一种基于总线模型的数据清洗方法

杨梦宁 赵 鹏 张小洪 李 朋
(重庆大学软件学院 重庆 400044)

摘 要 数据清洗是保证数据质量的重要环节。已有的清洗方法往往过于依赖特定应用,不容易得到重用。从提高数据清洗方法的可重用性和可扩展性的角度出发,提出一种基于总线模型可复用的数据清洗框架。具有相对独立功能的清洗工具以组件的形式,通过适配器挂接到清洗总线上,通过总线控制对清洗组件实现清洗。最后用具体应用来描述基于总线模型的数据清洗方法的工作流程。实践结果证明该方法具有良好的性能和应用价值。

关键词 数据清洗,总线模型,组件,可复用
中图法分类号 TP391 **文献标识码** A

Data Clean Method Based on Bus Model

YANG Meng-ning ZHAO Peng ZHANG Xiao-hong LI Peng
(School of Software Engineering, Chongqing University, Chongqing 400044, China)

Abstract Data cleansing is an important part for ensuring data quality. The existing cleaning methods are often too dependent on a specific application, can not be reused. In order to improve the reusability and scalability of the clean method, a data clean framework was build which is based on bus model and reusable. The data clean tool which has independent clean function is registered on the bus through the adapter. The clean function is finished by calling the clean components which is registered on the bus. Finally, how the method works in the really scene was described. The method was proved has good value of application.

Keywords Data clean, Bus model, Component, Reusable

1 引言

数据库系统、决策支持、数据挖掘都要求高质量的数据,而数据质量问题在企业应用系统中普遍存在,主要表现为数据的正确性、一致性、完整性、可靠性等。造成这些问题的主要原因有:①业务系统在录入数据时数据源的复杂性,其中包括滥用缩写词、惯用语、数据输入错误、数据中的内嵌信息错误、重复记录、丢失值、拼写变化、不同的计量单位和过时的编码等;②在建设数据集成和数据仓库时,由于原有业务系统的差异也会造成数据混乱冗余。这些存在质量问题的数据在提供决策支持时,很可能提供错误的信息,并且不能被新的业务系统所使用[1]。针对上述数据质量问题,研究人员提出了多种数据清洗解决方案,主要表现在数据清洗的系统框架、模型、语言、方法、技术、构件设计、工具开发等方面的研究[2-4]。然而,面对实际项目中的数据清洗工作时,由于现有数据清洗方法繁多且多与具体领域紧密结合,因此需要结合项目重新开发一套数据清洗工具。另外,从软件工程中需求的角度分析,作为数据清洗系统除了应满足业务的要求之外,还应当保证满足通用性、可集成和扩展性、可复用性和易维护性等需求,但是现有实现方法不尽人意。为解决现有数据清洗领域的问题以及满足现代软件工程理念的要求,本文提出一种基于总线模型的数据清洗方法,通过总线的方式集成清洗组件,由清洗总线选择调用清洗组件以完成数据清洗,同时可以实现组件的可复用性以及清洗系统的可扩展性。

2 基于总线模型的数据清洗方法

2.1 整体框架

从软件工程角度考虑,一个好的软件体系结构应该具有好的可复用性。软件总线模型是为了解决复杂软件系统的集成问题而提出的系统解决方案。软件总线的概念来源于计算机硬件技术,它是硬件总线的虚拟和映射,任何符合一定标准的软件模块都可以通过适配器以插件方式获得软总线的支持,与总线上的其它部件相互通讯、协调与控制,从而完成相应的功能。各种符合软总线接口规范的构件在软总线上可以实现"即插即用",从而易于实现大规模软件系统的集成和定制。总线型软件体系结构具有如下优点:①良好的可扩展性;②系统体系结构清楚,因此可以降低软件开发难度;③高内聚、低耦合的特点,因此易于系统扩展和维护[5]。

