008)19—0061—03 文献标识码：A 中图分类号：TP391

# 基于任务的数据交换平台

贾燚星，陈梦东，刘连忠
(北京航空航天大学电子政务研究所，北京 100083)

**摘 要：**上下级单位以及同级单位之间的数据交换日渐频繁，这些单位采用的数据库多种多样，数据定义在语义、内容上存在冲突，需要交换的数据格式并不固定，随着业务的变化而变化。该文设计并实现了一个基于任务的数据交换系统，以端到端交换模型为基础，使其可以支持多种数据库之间数据交换和并发的数据交换任务，并通过配置满足交换内容变化的需求。

**关键词：**数据交换；交换引擎；数据抽取；数据加载

# Task Based Data Exchange Platform

JIA Yi-xing, CHEN Meng-dong, LIU Lian-zhong
(Institute of E-government, Beijing University of Aeronautics & Astronautics, Beijing 100083)

**【Abstract】**Data exchange becomes more and more frequent between different departments. But these department's databases are varied. They have different data formats, so there exist conflicts both in semantics and content. In addition, the data need to be exchanged are not fixed, possibly changes along with the service change. This paper proposes a data exchange platform, which takes the end-to-end exchange model as the foundation, and improves it to support the concurrent data exchange ability. It also can satisfy the unfixed format data exchange.

**【Key words】** data exchange; exchange engine; data extraction; data load

## 1 概述

关于组织、干部管理业务，人们已经开发了大量的软件，如干部信息系统、机关人事管理系统、专家信息管理系统、干部管理辅助决策系统、机构管理系统、领导干部查询系统，举报系统等。这些系统基本上都是各单位根据其具体应用自主开发的，没有统一的标准，数据库种类多，既有Oracle, DB2, SQL Server等大型数据库，也有Mysql, Foxpro等小型数据库，数据定义也并不统一，形成众多信息孤岛。为了加强全国组织系统计算机网络信息化建设，实现全国组织部门之间的数据交换，结合新形式下干部人事管理工作的实际，在新修订的基于XML的信息交换标准[1]的基础上，本文开发出一个数据交换平台，以完成全国组织、干部、人事管理系统间的信息交换工作。

该平台不仅可以实现上下级单位间的数据交换，也可以实现平级单位间的数据交换，此外该平台还具有多任务处理能力，能同时与多个对等交换平台进行数据交换工作，同时，对于扩展的数据交换应用可以通过简单的定制来实现。

## 2 相关技术

对于数据库间的数据交换，之前采用较多的是端到端的交换方法，每两个需要交换的系统间建立一对一的互相转化模式[2]，将本地数据库的内容转换成特定的数据格式并提交给目标应用程序完成数据交换[3]。该方法一般采用一些通用的数据库访问技术，如JDBC, ODBC, OLEDB等，由于此类方法针对性强，因此其具有较高的转换与交换效率。但是存在以下问题：

(1)异构数据源在数据存储格式上差异较大，如字段名称、字段类型、字段精度定义不同，为了实现彼此间的交换，如利用通用的数据库访问技术，则需要程序员编写大量代码，这些代码仅仅能实现特定的数据格式间的转换，一旦数据格式内容发生变化，程序就要重新编写，给实现和扩展数据交换应用带来了不便。

(2)根据上下级单位之间的关系，可以方便地知道交换双方数据库的结构，建立起转换关系，但是在平级单位之间获得对方的数据结构可能比较困难，例如开发者可能不是很清楚“函件字号”与另一要交换的数据源的哪个字段的信息对应，此时就不能用这种端到端的交换方式交换数据。

各DBMS厂商针对其自身的数据库也开发出了具体的交换工具，这些工具充分考虑到了他们所支持的数据库的特点，针对性比较强，但是此类数据库交换工具一般要求是指定数据库间的交换，如Oracle到Oracle, SQL Server到SQL Server，甚至某些工具还要求改变应用系统的数据库结构，因此，这些交换工具应用在多数据库情景下的数据交换有较大难度。

集中式的数据交换方案也是现在采用较多的一种数据共享与交换的方法，但是对于电子政务中现存的系统而言，采用此种方式花费巨大，且一旦一个系统的数据库模式发生变化，或者有一个新的系统要加入进来，就必须重新调整全局模式[4]，增大维护的工作量，并不适合各地组织人事部门间的数据交换。

