ETL 工作流过程建模与可视化开发①

ETL Workflow Process Modeling and Visual Development

龙昱磊 (湖南大学 软件学院 湖南 长沙 410082)

摘 要: ETL 工作流过程建模需要明确数据中各个字段的对应关系和实现过程中各个步骤的因果关系。本文针对来自平面数据源经过抽取、转换和加载到数据仓库过程中的若干问题进行分析，运用过程建模的概念模型和逻辑模型定义和描述其中的各种关系，以此来设计较好的 ETL 过程的解决方案，并给出具体的实现方法。

关键词: 工作流 ETL 概念模型 逻辑模型 数据仓库 平面数据源

1 引言

1.1 基本介绍

由于信息技术的发展，独立、零散的办公自动化和计算机应用不能满足日益灵活多变的工作需要，激烈的商业竞争需要综合的、集成化的解决方案。工作流技术作为一种对常规性事务进行管理、集成的技术，它所具有的柔性 (flexibility)、集成性(integration)、重用性(reusability)和可扩展性(scalability)等等的特点，使其受到广泛的重视。工作流是将工作流程中的工作组织在一起的逻辑和规则，以恰当的模型进行表示并对其实施计算的计算模型。工作流要解决的主要问题是：为实现某个业务目标，在多个参与者之间按某种预定规则自动传递文档、信息或者任务。

ETL 过程即数据抽取(Extract)、转换(Transform)、装载(Load)的过程。ETL 是 BI 的核心，它按照统一的规则集成并提高数据的价值，负责完成数据从数据源向目标数据仓库转化的过程；ETL 工作量占整个挖掘工作量的 60%以上，是实施数据仓库的重要步骤。ETL 过程的目的是将企业中分散、零乱、标准不统一的数据整合到一起，为企业的决策提供分析依据。ETL 首先是“抽取”：将数据从各种原始的业务系统中读取出来；其次“转换”：按照预先设计好的规则将抽取得数据进行转换，使本来异构的数据格式统一。最后“装载”：将转换后的数据增量或全部地导入到数据仓库中。

把工作流过程建模技术引入到 ETL 过程开发之中，可以改进和优化整个 ETL 流程，提高工作效率，实现更好的过程控制以及提高流程的柔性。

1.2 文章安排

本文通过针对平面数据源向数据仓库转换加载的实施，探讨在 ETL 工作流设计和实施过程中，如何设计较好的 ETL 解决方案，解决实现数据仓库 ETL 实施过程出现的问题。本文第 2 节介绍工作流参考模型。第 3 节着重分析数据转换过程中的问题，并建立相应的概念模型和逻辑模型。第 4 节给出针对 ETL 模型的具体实现步骤和实现方法。第 5 节给出相应的结论。

2 工作流参考模型

工作流参考模型来源于对工作流程序结构的分析，确定结构中的接口，使不同组件在不同的结构层次上协同工作。工作流管理联盟(WFMC)1994 年发布的工作流参考模型约定了工作流管理的规范。工作流参考模型包括六个基本组件和六个基本接口，其中六个组件的功能分别是:

(1)工作流执行服务器(Workflow Enactment Services)：工作流执行服务器为过程实例和活动提供运行环境，负责解释和激活过程定义，与过程所需的外部资源进行交互。在模型中，过程与活动控制逻辑间有一个逻辑上的分离，活动控制逻辑构成工作流执行服务器；过程与应用工具间、与终端用户任务间也有一个逻辑上的分离，应用工具和任务建立起对每个相关活动的处理。

(2)过程定义 (Process Definition)： 设计活动和最后的过程模型输出，称为过程定义。在运行时期过程定义可以被工作流引擎解释。WFMC 在此部分作

① 收稿时间:2008-07-31

了以下两个方面的工作：

① 提出了一个元模型，可以用来表示过程定义中的对象、对象间的关系和属性。这个元模型为不同的产品间的过程定义相互转换奠定了基础，并形成了一套转换格式。

② 工作流系统间或工作流系统与过程定义组件间的 API 调用，提供了公共的方法来访问工作流过程定义。

(3) 工作流客户端功能(Workflow Client Functions)：在工作流模型中，通过客户端应用程序与工作流引擎之间的定义良好的接口进行交互。在这个接口中包含任务表——由工作流引擎分配给用户的任务序列。同时，可以为从工作流应用程序到工作流引擎和任务表的访问提供应用程序接口。