系统采用上述的软件总线模型,集成数据清洗组件,各个清洗组件之间高度独立,不存在依赖关系。整体框架如图1

到稿日期:2009-10-20 返修日期:2009-11-23 本文受国家自然科学基金(60975015),重庆市科委科技攻关计划项目(2009AC2057),重庆市科委自然科学基金(2009BB2364),重庆大学青年骨干教师创新能力培育基金资助。
杨梦宁(1980-),男,讲师,主要研究方向为数据挖掘等,E-mail:mnyang@cqu.edu.cn;赵 鹏(1986-),男,硕士生,主要研究方向为数据挖掘等;张小洪(1973-),男,副教授,主要研究方向为图像处理等;李 朋(1985-),男,硕士生,主要研究方向为数据挖掘等。

所示。从整体来看，系统的输入为脏数据，输出为经过清洗总线上的组件清洗过的结果数据。各个清洗组件通过总线适配器挂接到数据清洗总线上，适配器负责总线与组件间的通信，配置管理用于对清洗组件进行装入卸载，总线控制器负责总线的状态管理。

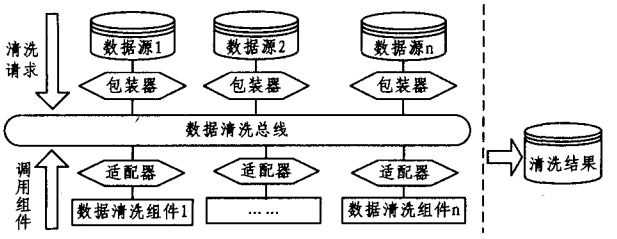

图 1　整体框架结构图

2.2　系统模块分析

总线模型设计和实现的核心问题主要体现在 3 个方面：①定义总线的接口和协议，这样符合协议的组件才可即插即用；②组件的调度与管理；③组件间的通信管理[6]。

系统采用 XML 数据文件作为数据传递和系统集成的媒介，通过预定义各系统模块之间集成信息的 DTD（全称为 Document Type Definition，用来定义 XML 文件元素）标准描述来作为清洗组件集成时数据传递转换结构的标准规范，从而保证各个模块之间交换的 XML 数据文件能够相互解析。

2.2.1　包装器

数据源不同，造成待清洗数据的数据格式也不同。数据类型大体上分为 3 种类型：单个数据值、单行数据和多行数据[5]。包装器对待清洗的数据，首先对其进行分析，以检查脏数据是否需要清洗，需要清洗的数据类型有哪些。原始数据的格式并不能符合总线的要求，包装器针对不同的数据来源，对其先按照总线规范进行封装，以满足清洗总线的需求。其封装的格式 DTD 描述如图 2 所示，其中 cType 表示待清洗数据的类型，row 表示多行的数据集。

```
〈! ELEMENT Data (row*,cType*)〉
〈! ATTLIST row id CDATA ""〉
〈! ELEMENT row(cell*)〉
〈! ATTLIST cell type CDATA ""
                value CDATA ""〉
〈! ELEMENT cType (#PCDATA)〉
```

图 2　脏数据封装后的 DTD 描述

2.2.2　数据清洗组件

数据清洗组件可以是针对某一类型的数据开发的，具有某种清洗功能的相对独立组件，也可以是针对此总线标准开发的清洗组件，这种情况下组件与总线的交换就不需要总线适配器了。对于组件的定义，通过 DTD 进行描述，如图 3 所示。

```
〈! ELEMENT ComponentSet (Component*)〉
〈! ATTLIST Component id CDATA ""
                      name CDATA ""
                      priority CDATA ""〉
〈! ELEMENT Component (cType,filename,
                      className,method)〉
〈! ELEMENT cType (#PCDATA)〉
〈! ELEMENT filename (#PCDATA)〉
〈! ELEMENT className (#PCDATA)〉
〈! ELEMENT method (#PCDATA)〉
```

图 3　组件和组件库的 DTD 描述

2.2.3　适配器

引入了适配器 Adapter 后，组件只负责完成特定清洗功能，从逻辑上独立于系统的逻辑[8]。清洗组件能插拔到不同的系统，从逻辑上只与特定清洗功能相关联，这就可以把组件看作是一个相对稳定不变的单位。而为了能在不同的系统中使用又必须要能与不同的系统总线相连接，因此其变动部分的工作就分配给适配器来完成。

