科技评价系统中的元数据表示与管理

龙 军,陈舜子,刘高嵩
(中南大学 信息科学与工程学院,湖南 长沙 410075)

摘 要:当前,科技评价系统主要以申报书的形式接收评审项目的信息,各个系统申报书的异构性导致大量的异构科技评价信息分散在各个系统中,无法进行有效的管理和利用。为了提高科技评价系统的可维护性和可用性,文中以元数据为指导思想,通过对异构申报书进行对比分析,总结出它们的共同特征,定义了科技评价元数据实体来对异构信息进行统一描述;比较各种元数据表示方式的优劣,并结合科技评价系统的实际需求选择关系模型作为元数据表示方式;最终提出一套完整的集中化元数据管理策略并以之建立了科技评价元数据管理系统对元数据描述信息进行统一管理,从根本上解决了科技评价系统中的信息异构问题。

关键词:元数据;管理;异构;科技评价
中图分类号:TP18 **文献标识码**:A **文章编号**:1673-629X(2010)10-0012-04

Metadata Representation and Management in Scientific Assessment System

LONG Jun, CHEN Shun-zi, LIU Gao-song
(Institute of Information Science and Engineering, Central South University, Changsha 410075, China)

Abstract: Nowadays scientific assessment systems mainly use the application sheet to collect information of the projects to be assessed. The heterogeneity of application sheets used in different systems causes heterogeneous data to be spread in these assessment systems, making it extremely difficult for data maintenance and utility. To raise the maintainability and usability of the scientific systems, follow the leads of metadata, analyze heterogeneous application sheets to find their common ground, define scientific assessment metadata entity to represent heterogeneous scientific assessment data uniformly; compare different measures of metadata representation to be aware of their pros and cons, use relational model to represent scientific assessment metadata after considering the actual requirements of the system; provide a central metadata management strategy based on which build a scientific assessment system, to solve the heterogeneous problems in the scientific assessment systems.

Key words: metadata; management; heterogeneous; scientific assessment

0 引 言

近年来,随着互联网的蓬勃发展,越来越多的国家基金开始采用网络申报的方式来收集科技评价所需要的项目信息。网络申报提高了工作效率,但是也带来了新的问题。从用户的角度来说,由于各个基金的申报书格式不同导致各个科技评价系统的数据异构性,难以对申报信息进行统一的查询和统计分析;从开发者的角度来说,描述申报书格式的信息分散在各个系统中,无法进行有效的统一维护和管理。

文中通过分析异构申报书的特点来定义科技评价元数据实体,根据科技评价系统的实际需求选择合理的元数据表示方式,对异构申报书进行统一描述来提供异构科技评价信息的统一操作接口;同时提出了一套完整的元数据管理策略来对元数据进行有效管理,以解决上述科技评价系统中的异构问题。

1 科技评价的元数据分析

元数据是关于数据的信息,是具有描述、解释、定位信息资源功能的结构化信息,是说明数据内容、质量、状况及其他有关特征的描述信息[1,2]。它在科技评价系统中主要用于描述申报书格式,在指明申报书格式具体组成的同时,又规定了这些组成项如何被存储、获取和管理。有了元数据可以使得申报书格式的

收稿日期:2010-02-04;**修回日期**:2010-05-23
基金项目:国家自然科学基金项目(60873081,U0835003,60970095,M0921005)
作者简介:龙 军(1972-),男,副教授,博士,研究方向为网络资源管理与服务;刘高嵩,副教授,研究方向为软件工程、计算机网络技术。

维护更加容易,极大地方便了异构科技评价信息的管理。

1.1　科技评价信息的异构性

要制定出科学的元数据标准,必须先对申报书的异构性进行分析。申报书的异构性主要表现在以下几个方面:

(1)申报书的结构不同。通常申报书由一些大的组成项构成,比如申请人的个人信息、项目基本信息、项目的工作计划等;而这些项又由更小的项组成,比如申请人的个人信息,包括姓名、性别、年龄等。组成项不同是申报书异构的根源。