## 3 基于任务的数据交换平台

鉴于以上问题，本文开发了一种基于任务的数据交换平台，系统关联如图1所示。同一单位内部的不同应用之间通

**基金项目：**国家“863”计划基金资助项目“面向全国组织、干部系统的应用集成中间件平台及其应用”(2005AA113040)
**作者简介：**贾燚星(1983－)，男，硕士，主研方向：电子政务，数据交换；陈梦东，副教授、博士；刘连忠，副教授
**收稿日期：**2007-12-29 **E-mail：**jiayixing_@163.com

过本单位的交换平台实现数据的交换，在不同单位之间，通过两单位间的数据交换平台进行数据的交换。一个交换平台可以同时与多个交换平台进行数据交换。交换平台通过适配器可以支持多种数据库系统。

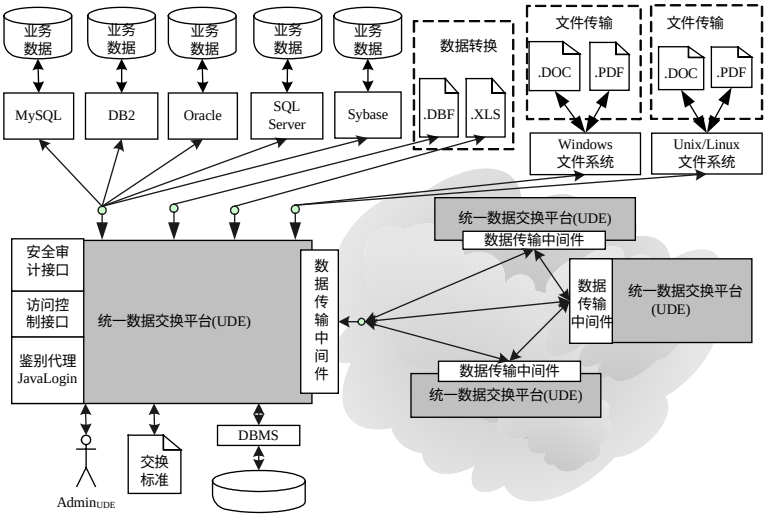


**图 1　交换平台关联**

从本质上讲，所有数据交换平台所要做的事情就是抽取转换数据、传输数据和加载数据。对应于此，本文将其定义为 3 种类型的任务，分别为抽取任务、加载任务和传输任务。根据任务的特点，可以将此 3 类任务分为以下两种情况：

(1)周期性任务。一个应用系统需要另一个应用系统定期地从数据库中取出数据，并进行交换，具体体现就是下级单位向上级单位汇报的日报、周报、月报等，此类型任务只需要操作人员一次建立，以后由平台定期地自动执行。

(2)偶发任务。一个应用系统突发性地要求另一应用系统提供数据，这种任务多出现在同级单位之间接到更上级通知要求协作时，这些任务仅执行一次。

从功能上划分，数据交换平台共有 10 个部分构成，分别是标准管理、任务管理、任务定制、策略定制、规则定制、交换引擎、数据抽取、数据加载、适配器和传输部分构成。根据这些功能的特点，将这几部分分为两类：