适配器针对清洗组件和总线之间接口的输入和输出做转换，以使清洗组件能与总线之间正常交换。针对总线向组件发送清洗请求，适配器将其解析为组件要求的输入格式，再传递给组件处理。

2.2.4　数据清洗总线

在数据清洗总线中定义总线的接口和协议、清洗组件挂接的平台。为了方便快速查询合适的清洗组件，在总线初始状态时，将组件库中的组件的引用根据清洗类型以哈希表的形式缓存到内存中，每种待清洗的数据类型，对应一个清洗组件的链表，如图 4 所示。

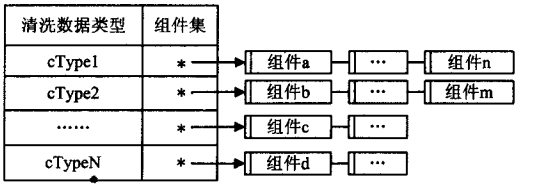

图 4　组件集在内存中的 Hash 表结构

清洗总线将根据传入的待清洗数据类型，从内存中的哈希表中选择清洗组件，并将清洗组件按照其优先级进行排序，然后通过适配器对组件进行调用。

2.3　清洗流程描述

结合上文中对总线模型清洗框架的描述，本节将对清洗具体流程进行描述，如图 5 所示。

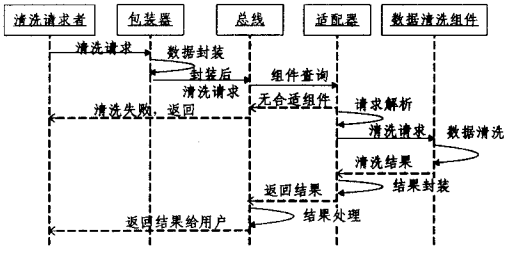

图 5　数据清洗方法时序图

①选择待清洗的数据源。将待清洗的脏数据提交给对应的数据包装器，将待清洗的数据封装为符合总线规格的 XML 报文。

②将封装后的报文传入数据清洗总线。数据清洗组件对其报文进行解析，查询所适合的清洗组件。

③将清洗组件按照优先级进行排序，然后通过适配器依次进行调用。

④适配器从报文中读取符合其清洗要求的数据，按照清洗组件所需要的格式要求，将解析后的数据传入对应的清洗组件。

⑤清洗组件进行清洗处理。

⑥清洗结果通过适配器转化为总线所能接受的报文格式，并传递给总线。

⑦清洗总线再将报文解释为用户所需要的数据传递给用

户。一次数据清洗过程结束。

3 应用场景

实际应用中，对某公司的客户数据进行清洗。由于种种原因，客户数据存在联系地址不规范、电话号码表示形式不规范、邮编与地址不一致等错误，因此对于这些数据都需要进行清洗。这些客户数据来源不同，针对不同的来源需要有不同的数据包装器。

在清洗的过程中，我们采用基于总线模型的框架，不同类型的清洗方法被看作相对独立的清洗组件，挂接到总线上。这样可以方便系统的扩展和对清洗组件的复用。根据所需要清洗的数据类型，有选择性地调用清洗组件。图 6 描述了应用中数据清洗的流程。