(4) 应用程序调用功能(Invoked Application Functions)：可以被调用的应用都应该是提供了工作流引擎接口的应用程序，用以对应用数据进行处理。

(5) 工作流协同能力(Workflow Interoperability)：异种工作流引擎间的协同工作，可能需要在工作流引擎间传递应用程序调用信息，或者作为运行时期数据交换的一部分，或者通过在过程定义阶段后导入过程定义来实现，以实现多个工作流运行服务的交互。

(6)系统管理(Systems Administration)：通过公共的接口，使几个不同的工作流执行服务器可以对工作流的状态进行共享、管理和监视。

3 问题建模

ETL 工作流过程主要针对工作流参考模型中的过程定义这一组件进行功能的引用和扩展。文献 2 提出了 ETL 工作流过程的元模型，即一个 ETL 活动的概念模型，该模型定义了 ETL 活动的实体、关系和属性，同时还形成了一套转换格式，如图 1 示。

同时，Alkis Simitsis 在文献 3 中接着提出 ETL 活动的逻辑模型，该模型着重定义数据流从源到数据仓库过程中一系列的活动，如图 2 示。

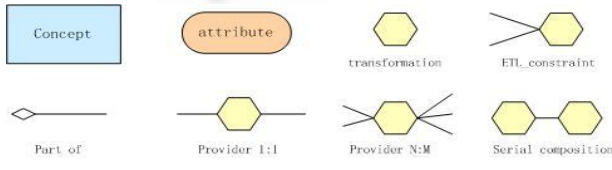

图 1　概念模型定义图

通过对 ETL 的概念模型和逻辑模型的定义，ETL 过程的数据抽取和转换可以用工作流过程建模技术进行可视化的开发。

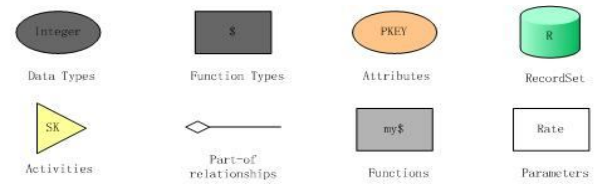

图 2　逻辑模型定义图

在实际的数据抽取和转换的过程中，可能会碰到许多异想不到的问题，比如许多数据源格式不一样，文本格式的数据各个字段之间分隔符有些为空格，有些为“|”等；数据源字段与目标数据库字段格式不匹配；

数据源中有一些字段为空值；数据源中的字段在目标数据库中不存在等等。因此，在设计数据抽取时，首先要考虑以下问题：

①数据源和目标数据库格式是否一致?

②从数据源中要访问哪些文件，如何访问?

③如何从数据源中提取相应的字段?

在设计数据转换时，由于数据源之间往往存在着不一致的问题，因此数据转换必须做到数据名称及格式的统一，同时对于源数据库中可能不存在的数据需要创建新的数据逻辑视图并进行相应的转换。概括起来需要如下的处理：

①直接映射：数据源字段和目标字段长度或精度相同，则无需做任何处理。

②字符串处理：从数据源的字符串字段中获取特定信息作为目标数据库的某个字段，则对字符串的操作有类型转换、字符串截取等。由于字符类型字段的随意性也可能造成脏数据的出现，所以在处理这种规则的时候，需要异常处理。

③字段运算：对于数值型字段来说，有时数据源的一个或多个字段进行数学运算而得到目标字段，则需要某些字段运算。

④空值判断：对于数据源字段中的 NULL 值，可能在目标数据库进行分析处理时会出问题，因此必须对空值进行判断，并转换成特定的值。

⑤日期转换：由于目标数据库中的日期类型格式是统一的，所以对数据源字段的日期格式需要相应的转换。

⑥聚集运算：对于目标数据库事实表中的一些度量字段，通常需要通过数据源一个或多个字段运用聚集函数得来的，比如 sum,count,avg,min,max，因此需要做相应的转换。

⑦既定取值：这条规则对于目标字段取一个固定的或是依赖系统的值，而不依赖于数据源字段。

基于对上述问题的分析，针对若干平面数据源进

行数据的转换可能涉及以上若干类型。对于特定的文本数据，其转换过程如图 3 所示：

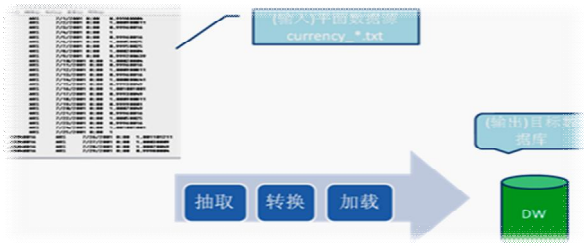

图 3　转换过程图

如图 3 所示，平面数据源为若干文本文件 currency_*.txt，它包括四个字段：CurrencyAlternateKey，TimeAlternateKey，AverageRate，EndOfDayRate；目标数据库为事实表 FactCurrencyRate，它包括四个字段：CurrencyKey，TimeKey，AverageRate，EndOfDayRate。