(1)静态部分：

1)标准管理负责管理系统要用到的所有内容格式标准。标准在整个数据交换模型中起桥梁的作用，它是各个部分工作的基础。

2)策略定制就是用户定义具体交换策略，如增量抽取策略、范围策略、查重策略等。在建立任务的过程中，用户可以选择定制过的策略用于本次任务的执行。

3)任务管理部分负责定制具体的交换工作，可以定制抽取、加载和传输任务。

4)转换规则定制用于定义具体的不同格式数据之间的转换细节。

(2)动态部分：

1)数据交换引擎是系统运作的中心，负责对任务的解析、执行调度以及数据的验证等。

2)数据抽取负责按照标准的定义从数据库中采集所需数据。

3)数据加载负责把经过交换引擎处理过的数据加载到数据库中。

4)传输部分负责把数据或者文件传递到指定的位置。

交换平台结构如图 2 所示。

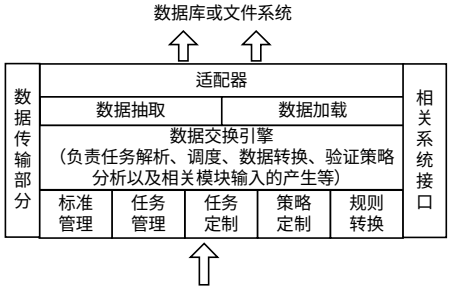


**图 2　交换平台结构**

## 4　数据交换过程

进行数据交换的第 1 个阶段即协商阶段：交换数据的双方协商要交换数据的内容，由交换提出方提出所要数据内容，然后数据提供方判断是否可以提供对应的数据内容，当双方都确认无误时该阶段结束。在本系统中，所要交换数据用 XML 语言描述，其格式如下：

```
<?xml version="1.0" encoding="UTF-8"?>
<standard>
<head>
 <requester></requester>
 <receiver></receiver>
 <time></time>
    …
</head>
<content>
 <informationset name="基本人员信息集" code="A01">//信息集
  <informationitem name="姓名" code="A0101" type="String" length="18" codetable=""/> //信息项
…
</informationset>
<informationset name="基本职务信息集" code="A02">
 <informationitem name=""code=""type="
"length=""codetable=""/>
…
</informationset>
…
</content>
</standard>
```

其中，standard是最高级元素，其下的子标签head描述本次交换的一些相关信息，具体的要交换的数据内容在content标签中描述。信息集[1]是信息结构体系的中间层次，即信息群的下级层次，其中，name属性表示的是该信息集的中文名称；code属性表示的是该信息集所对应的代码表示。在信息集标签下是信息项标签，它是信息结构体系的最低层次，即信息集的下级层次，多个信息项共同组成一个信息集。其中，name属性为该信息项的中文名称；code为信息项的代码值；type属性为该信息项的类型；length属性为该信息项的长度；codetable属性为该信息项所引用的代码表，可空。

采用此种方式描述所需要数据，XML 文件的结构性可以方便地映射到数据库中的表结构，如可以用信息集表示一个表，用信息项表示表中的字段，可以方便地在数据库模式与 XML 模式间建立映射。