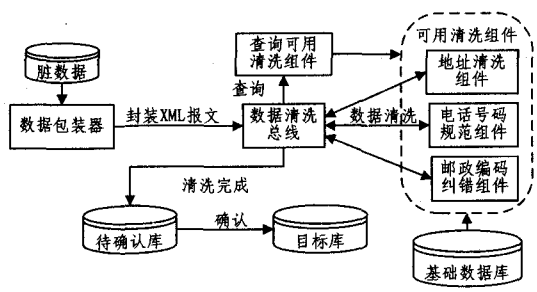

图 6 实际应用中的数据清洗流程

```
〈? xml version="1.0" encoding="UTF-8"?〉
〈Data〉
    〈row id="1001"〉
        〈cell〉
            〈type〉address〈/type〉
            〈value〉重庆沙坪坝沙正街学府小区 1-6-1
            〈/value〉
        〈/cell〉
        〈cell〉〈type〉level1〈/type〉
            〈value〉〈/value〉〈/cell〉
        ……
        〈cell〉〈type〉level6〈/type〉
            〈value〉〈/value〉〈/cell〉
        〈cell〉〈type〉phone〈/type〉
            〈value〉023-65108029〈/value〉〈/cell〉
        〈cell〉〈type〉zipcode〈/type〉
            〈value〉400043〈/value〉〈/cell〉
    〈/row〉
    〈row id="1002"〉
        …
    〈/row〉
    ……
    〈cType〉address〈/cType〉
    〈cType〉phone〈/cType〉
    〈cType〉zipcode〈/cType〉
〈/Data〉
```

图 7 脏数据封装后的报文

在本应用中，可用的清洗组件有地址清洗组件、电话号码规范组件、邮编纠错组件。地址清洗组件主要完成中文地址的补全、纠正错别字、地址别名匹配等功能；电话号码规范组件主要完成对电话号码按照规定的格式进行规范，并且可以通过地址判断区号是否有效；邮编编码纠错组件主要完成根据地址来补全纠正邮政编码。

实际应用中具体的清洗过程描述如下：

①根据应用领域，对脏数据按照总线所要求的报文格式进行封装。报文格式如图 7 所示。

②总线接受传递过来的报文，并对报文进行分析。根据报文中需要清洗的数据类型，查找可用的数据清洗组件。经过查询后，发现可用组件有地址清洗组件、电话号码规范组件、邮政编码纠错组件，并按照组件的优先级对其进行排序。组件的优先级可以通过修改配置文件进行调整。

③将待清洗的报文提交给地址清洗组件的总线适配器。总线适配器将报文中的待清洗的地址信息提取出来，提交给地址清洗组件。对地址进行清洗后，将地址分解成省、市、区县、街、门址等字段返回给适配器。总线适配器将其填充到 XML 报文，返回给清洗总线。

④将完成地址清洗的报文提交给电话号码规范组件的适配器。其总线适配器提取报文中的省、市、区县字段，组织成完整地址信息，提取电话号码字段，将其提交给电话号码规范组件。经过清洗后，电话号码返回给总线适配器，总线适配器再将其填充到 XML 报文，返回给清洗总线。

⑤邮编纠错组件采用和上一步同样的方式进行，将电话号码改为邮政编码。

⑥完成上述清洗过程后，得到的是包含了清洗结果的 XML 报文，对报文进行解析，将其存入待确认库。经过人工确认后，存入目标库。

结束语 基于总线模型的数据清洗方法采用总线模型集成数据清洗组件，用 XML 语言作为描述语言，通过适配器实现组件与总线间的交换，实现了系统高度的松耦合。除了能在清洗组件上得到复用，对于整个清洗框架，也能结合相关领域得到复用。

参 考 文 献

[1] Rahm E, Do H H. Data cleaning: problems and current approaches [J]. IEEE Data Engineer Bulletin, 2000, 23(4): 3-13

[2] Raman V, Hellerstein J. An Interactive Data Cleaning System [C]// VLDB. 2001: 381

[3] 叶舟. 基于规则引擎的数据清洗[J]. 计算机工程, 2006, 32(12): 100-103

[4] 陈伟, 丁秋林. 可扩展数据清理软件平台的研究[J]. 电子科技大学学报, 2006(01)

[5] 王浩. 数据仓库环境下以用户为中心的数据清洗过程模型[J]. 计算机科学, 2003, 31: 55-57

[6] 杨芙清, 梅宏, 李克勤. 软件复用与软件构件技术[J]. 电子学报, 1999, 127(2): 68-75

[7] 袁占亭, 张秋余, 张冬冬, 等. 基于软件总线技术的软件开发[J]. 计算机工程, 2005(01): 105-107

[8] 徐正权, 潘晓波. 基于 Adapter 的软件总线体系结构[J]. 华中科技大学学报: 自然科学版, 2005(05): 10-12