其中，数据源字段 CurrencyAlternateKey 与目标数据字段 CurrencyKey 对应，但键值不同；

同时，数据源字段 TimeAlternateKey 与目标数据字段 TimeKey 对应，但键值不同；

数据源字段 AverageRate 与目标数据字段 AverageRate 对应，但键值相同；

数据源字段 EndOfDayRate 与目标数据字段 EndOfDayRate 对应，但日期格式不同。

因此，要想从文本 currency_*.txt 的数据内容加载到数据库表 FactCurrencyRate 中，则需要考虑四个相应字段的对应关系。根据参考文献[2]中提出的 ETL 概念模型的描述，对上述问题建立概念模型，如图 4 示：

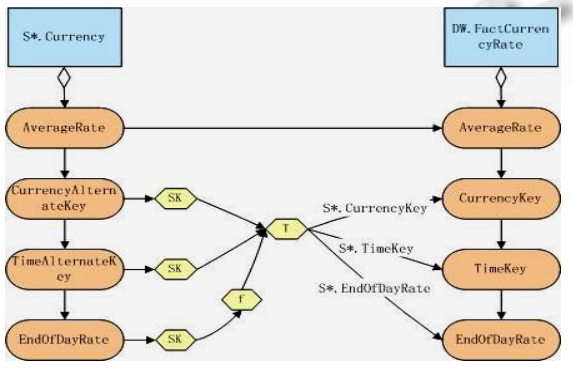

图 4　概念模型图

可以看到，图 4 清晰地展示了文本数据源字段与目标数据库字段之间的转换关系。其中，字段 AverageRate 可以相互直接映射，CurrencyAlternateKey 和 TimeAlternateKey 需要代理键(SK)的字段转换，字段 EndOfDayRate 需要代理键(SK)的字段转换，同时也需要日期格式的日期转换。

建立概念模型可以很清晰地反映完成一项 ETL 转换要做什么，但不能反映 ETL 过程转化的流程和具体实施的事务，即如何做。因此可以根据参考文献 3 中提出的 ETL 逻辑模型进行建模，可以清楚地描绘 ETL 流程如何具体实施及实现，如图 5 所示：

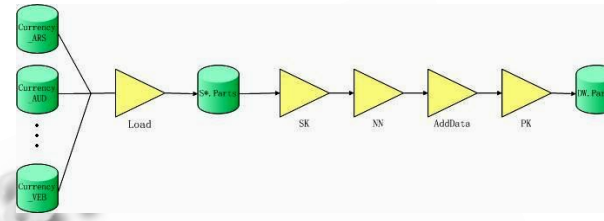

图 5　逻辑模型图

图 5 清晰地反映了源数据通过加载(Load)、代理键转换(SK)、空值判断(NN)及既定取值(AddData)等工作流处理流程。

4　ETL过程组件模块的实现

通过对一个 ETL 过程的分析和模型的建立，可以很清楚的明白要做什么、如何做以及做的步骤。以下只需要选择适当的 ETL 工具具体实现上述模型。基于微软的 SSIS(SQL Server 集成服务)的强大性能，IS 包成为理想的 ETL 实现的工具。运用 Integation Service 项目解决方案中的控制流组件和数据流组件，一个针对上述 ETL 过程的 IS 包可以方便地建立起来。

以下方法可以从平面文件到数据库表的转换和加载：

(1)建立 IS 包，其实现的静态方法 CreatePackage 及相应代码如下：

```
Package p = new Package();
p.PackageType
DTSPackageType.DTSDesigner90;
p.Name = Name;
p.Description = Description;
p.CreatorComputerName=System.Environ
ment.MachineName;
p.CreatorName
System.Environment.UserName;
```

(2)针对 IS 包添加 OLEDB 和平面文件的数据连接管理器 AddConnectionManagers：

```
ConnectionManager DW=package.Connecti
ons.Add("OLEDB");
ConnectionManager flatFile=package.Conn
ections.Add("FLATFILE");
```