(2)组成项约束不同。组成项的约束主要表现为施加在申报信息上的验证规则,比如有的基金要求主要申报人必须具有博士学位,有的对申请人的职位有限制等等。

(3)申报书的版本不同。同一个基金的申报书,在不同的年份可能有所差别,需要对不同的版本进行追踪管理。

1.2　科技评价的元数据实体

根据上述分析,科技评价元数据至少应描述三个方面的内容,即申报书组成项、约束以及版本信息。同时注意到,申报书格式事实上具有递归的结构特征,可以把一份申报书格式看做一个组成项,即它本身;这一组成项又可划分为几个部分,可视为较大的子组成项,然后可以进一步细分直至不可分割的原子组成项。故此,为科技评价元数据定义以下实体:

(1)组成项。它的属性包括基本信息,如数据类型、标题、提示、默认值;版本信息,如创建日期、修改日期、版本号、作者;存储信息,如对应的表名和字段名;同时与一个或多个验证规则相关联。

(2)验证规则。验证规则的存在是为了确保申报信息的合法性,它包括验证类型和验证参数两个属性。验证类型与可调用的验证函数对应,传入验证参数即可执行相应的验证功能。验证规则可以与组成项或数据类型相关联。

(3)数据类型。数据类型本质上是对验证规则的一种默认集成,同时也是对元数据语义的强化[3]。比如为了能给出准确的验证反馈消息,可能为身份证组成项设置多种验证规则,如必须是数字与字母的组合,必须符合特定的格式等。设置身份证数据类型,默认集成所需的验证规则,那么在创建组成项时只需指定数据类型,避免每次都重复地添加每条验证规则。

(4)组成项类别。组成项类别是对组成项的一种逻辑归类,以方便用户能够迅速地从元数据库中找到所需要的组成项[3],组成项类别与组成项之间是多对多的映射关系。

2　科技评价元数据的表示

2.1　元数据表示方式的选取原则

选择元数据的表示方式,首先要在开放性和可用性之间取得平衡[4]。开放性指的是一个系统的元数据能否与外部系统兼容,即在外部系统使用本系统元数据难易程度。可用性指的是在本系统中使用元数据的效率。

通常来说,使用 XML[5]表示方式的元数据具有较高的开放性[4]。XML 是通用的结构化表示语言,能够用来描述各种领域的结构化数据;同时它本身也是一种开放性的标准,可以与现存的几乎所有编程语言兼容。但是 XML 在可用性方面不太理想,因为使用 XML 会引入大量标签,将实际数据嵌入到标签当中。因此在程序中不可避免地要在 XML 文件和实际数据之间进行转化,这为系统引入了额外的负担。另一方面,使用编程语言直接表示的元数据,具有最好的可用性[3]。但是若要在使用不同语言的系统间进行交互,则非常困难,具有较差的开放性。

除了上述原则外,元数据的表示方式必须反映系统自身的需求[3]。在科技评价系统中,元数据所描述的申报书格式是需要被修改和创建的实体,直接用编程语言来表示将导致程序频繁地修改和重编译。另外,由于科技评价元数据主要应用于已有的各个科技评价系统,对各个系统的元数据进行集中式的管理,与外部系统进行交互的需求相对较少。

2.2　科技评价元数据的表示方式

综合以上考虑,决定采用数据库的关系模型做为科技评价元数据的表示方式。关系型数据库是目前的主流数据存储方式,用关系模型表示的元数据结构,能够适应各种关系型数据库环境[6]。当前的主流编程语言对关系型数据库的操作都提供了良好的支持,比较常见的方式是提供 ORM(Object Relational Mapping)类库,使得程序员能够直接对领域模型中的对象进行持久化操作。因此,用关系模型作为表示方式具有良好的可用性。下面给出了组成项实体的关系模型和用 Python 语言的 SQLAlchemy ORM 类库描述的映射模型。

```
// 组成项的关系模型
template_schema = (
id, version, name, author, create_time, last_edit_time, parent,
title, tooltip, template_type_id, default_value, table_name, col-
umn_name,
)
```

```
// 用 Python 语言的 SQLAlchemy ORM 类库描述的映射模型
template_table = schema.Table(
'template', metadata,
schema.Column('id', types.Unicode, primary_key=True),
schema.Column('version', types.Numeric, default=1.0),
schema.Column('name', types.Unicode, nullable=False),
schema.Column('author', types.Unicode, nullable=False),
schema.Column('create_time', types.DateTime, default=now),
schema.Column('edit_time', types.DateTime, default=now,
onupdate=now),
schema.Column('parent', types.Unicode, schema.ForeignKey
('template.id')),
schema.Column('title', types.Unicode, nullable=False),
schema.Column('tooltip', types.Unicode),
schema.Column('template_type_id', schema.ForeignKey('tem-
plate_type_id')),
schema.Column('default', types.Unicode),
schema.Column('isSection', types.Boolean, default=False)
)
```

