计算机科学2002Vol. 29№. 9

基于 XML 数据字典的元数据对象分级控制模型的研究

Research on Meta Data Model based on XML Data Dictionary

徐 鹏 谢晓芹 王克宏

(清华大学计算机系 北京100084)

Abstract In a Web-based e-Government and e-Commerce system, data acquisition and management is the core function of the system. Usually, the amount of data in this system is very large, and the data model is more complex. Because the system requirements and rules on data statistic are often changed, we must modify the data model and algorithms on statistics. It will lead to update the system usually. Additionaly, we must manage and store the data in the system into the database because of the importance of data. So, how to implement the functions of data acquisition and storage based on this kind of data model is the research topic for software designers. In this paper, we will reform the existing definition of data dictionary, and present a whole new dictionary for data control model for integrating data definitaion, data representation, and data checking and statistics together.

Keywords Data acquisition and management system, Data dictionary, Meta data object, Control model, XML

1. 引言

B2B 电子商务和电子政务系统中涉及的信息与在线购物系统等简单的数据采集系统中涉及的信息具有明显的差别，这些信息对于整个业务流程均产生重要的影响，信息的种类繁多，数据模型复杂，覆盖范围广，且数据正确性和安全性要求很高。数据采集和管理系统所面对的数据对象具有多维的特性[1]。每个维又划分为多个汇总层次，同一层次可以有多个元素，相互之间称为兄弟，上一层次的元素由下一层次的元素汇总而成，最低层元素为叶子元素，所有元素及其层次关系构成属性结构。各维汇总层次根据用户需求灵活定义，形成各个方向的多层次汇总，包括轻度汇总和高度汇总。

在一个普通的数据采集系统中，可以将系统生成的数据文件(DATA_FILE)看成最大的数据实体单位，也是整个数据树的根节点，这是一个多维结构的数据实体。而数据文件所包含的各个内部表单(DATA_FORM)可以作为根节点的子节点，内部表单中包含具体的科目。内部表单的划分将根据构成数据文件的各部分数据之间的包含关系和耦合程度决定。而对于科目来说又可分为基本科目(DATA_ITEM)和记录科目(DATA_RECORD)，基本科目对应一个具有基本数据类型的数据条目，将作为数据树的叶子节点；而记录科目则类似于数据库中的数据记录的概念，由一系列记录构成，作为一个实体存在于内部表单中，而每个记录又是由相同的基本科目构成。

定义1 数据文件 FL 是内部表单实体的集合，其结构如下：

$$FL=\bigcup_{i=1}^{m} FM_i$$

其中 FM 代表构成数据文件的内部表单实体。

定义2 内部表单 FM 是一个二元组数据项的集合，其结构如下：

$$FM=\bigcup_{i=1}^{n} FI_i$$

其中$\forall FI_i, FI_i=IT_m \vee TB_n$。IT 代表基本数据项科目，TB 代表基于同一二维数据表结构的记录科目的集合。

定义3 数据表 TB 是记录科目的集合，同一数据表中所有记录科目具有相同数据模型定义方式。每个记录科目实体由多个基本数据项实体构成。数据表的结构如下：

$$TB=\bigcup_{i=1}^{m} RD_i$$

其中 RD 代表记录科目。

定义4 记录科目 RD 是基本数据项实体的集合，它是最小粒度的实体。其结构如下：

$$RD=\bigcup_{i=1}^{m} IT_i$$

这种树型结构以及各个元素之间的具体运算与现实操作中特定表单中各个项目的构成及相互关系相一致，只有应用系统或数据库按照这种结构进行汇总，才可以产生满足管理需求的表单数据。

在传统的应用系统设计中，往往将 MVC(Model-View-Control)三层结构中有关数据对象结构和取值类型、针对数据对象的逻辑和算术运算算法、数据对象在用户界面表示时对于相关控件的特殊配置要求等等复杂的控制信息封装在特定的类中。这样造成了系统在设计和应用过程中不易修改和调整，从而导致系统的可扩展性和可维护性较差。在企业化的数据采集系统中，一个数据源中提供的数据在数据申报和审批过程中，通常以文件的形式进行流通，此外出于管理的需要，数据还可能保存在多个不同的数据库系统中，而每个数据库的库表结构和数据字段定义又不尽相同。业务流程中不同阶段的数据汇总分系统可以按照不同的统计算法和汇总规则对数据进行分析和汇总处理，得到汇总数据，并将这些数据存储到不同数据库的不同表中，也可以作为新的记录域插入已有的数据表中。传统数据库系统中的数据字典模型的应用领域有限，且功能单一，主要为适应一维查询而设计，不能够充分满足 Web 应用环境中以数据为核心的复杂应用系统的设计需要，本文提出了针对 Web 应用系统设计过程的数据字典