(3)创建数据流任务 AddDataFlowTask：

```
private MainPipe dataFlow;
TaskHost th=package.Executables.Add("DTS.Pipeline") as TaskHost;
th.Name = "DataFlow";
th.Description = " DataFlow Task";
dataFlow = th.InnerObject as MainPipe;
dataFlow.Events = pipelineEvents as wrap.IDTSComponentEvents90;
```

(4) 为数据流任务添加平面数据源 AddFlatFileSource：

```
flatfileSource=dataFlow.ComponentMetaDataCollection.New();
```

(5)添加 Lookup 组件 AddLookup:

```
lookup=dataFlow.ComponentMetaDataCollection.New();
```

(6)添加数据流目标 AddOleDBDestination:

```
oledbDestination=dataFlow.ComponentMetaDataCollection.New();
```

(7)完成。

以上是实现一个数据流转换的具体步骤和实现方法，而通过组件化技术可以很直观地实现这一数据流任务。数据流任务封装数据流引擎，该引擎在源和目标之间移动数据，并可方便地在移动数据时转换、清除和修改数据。将数据流任务添加到包控制流使得包可以提取、转换和加载数据。

SSIS 提供三种不同类型的数据流组件：源、转换和目标。源从数据存储区中提取数据；转换修改、汇总和清除数据；目标将数据加载到数据存储区，或创建内存中的数据集。IS 包中的数据流用下列不同类型的数据流元素构造而成：提取数据的源、修改和聚合数据的转换、加载数据的目标以及将数据流组件的输出和输入连接为数据流的路径。图 6 为设计 IS 包的数据流图。

数据流任务是通过数据流设计器来实现：即平面数据源通过查找组件实现两次代理键转换（SK），把源文件的相应键值的内容加载到目标数据库 DW 中。通过控制流任务和数据流任务设计的实现，一个 ETL 过程相应的完成。从运行的进度结果显示是成功的，同时在数据库中也确认新添加了相应的记录。

5　结论

本文主要通过对平面数据源（文本文件）如何抽取、转换和加载到目标数据库的过程的分析来探讨和建立 ETL 工作流过程的概念模型及逻辑模型，两种模型的定义在文献[2]和[3]中有详细的分析，此处对两种模型的定义作相应的扩展和应用。最后通过 ETL 工具

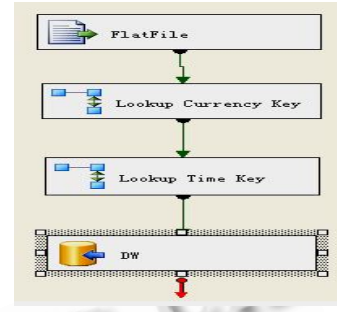

图 6　数据流图

给出该模型的实现，由于 SSIS 的可视化、组件化技术和强大的功能，针对模型中出现的各种数据不一致的问题基本能得到圆满的解决。

参考文献

1 Tang ZH, MacLennan J. Data Mining with SQL Server 2005. Wiley Publishing, Inc., Indianapolis, Indiana USA .2005: 303－327.

2 Vassiliadis P, Simitsis A, Skiadopoulos S. Conceptual Modeling for ETL Processes. Proc. 5th ACM Int'l Workshop on Data Warehousing and OLAP. 2002: 14－21.

3 Simitsis A. Mapping Conceptual to Logical Models for ETL Processes. Proc. 8th ACM Int'l Workshop Data Warehousing and OLAP, 2005: 67－76.

4 Simitsis A, Vassiliadis P, et al. Optimizing ETL Processes in Data Warehouses. Proc. 21st IEEE Int'l Conf. Data Eng., 2005: 1084－4627.

5 Simitsis A, Vassiliadis P. A Methodology for the Conceptual Modeling of ETL Processes. Proc. Decision Systems Eng., 2003 : 305－316.

6 Simitsis A， Vassiliadis P, et al. State-Space Optimization of ETL Workflows. IEEE Transactions on Knowledge and Data Eng, 2005, 17(10):1404－1419.

7 罗海滨.工作流技术综述.软件学报,2000,11 (7): 899－907.

8 范玉顺.一种提高系统柔性的工作流建模方法研究.软件学报,2002,13(4): 833－838.

9 蒋国银.工作流过程建模理论综述.计算机系统应用,2006,15(3):90－93.

10 张成.基于构件库/工作流的可视化软件开发.计算机工程与应用,2008,44(10):82－87.