此种格式的文件对于操作人员来说，其可读性依然较差，

因此，在平台的标准管理部分中，有专门的功能用于生成交换内容格式文件，并可以自动生成配套的说明文件，说明文件是一个 word 文件，其内容是对格式文件内容的解释。

当双方确认交换内容以后，数据提供方就要建立从交换内容格式标准文件到本地数据库之间的映射，这些工作是通过交换平台提供的映射功能模块完成的，流程如图 3 所示。

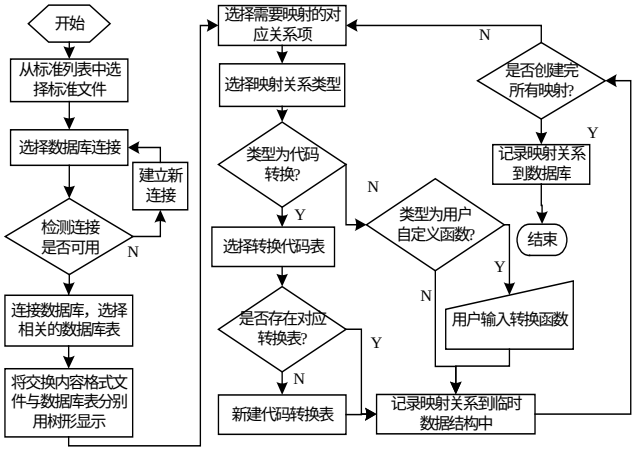


**图 3　映射建立流程**

操作人员需要从众多交换内容格式标准中选择要建立映射的标准文件，建立到具体数据库的映射，在本系统中可以解决以下几种语义冲突和数据冲突：

(1)语义冲突

1)表的冲突：表名称冲突，表结构冲突和表的约束不一致冲突等。

2)属性的冲突：属性名称冲突，默认值冲突，属性限制冲突以及属性之间多对多的冲突等。

(2)数据冲突

1)同一数据不同表现(如：不同的表示方式，精度不一致，计量单位不一致等)，数据项的拆分与合并问题等。

2)对于一些比较特殊的转换要求，本文提供了用户自定义函数，此函数有一些简单的语法，可以完成特殊的转换。

代码转换见表 1。对于代码表的不一致问题，可以采用代码表转换表的方式解决。

**表 1　代码转换**

| Standard | Database |
|---|---|
| 0 | 3 |
| 1 | 2 |
| 2 | 1 |
| 9 | 3 |

映射建立完成以后就是建立具体的抽取任务，主要是指明任务的执行时间等相关参数，在具体的抽取任务执行时，其生成的数据文件格式是一种与本地数据库关系密切的数据格式，用 XML 描述，其格式如下所示。之后采用建立映射时建立的转换规则，可以生成满足交换标准要求格式的数据文件。

```
<?xml version='1.0' encoding='utf-8'?>
<database>
  < A01 type="table">
     < A0101 type="column"></ A0101>
< A0104 type="column"></ A0104>
…
</A01>
< A02 type=" table">
  < A0203 type="column"></ A0203>
</A02>
…
</database>
```

把文件格式的转换分为两步的原因为：(1)为了验证方便；(2)为了避免转换代码与数据格式相关，从而方便地实现应用的扩展。根据传输任务定制时的要求启动传输过程，数据交换发起端接收到数据以后，就会根据要求在合适时刻启动数据加载，其过程与抽取相反。加载数据按照一定的策略进行加载处理，具体的加载方法可以有不同的实现，但是必须保证以下几点：

(1)对于要加载的数据，按先后顺序进行加载。

(2)要有较完善的主键生成机制，以此保证加载数据约束关系，如可以采用序列、自定义、指定等方法。

(3)必须保证加载任务执行的完整性，当加载过程中出现错误、突发事件(如系统死机、断电)时，必须可以提供恢复的手段。

(4)当加载过程中发现错误数据时，要记录相关信息，并反馈给操作人员，并提供相应的解决方案。

数据加载结构见图 4。

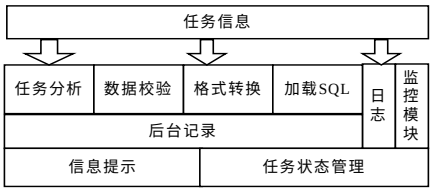


**图 4　数据加载结构**

为保证加载任务的完整性，本文采取的措施是：首先记录每条加载数据关键特征，然后执行入库操作，入库成功后进行检测，检测无误后，才执行下一条数据入库，当所有的数据都无误入库后，加载任务才正确结束，否则若中途发生错误，如断电、系统异常死机等突发事件。因为所有已经被存储到数据库中的数据都已经被记录到交换平台的后台数据库中，这些信息是不会丢失的，由于是突发事件，任务状态管理部分不会改变任务的状态，依然为正在执行状态，因此当下一次系统重新启动时，会首先检测所有任务的状态，若有正在执行的任务，说明该任务未完成，则根据平台数据库中的记录删除前面已入库的数据，当然也可以继续执行加载工作，这取决于操作人员的操作。

单位间的数据交换的一个特征就是其时间分布不均，其交换高峰一般集中在每个月的月末，此时会发生多个单位同时提出进行数据交换的要求，特别是级别越高的单位这种现象越明显，因此，交换平台必须具有任务的调度能力，对所有的交换任务进行合理的调度，避免因交换任务过多使系统资源紧张。所有这些均由交换引擎的任务调度器完成，交换引擎的结构如图 5 所示。每种任务都有专门的处理进程进行处理，在平台中反映为抽取任务处理进程阵列、加载任务处理进程阵列和传输任务处理进程阵列，所有这些进程的数量并不固定，而是根据任务的到达情况动态调整。但是 3 类处理进程数量之和是确定的，由系统配置文件根据系统硬件资源状况确定，通过对可并发任务总数进行限制，从而避免因任务过多导致系统负载过大。调度器会根据任务的性质进行合理的调度，调度算法以文献[5]中的算法为基础，保证任务在截至期限之前完成，并尽量提高系统资源的利用效率。

（下转第 66 页）