徐 鹏 博士生，主要研究领域为基于 Web 的半结构化和非结构化信息的抽取，基于 Web 的电子商务系统。**谢晓芹** 博士生，主要研究领域为网络计算模式下的知识工程和构件化设计。**王克宏** 教授，博士生导师，研究方向为网络计算模式下的知识工程。

模型，以及基于这种字典模型的元数据控制模型，旨在解决以上这些问题。

2. 数据字典模型的改造

基于 Web 的数据采集和管理系统中，在数据对象的定义、操作和管理等控制模块的设计上，可以充分利用传统数据字典的设计思想，并通过对数据字典模型的改造，实现对系统数据模型的有效控制。

数据字典的思想以前一直只应用在数据库管理系统中。传统多维数据库中数据字典的定义包括：数据类型的定义、关系表结构与属性和多维数据库的层次定义。数据类型定义在综合考虑业务系统的数据特征和信息分析的数据要求后，对数据仓库中所有关系表的数据字段进行分类，并对其属性作出了统一的定义。当创建新的关系表时，各个字段的数据属性采纳相应的数据类型分类定义。数据库管理系统中的数据字典定义主要针对数据表和数据记录，为了保证数据字典数据的语义完整性，通常将约束划分为记录和数据项两级，并通过主键和外键实现数据约束规则的定义[2,3]。

面向数据库的数据字典模型经过改造后，可以同样应用于数据采集和管理系统等以数据对象操作为核心的应用系统中。通过数据字典的建立对系统的数据模型进行控制，并采用数据驱动的方法来支持信息系统的设计。面向数据库的数据字典应用时，如果没有对象数据库管理系统，则无法实现数据和操作的封装。而在数据采集和管理系统中，系统针对数据库进行的一系列操作仅仅是整个业务流程中的几个环节，因此数据字典的设计不应局限在数据库上。我们应当将数据字典作为整个系统的中介，将数据与对应的校验统计操作、用户界面中对应界面控件以及其他以数据操作为核心的功能（例如数据自动入库操作等）联系起来。

改造后的数据字典由两部分构成，分别是针对元数据建模而设计的数据字典和针对应用系统中数据自动入库功能的实现而设计的数据字典。由于以数据操作为核心的应用系统在实现过程中可能采用 Client/Server 或 B/W/D 等不同的模式进行设计，因此我们采用 XML 文件的形式来保存数据字典中对于元数据模型的所有约束信息。XML 是针对包含结构化信息的文档而设计的一种标记语言。这种新标准正在 Internet 通讯管理和数据传送领域越来越流行。应用程序之间的联接对于开发分布式系统和提供电子商务和灵活性需求来说非常重要。XML 可以将 Internet 转变为一个基于无限知识仓库的全球计算平台。最终的环境可以被看成是实现电子数据交换的强大基础架构。

基于 XML 的数据字典的一个主要特点就是不同层次上元数据模型定义以内容“块”的形式出现，每个内容块可以作为可复用的构件直接应用于其他应用系统的数据字典定义中。这意味着通过数据字典定义的信息可以很方便地满足不同目标的需求。当通过使用 XML 完成数据字典定义后，在数据字典管理系统中的信息可以变为系统设计者的一项重要资产。由于 XML 结构的特点，可以通过数据挖掘处理和其他智能化信息处理工具推断出一个应用系统中数据的规律性信息。另外，也可以发挥 XML 技术在数据国际化等多方面的优势。

基于 XML 数据字典的元数据建模需要达到的目标包括：1）. 针对一个应用系统的不同版本，建立对应的系统元数据对象模型中报表和科目的编码库，并建立编码与实际数据文本之间一一对应的关系，使得系统能够明确表示出每个基本数据科目的含义；2）. 提供各种关于数据项的统计、校验规则和状态信息；3）. 针对特定数据项的特点，实现对界面显示控件的配置；4）. 确保数据元素命名的标准化。

与针对数据库系统数据字典模型不同，新的模型对数据的语义完整性约束分为数据文件、内部表单、记录和基本数据项四级。在文件级维持（1）主键约束，每个数据文件均存在单义性的主键；（2）值约束，即数据文件中包含的不同内部表单中的数据项取值之间存在约束；（3）版本约束，在表单级维持（1）主键约束，即每个表单都存在单义性的主键，通常此主键与文件级的主键相同；（2）值约束，即表单内部包含的不同数据项取值之间存在的约束。在记录级维持（1）外键约束，如果一个表单级对象 F 中包含的数据项 Di 作为一个记录级对象 R 的主键，则称 Di 为 F 的外键（此定义与传统数据库系统中的同名概念的定义不同，以适应数据采集系统的独特要求）；（2）数量约束，即能够支持的最大记录数。在基本数据项级维持（1）值约束，对单个数据项值域的约束。