上述映射关系可以使得我们方便地对组成项这一元数据实体进行持久化操作，比如创建一个新的对象并进行保存，只需执行如下语句：

```
template = Template() //创建新对象
template.name = "real_name"
template.title = "姓名"
template.default = "null"
..... //完成赋值操作
session = Session()
session.add(template) //将新对象保存到数据库
```

当需要与外部系统进行交互时，先将元数据从数据库中导出并转化为 XML 的格式再进行传输，以保证系统的开放性。比如一个组成项的信息可以表示为如下格式：

```
<metadata>
  <type>template</type>
  <name>real_name</name>
  <version>1.0</version>
  <title>姓名</title>
  <default>null</default>
  .....
</metadata>
```

3 科技评价元数据的管理

3.1 科技评价元数据的管理模式

对元数据采用集中式管理[7,8]，科技评价元数据管理系统运行在独立的元数据服务器上，负责提供对元数据的添加、修改、查询、删除等操作。

元数据的添加是指向系统中加入新的元数据实体，利用基本数据类型和验证规则和定义组成项，来对申报书格式进行描述。元数据管理系统的管理员对这些组成项进行筛选后分类归入元数据库。在后续的添加过程中，若遇到类似的组成项，则可直接从类库中复制已有的组成项并在其基础上进行修改，如图 1 所示。建立元数据库的好处是，系统中已注册的申报书格式越多，则从类库中找到所需组成项的可能性越高，避免重复定义类似的组成项，彻底改变了未使用元数据时申报书越多越难以管理的状况。

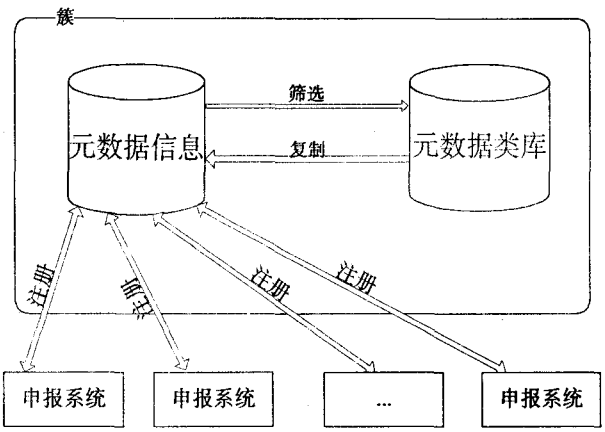

图 1 科技评价元数据的创建

对于元数据的修改引入版本控制的概念。一方面可以对元数据的各个版本进行有效管理，另一方面可以进行版本恢复操作，提高容错性[9]。对于一般元数据信息，组成项将继承其所属组成项的版本信息。任何对申报书子组成项的修改都将被视为是对申报书格式的修改，进而导致申报书格式版本的变化。

实现版本控制最简单的方法是为申报书格式的每个版本都保留完整的副本。但是当申报书较为复杂时，这种方式将变得非常低效并且浪费存储空间[10]。为了避免这一问题，本系统采用渐进式的版本管理模式，即对于当前版本 C，记录以下信息：版本 C 的前一个版本 P，版本 A 的下一个版本 N，版本 C 新增的组成项，版本 P 中被删除的组成项，版本 P 中被修改的组成项。若需要将当前版本的元数据恢复为之前的某个版本 A，则可根据版本 V 以及 V 与 A 之间的过渡版本计算出 V 与 A 之间的差异，并以之进行恢复操作。具体算法如下所示：

```
ALGORITHM RECOVER(m, a, current)
//Recover the current verision of metadata to a previous version
//Input: metadata m, current version, version a
//Output: metadata m in version a
v <- current
while v.previous is not a do
v <- UNION_VERSION(v, v.previous)
new_current <- a new Version instance
```

```
new_current.addition <- v.obsolete
new_current.obsolete <- v.addition
new_current.old <- the content going to be replaced by v.old in m
new_current.previous <- current
delete v.addition from m
add v.obsolete to m
replace the corresponding content with v.old in m
return m
ALGORITHM UNION_VERSION(a, b)
//Implements the union of two version
//Input: version a, version b, a = b.next, b = a.previous
//Output: the union of the two version
v <- a new Version instance
for metadata m in b.addition do
if m not in a.obsolete
v.addition += m
v.addition += a.addition
for metadata m in a.obsolete do
if m not in b.addition
v.obsolete += m
v.obsolete += b.obsolete
for metadata m in a.old do
if m not in b.old and m not in b.addition
v.old += m
v.old += b.old
v.previous = b.previous
return v
```

元数据的查询操作通常根据元数据的类别来进行。一个组成项可以被归为多种类别,以提高被查到的几率。元数据的删除操作将导致该元数据所有版本的信息一齐被删除。

3.2　基于元数据的申报信息统一访问接口

为异构的科技评价信息提供统一的访问操作接口是制定科技评价元数据主要目的之一。在科技评价元数据管理系统的基础上,异构的申报信息格式已经统一为元数据描述。从这个角度看,元数据指明了标准化的申报信息格式与异构的申报信息格式之间的映射关系[11,12]。首先从元数据类库中选取主要的组成项作为查询特征,提供统一的查询接口。当用户向不同的申报系统提交查询条件时,先从元数据库中提取相应系统申报书格式的的元数据描述,将标准查询条件转化为该系统可以执行的查询语句。然后将该语句提交到相应的申报系统中执行并获得返回结果。最后将查询结果转化为标准格式返回给用户,如图 2 所示。

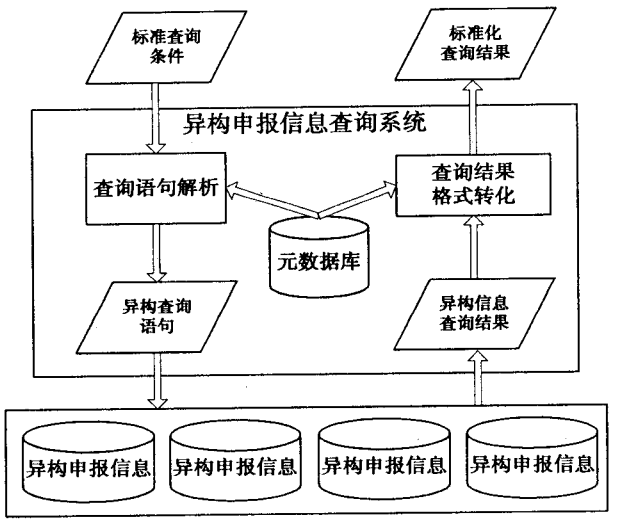

图 2　异构申报信息统一查询系统的流程

4　结束语

针对目前科技评价系统中普遍存在的信息异构问题,文中以元数据为指导思想,定义了科技评价信息的元数据标准,提出了以关系模型为基础的元数据表示方式和集中化的元数据管理策略。本方案具有以下几个特点:

(1)以关系型数据库作为元数据的存储方式,在元数据的可用性和开放性之间取得较好的平衡;

(2)建立组成项类库,开发新的申报书时可利用已有的组成项,提高了工作效率;

(3)引入版本控制的概念,大大提高了元数据管理系统的有序性和可维护性。

参考文献:

[1] NISO Press. Understanding Metadata[EB/OL]. 2004. http://www.niso.org/standards/UnderstandingMetadata.pdf.

[2] 应　英.电子政务元数据管理系统的研究与设计[D].西安:西北工业大学,2007.

[3] 高　妮,周明全,耿国华,等. 网络科技资源平台中元数据的设计与实现[J]. 计算机工程与应用,2009,45(25):141-144.

[4] Widener P, Schwan K, Eisenhauer G. Open Metadata Formats: Efficient XML-Based Communication for Heterogeneous Distributed Systems[C]//21st IEEE International Conference on Distributed Computing System. Mesa: [s. n.], 2001: 739-742.

[5] The extensible markup language (XML) [EB/OL]. 1998. http://www.w3.org/TR/1998/REC-xml-19980210.

[6] 张智海,张晓清,潘　清,等. 分布式海量数据管理系统 Hypertable 元数据表分析[J]. 计算机与现代化,2009,(8):113-115.