针对四级数据模型定义完整性约束信息的表达方式如下：

（1）数据项唯一标记（标识符）定义 用于说明主键和外键

内部表单 ID. 基本数据项 ID
内部表单 ID. 记录科目 ID. 基本数据项 ID

（2）数据字典版本定义

```
<!--定义数据文件版本 -->
<! ATLIST DATA_FILE VERSION_ID CDATA # REQUIRED>
```

（3）实现值约束的逻辑表达式

用于说明单个数据项或不同数据项之间的值约束。在逻辑表达式中，可以使用下列算符：<、>、=、<=、>=、<>、IN、AND、OR、NOT 以及各种算术运算符。

逻辑表达式的使用，其目的是对数据采集系统中的数据项统计规则和校验规则进行定义。具体应用系统在运行时将解析逻辑表达式，并完成相应的算术运算和逻辑判断。逻辑表达式的定义应当在数据字典的内部表单级中完成。

（4）格式描述符串

用于规定数据项的输入格式，同时用于定义数据项对应用户界面控件的事件处理方式。通过格式描述符串的使用实现了对用户输入字符类型和大小写的规定。开发者可以使用以下选项值：

- A：允许任意大写字母（非数字）；
- a：允许任意小写字母（非数字）；
- N：允许任意数字（非符号）；
- X：允许任意数字、符号、或大写字母；
- x：允许任意数字、符号、或小写字母；
- M 或 m：允许任意字母（大小写均可）、数字或符号；

为了指定特定类型字符的数目为零个或无限个，可以使用“*”。例如说明符 NN*M 则允许用户输入两个数字，后跟无限多个字母数字字符。如果希望指定特定类型字符的输入个数，可以在类型后跟一个数字定义符，例如 N5M 指定了一个数字字符后跟5个字母数字字符。

为了实现在用户界面实现中能够支持特定字符的自动添加功能，在指定输入字符格式的字符串定义中提供特定的表达方式。为了指定一个自动字符，可以在其他格式说明符字符之间插入该字符，该字符前面需要加入反斜杠（\）。当用户输入数据时，相关控件自动在特定的位置插入自动字符。假定使

用格式说明符 \(NN\)-,该格式说明符要求电话自动插入一个左括号;当用户输入两个数字之后,自动插入右括号和一个破折号。

为了体现数据字典语义的完整性和不同级别的约束条件,采用 XML DTD 定义的数据字典文件的结构和关系如下:

(1)数据字典的多维层次结构定义

```
〈!DOCTYPE DATA_FILE〉
〈!ELEMENT DATA_FILE(DATA_FORM)+〉
〈!ELEMENT DATA_FORM(DATA_ITEM|DATA_RECORD)+〉
〈!ELEMENT DATA_RECORD (DATA_ITEM)+〉
```

(2)内部表单元素属性定义

对于数据采集系统来说,首要的基础工作是创建系统中使用的各个内部表单实体。因此,首先应当建立表单属性字典,该字典中定义了被系统所使用的所有表单的属性。它以 FORM_ID 字段作为主键,其基本结构如下:

```
〈!ELEMENT DATA_FORM(DATA_ITEM|DATA_RECORD)+〉
〈!-- 定义表单 ID --〉
〈!ATLIST FORM ID CDATA #REQUIRED〉
〈!-- 定义表单名称 --〉
〈!ATLIST FORM NAME CDATA #REQUIRED〉
〈!-- 定义表单规则 --〉
〈!ATLIST FORM RULE CDATA #IMPLIED〉
]〉
```