(下转第 20 页)

表 6 对 UCI 中五个数据库使用不同属性约简算法处理的比较

数据集	样本数_ n	属性数_ n	约简集属性个数_ n		
			遗传算法	动态约简算法	新算法
Monk1	124	6	3 或 4	3 或 4	3
Heart Disease	294	14	3 或 4 或 5	3 或 4 或 5 或 6	3
Mushroom	8124	22	5 或 6 或 7	5 或 6 或 7 或 8	5
Breast－Cancer	699	10	5 或 6 或 7	5 或 6 或 7	5
Slope Collapse	3436	24	9 或 10 或 11	9 或 10 或 11 或 12	9

通过实验和比较可知，使用标准遗传算法和动态约简算法都得到了好几个属性约简集，无法得知哪个属性约简集是最好的。而我们提出的约简算法仅得到具有最少属性个数的属性约简集，有效且提高了约简速度。

3 结束语

启发式属性约简算法由于首先是从属性核开始添加选择的属性，这样就避免了对属性之间随机组合情况的搜索，又由于每一步骤的属性选择时，都通过不一致计数和互信息增量的计算对待选属性的重要性进行了排序，就避免了对超出最小属性约简个数的属性组合情况的搜索，因此可以提高求解速度，并可以得到最小属性约简结果。此方法适合于有多个条件属性的情况，条件属性与属性核的选择需要根据具体情况来确定。

参考文献：

[1] Pawlak Z. Rough sets[J]. International Journal of Computer and Information Science, 1982, 11(5): 341－356.

[2] Pawlak Z. Rough set－Theoretical Aspects of Reasoning about Data[M]. Dordrecht: Kluwer Academic Publishers, 1991.

[3] Pawlak Z, Slowinski R. Rough set approach to multiattribute decision analysis[J]. Ivited Review, European Journal of Operational Research, 1994, 72: 443－459.

[4] Tseng T L, Huang C C. Rough set－based approach to feature selection in customer relationship management[J]. Omega, 2007, 35: 365－383.

[5] 梁吉业，曲开社，徐宗本. 信息系统的属性约简[J]. 系统工程理论与实践，2001(12): 76－80.

[6] 芦晓红，陈世权，吴今培. 基于可辨识矩阵的启发式属性约简方法及其应用[J]. 计算机工程，2003, 29(1): 56－58.

[7] 徐章艳，杨炳儒. 一个基于决策表的快速属性约简算法[J]. 小型微型计算机系统，2006, 27(5): 858－861.

[8] Wong S K M, Ziarko W. On optimal decision rules in decision tables[J]. Bulletin of Polish Academy of Sciences, 1985, 33: 693－696.

[9] Jelonek J, Krawiec K, Slowinski R. Rough set reduction of attributes and their domains for neural networks[J]. Computational Intelligence, 1995, 11(2): 339－347.

[10] 王国胤. Rough 集理论与知识获取[M]. 西安：西安交通大学出版社，2001.

[11] 王国胤，于 洪，杨大春. 基于条件信息熵的决策表约简[J]. 计算机学报，2002, 25(7): 759－766.

[12] Ning Z, Juzhen D, Setsuo O. Using Rough sets with Heuristics for Feature selection[J]. J. Intell. Inf. Syst, 2001, 16: 199－214.

(上接第 15 页)

[7] 刘 峰，顾君忠. 元数据管理应用系统的设计与实现[J]. 计算机工程，2009, 35(11): 29－31.

[8] 沈盛彧，刘 恺，姚文龙. 基于 Web 服务的空间元数据管理平台研究[J]. 测绘与空间地理信息，2009, 32(4): 13－15.

[9] 谢福成，王备战，史 亮，等. 基于银行数据仓库的元数据管理系统[J]. 计算机工程，2009, 35(9): 79－81.

[10] 蔡昭权，卢庆武. 郑宗晖，等. 基于元数据的快速开发平台设计与实现[J]. 计算机工程，2009, 35(9): 60－62.

[11] 徐财江，陈和平，陈志荣. 土地利用现状数据元数据管理系统的设计与实现[C]//2006 年中国土地学会学术年会论文集. 重庆：[出版者不详]，2006: 707－713.

[12] 何清林，杨 森，徐泽同. 基于元数据和 Web Service 中间件的分布式资源库集成[J]. 计算机工程与设计，2009, 30(9): 2201－2203.