(3)基本数据科目字段属性定义

在实际开发过程中,统一版本数据采集系统中的不同表单却常常包含相同基本数据科目,如果对这些相同科目的相同属性多次重复输入,不仅影响工作效率,而且容易产生不完整、不一致,而数据字典最重要的目的之一就是存储一个基本科目的信息,以便进行重建和引用操作。因此在定义表单结构的基础之上,应当建立基本数据科目属性字典以定义应用项目的不同版本数据文件中的所有内部表单所使用的所有基本数据科目,其目的是为了确保相同的科目总是具有相同名称,并具有相同的属性——即使一个科目被多个内部表单使用,它也只被定义一次,这样确保了基本科目定义是标准化与规范化的。基本数据科目的基本结构定义:

```
〈!ELEMENT DATA_ITEM #PCDATA〉
〈!-- 定义字段 ID --〉
〈!ATLIST DATA_ITEM ID CDATA #REQUIRED〉
〈!-- 定义字段名称 --〉
〈!ATLIST DATA_ITEM NAME CDATA #REQUIRED〉
〈!-- 定义字段取值类型 --〉
〈! ATLIST DATA_ITEM VALUE_TYPE CDATA # REQUIRED〉
〈!-- 定义字段取值长度 --〉
〈!ATLIST DATA_ITEM VALUE_LENGTH CDATA #REQUIRED〉
〈!-- 定义字段取值单位 --〉
〈! ATLIST DATA_ITEM VALUE_UNIT CDATA # REQUIRED〉
〈!-- 定义取值是否允许为空 --〉
〈! ATLIST DATA_ITEM VALUE_NULL CDATA # REQUIRED〉
〈!-- 定义字段取值缺省值 --〉
〈!ATLIST DATA_ITEM DEFAULT_VALUE CDATA #REQUIRED〉
〈!-- 定义字段约束规则 --〉
〈!ATLIST DATA_ITEM RULE CDATA # REQUIRED 〉
〈!-- 定义字段在用户界面表现时使用的控件类型 --〉
〈! ATLIST DATA_ITEM CONTROL_TYPE CDATA #REQUIRED〉
〈!-- 定义字段取值的候选项 --〉
〈! ATLIST DATA_ITEM OPTION_ITEM CDATA # REQUIRED〉
]〉
```

针对数据自动入库功能的实现而定义的新型数据字典主要用于建立不同数据科目与数据库相应字段之间的对应关系和元数据模型的数据入库规则,以便为数据自动入库系统提供所需信息。为了建立不同数据条目与数据库相应字段之间的对应关系和元数据模型的数据入库规则,以便为数据自动入库系统提供所需信息,在改造后的数据字典中专门提供相关的定义,其基本结构如下:

```
〈!DOCTYPE DB_DICTIONARY〉
〈!ELEMENT DB_DICTIONARY (TABLE)+〉
〈!ELEMENT TABLE (FIELD)+〉
〈!ELEMENT FIELD (CAL_VALUE)+〉
〈!ATLIST TABLE NAME CDATA #REQUIRED〉
〈!ATLIST FIELD NAME CDATA #REQUIRED〉
```

其中 TABLE 代表数据库中的一个表,FIELD 代表数据库表中的一个字段,而两个元素所具备的 NAME 属性则声明了数据库中的表名和字段名。而 CAL_VALUE 元素的取值定义了数据源,对应于元数据模型中基本数据项标识符或通过基本数据项标识符构成的数学表达式。

与上面提出的用于实现元数据对象约束条件的数据字典不同,针对入库规则的数据字典将以数据库表的形式保存在特定数据库管理系统中。数据采集和管理系统只有在需要进行数据入库操作时才从数据库中加载该数据字典的定义,以实现自动入库操作,而在业务流程中的其他阶段,应用系统并不需要数据字典中的这部分定义。

3. 元数据对象控制模型的设计

根据数据字典中针对不同级别数据对象约束信息的定义,元数据对象控制模型的实现也划分为相应的级别。

多级控制模型的定义

(1)文件级和表单级控制模型

主要实现对元数据的值约束功能。采用 Hashtable(散列表)对象建立保存特定元数据项值约束规则的容器,将元数据项的标示符(表单 ID. 数据项 ID)作为键值,由于一个元数据项可能对应多条表单级交叉约束规则,因此将一个元数据项对应的所有约束规则表达式保存在一个向量对象(Vector)中作为 Hashtable 对象中的元素值。

(2)基本数据项级控制模型

将数据字典中对于一个内部表单包含的所有数据项约束规则的定义统一保存在一个控制类数据对象中。同时针对每个表单的控制类数据对象的共性实现一个抽象类 FormDataModel。抽象类的作用:所有表单数据控制类的父类,其中提供实现数据控制功能的模型和公共接口。各个表单所对应的数据类各自参照数据字典中的定义完成控制模型中各个成员变量的赋值。

a. 变量定义

FormDataModel 类中主要的变量包括:

- private ErrorList Errs;
- public boolean changed:表单数据经过修改的标志位;
- public boolean checked:表单通过完整校验的标志位;
- public int DataItemModel[][];
- public Hashtable DataItemInputRule;
- public Hashtable DataItemAccountRule;
- protected Object Data[];
- protected Object BufferData[];
- public int NumberOfData:当前表单中包含的数据项的总数;
- public int NumberOfComponent;

b. 变量和固定取值的说明

·DataItemModel 二维数组的构造:描述内部表单对象中所有表项控制信息的定义,取值将直接通过基于 XML 的数据字典文件解析得到。

(下转第68页)

锐化并缩小　　图像被扭曲　　图像被有损压缩

图3

总结　从以上实验结果可以看出，本算法对原始图像进行了分块求均值和方差，然后对水印图像按照计算所得的结果进行自适应的调整，从而使水印图像对于原图像的亮、暗灰度变化同样具有自动调整适应的能力，使得图像在嵌入水印后并不影响其观赏价值。另外，由于采用小波变换进行分频段水印嵌入，低频段嵌入水印低频部分，高频段嵌入水印高频部分，从而使得水印具有较强的鲁棒性，如果通过小波反变换试图去掉水印，则必然破坏原图像。从中还可以看出，算法对于常见的图像变换压缩等处理具有较好的抵抗性。

参考文献

1　Braudaway G W. Protecting Publicly-Available Images with a Visible Image Watermark IBM Research Division, T. J. Watson Research Center:[Technical Report 96A000248]　（下转第142页）

（上接第118页）

·DataItemModel[][0]：当前数据项在界面显示所使用的控件类型，例如0代表 TextField，1代表 TextArea、2代表 CheckBox 等，3代表 Table 等；

·DataItemModel[][1]：当前数据项在组件库中的位置索引；

·DataItemModel[][2]：当前数据项的数据类型，例如0代表 String，1代表 int 等；

·DataItemModel[][3]：当前数据项的取值是否允许为空；

·DataItemModel[][4]：当前数据项在界面中是否显示（处理那些数据位不连续的情况）；

·DataItemModel[]：第一维数组下标标识当前数据项的ID；

·DataItemInputRule：采用数据项 ID 作为键值，采用数据字典中针对特定数据项定义的输入规则字符串作为取值；

·DataItemAccountRule：采用数据项 ID 作为键值，Vector 类型对象作为取值。Vector 对象中的元素为数据字典中在数据项级上针对一个数据项定义值约束规则；

·Data[]：存放数据的具体取值，其下标和 DataItemModel[]对应；

·BufferData[]：是 Data[]的拷贝，所有数据在校验前都用它保存；

笔者曾经负责上海证券交易所上市公司定期报告申报和实时发布系统的设计和开发，对大型数据采集和管理系统所采用的技术、方法及实施过程都有切身体会，受篇幅所限，下面仅就如何利用数据字典定义的数据对象控制模型进行用户界面构造和数据库自动入库设计系统开发等几个基本模型开发进行探讨。

结论　经过对传统数据字典模型的改造，并将其应用于应用系统数据模型的定义上，新型的数据字典及其定义的元数据控制模型具有如下优点：

·针对数据申报系统而言，元数据项的名称经常需要改变。由于引入了数据字典，代表具体表格数据的类中只需要定义数据域的代码，而不必定义数据域对应的字符串表示信息，系统在需要这些表示信息时，可以直接从数据字典中获取。这样更改数据域表示名称的操作也变得非常简便，只需更改数据字典中的设置即可；

·通过整数类型的数据变量标识每个数据对象中的元数据，这样大大减少了数据对象所占用的内存空间；

·基于数据字典的元数据控制模型的引入使得开发自动入库设计工具成为可能。

·大大提高了操作的可重用性，这表现在两个方面：一方面整个系统在数据控制方面均通过一个数据字典配置即可完成，与程序代码无关；另一方面即使数据库表发生了结构变化，也只需要修改数据字典中有关数据项入库规则的配置而不需要修改程序本身；

·系统设计周密，逻辑性强，且编程接口统一，不因开发者不同而异；

·系统可扩展性强，维护容易。

本文中讨论的基于数据字典的元数据控制模型的设计，已经成功地应用于上海证券交易所上市公司定期报告在线采集和实时发布系统中，并在上交所管理的500余家上市公司中成功应用。

参考文献

1　车敦仁，周立柱，OLAP 及多维数据库技术. 见：第十三届全国数据库会议论文集. 1995
2　秦晓. 元数据字典及其实现. 计算机学报. 1994. 2
3　郭胜辉，孙玉芳. 基于数据字典库的信息系统的设计，计算机学报，2000. 4
4　徐鹏. 基于客户端 GUI 构件实例缓存技术的系统性能优化处理. 计算机工程与应用，2001. 1