基于规则的关系数据库到本体的转换方法 *

余 霞[1,2], 刘 强[1,2], 叶 丹[1]
(1. 中国科学院 软件研究所 软件工程技术中心, 北京 100080; 2. 中国科学院 研究生院 信息科学与工程学院, 北京 100049)

摘 要: 提出了一种新的全自动的关系数据库到本体的转换方法,通过分析关系模式的主键、属性、引用关系、完整性约束和部分数据来创建本体,尽量保持了关系数据库的信息,并在构建的过程中对信息进行初步的集成和分类。系统实践证明,该方法可自动进行关系模式和数据到本体的等价转换,而且完成了对关系数据库中部分语义信息的辅助挖掘。

关键词: 数据集成; 关系模式; 本体; 框架逻辑; 资源描述框架
中图分类号: TP311 **文献标志码**: A **文章编号**: 1001-3695(2008)03-0767-04

Rule-based approach to transform relational database to ontology

YU Xia[1,2], LIU Qiang[1,2], YE Dan[1]
(1. *Technology Center of Software Engineering, Institute of Software, Chinese Academy of Sciences, Beijing* 100080, *China*; 2. *College of Information Sciences & Engineering, Graduate School, Chinese Academy of Sciences, Beijing* 100049, *China*)

Abstract: The paper proposed a novel automatic transform method from relational database to ontology. By the analysis of primary keys, attributes, foreign keys, integrity constraints of relational model and partial data, this method could construct ontology while conserving the information of the relational databases and fulfilling primary integration and classification. The implementation of the system shows that this method can automatically make the equivalent transformation from relational schema and data to ontology, and accomplish aiding mining about partial semantic information in the relational databases.

Key words: data integration; relational model; ontology; frame logic; RDF

0 引言

计算机网络的迅速发展推动了信息化和全球化的进程。企业与企业之间、企业各部门之间的信息交换越来越频繁。由于地理位置的分布性和所采用技术的多样性,直接导致了数据源的异构性,数据模式和数据表示的差异给数据集成造成了很大困难。传统的数据集成基于关系模式,只考虑了数据的语法信息,这在很大程度上影响了数据集成的准确性。

知识表示中的重要支撑工具是本体(ontology)[1]。本体是概念化的、明确的规范说明[2]。其中,概念化是指抽象出客观世界中一些相关概念而得到模型,其表示的含义独立于具体的环境状态;明确是指所使用的概念及使用这些概念的约束都有明确的定义;规范是指使用标准的、独立于系统的形式化描述。作为一种概念的显式表达方式,本体能够以一个概念集的形式表示任何信息及它们之间的关系。

基于本体的数据集成主要借助于本体来描述数据源信息,通过定义共享词汇集来揭示数据源模式的语义及其他的语义信息。与基于关系模式的数据集成相比较,它可以进一步丰富数据模式的语义表达能力,有效处理各种语义冲突。如何将关系模式映射到本体并尽量保持数据库的语义信息是基于本体数据集成的重要组成部分。

1 相关工作

在深度数据集成和逆向工程[3]的研究等方面,对于显式定义和抽取数据库模型的语义信息已经作了不少研究,但是只有少部分方法把本体作为目标。其中,Irina Astrova 提出的方法[4]与本文提出的方法最为相似。它提出了一套模式映射规则,对关系进行分类,通过分析主键、属性和数据发掘关系之间的联系,将关系模式映射到本体,然后将关系数据映射到本体实例。但是如果是海量数据时,该方法在分析所有属性上的数据之间的全等、包含、交叉和分离是非常困难的。相对于 Irina Astrova 提出的方法,Stojanovic 等人提出的方法[5]考虑了主/外键上数据全等和包含两种情况,但没有数据交叉和分离的情况;此外该方法是半自动化的,在确定概念继承层次时需要更多的用户交互。Kashyap 给出的方案[6]需要大量的用户交互,进行语义注释,降低了系统自动化程度;此外该方案也没有建立公理系统,公理系统可以更深层次地描述本体实例间的关系。Dogan 和 Islamaj 提出了一种全自动的模式和数据的转换方法[7],把关系映射成概念、属性映射成对应概念的谓词、关系的元组映射成本体实例,但是没有考虑信息集成和继承层

收稿日期: 2007-01-19; **修回日期**: 2007-04-09 **基金项目**: 国家"863"计划资助项目(2004AA112010); 国家自然科学基金资助项目(60573126)

作者简介: 余霞(1981-),女,湖北天门人,硕士研究生,主要研究方向为数据集成、网络分布式计算、软件工程技术(yuxia04@otcaix.iscas.ac.cn); 刘强(1979-),男,博士研究生,主要研究方向为数据集成、网络分布式计算、软件工程技术; 叶丹(1971-),女,高级工程师,CCF 高级会员,博士,主要研究方向为应用集成技术、网络分布式计算、软件工程技术.

次,因此创建的本体更像一个关系型的本体。在国内,任保锋等人在关系数据库到本体的映射方面也作了一定的研究[8]。

本文提出了一种新的全自动的关系数据库到本体的转换方法。通过分析关系模式的主键、属性、引用关系、完整性约束和引用关联的关系中主键上数据的全等和包含关系来创建本体,给出了一组框架逻辑[9]描述的关系模式到本体概念和层次的映射规则,基于这种转换,将关系数据迁移到本体示例。相对于已有的转换方法,该方法在最大限度地保持关系模式的完整性的同时,转换和迁移过程无须人工参与,基于规则保证了转换过程的自动化。

2 基本数据模型

关系数据库的底层模型是关系模式,它是当前数据库领域最重要的数据模式,有着广泛的应用。本文扩展了通常的关系模式的形式定义。一个关系模式包括:

a)关系的有限集合 R;

b)属性的有限集合 A;

c)基本数据类型集合 T;

d)获取关系属性的函数 attr:$R\rightarrow 2^A$,给出特定关系的属性集合;

e)获取关系主键的函数 key:$R\rightarrow 2^A$,说明哪些属性是关系的主键。因此对于 $r\in R$,key(r)$\subseteq$attr(r);

f)获取属性数据类型的函数 type:$A\rightarrow T$ 给出每个属性的数据类型;

g)引入函数 data:$2^A\rightarrow V$,V 为值的集合,函数给出属性组的取值集合。

另外关系还具有完整性约束特征,这包括:

a)实体完整性。例如主属性不能取空值、default、not null、unique等。

b)参照完整性。约束关系间的引用关系。

c)用户定义完整性。例如某个属性的取值在 0 ~ 5 等。

由于本体的静态属性,本文只对上述关系模式进行映射,而 SQL-DDL 的动态特征(如触发器、断言等)将不予考虑。

下面给出一个关系模式的例子。关系 dept 有两个属性,即 deptID 和 deptName,基本数据类型分别是 integer 和 varchar,其中 deptID 是主键;关系 student 有两个属性,即 stuID 和deptID,基本数据类型都是 integer,其中 stuID 是主键,deptID 是参照关系 DEPT 的外键。

```
create table dept(deptID interger primary key,deptName varchar)
create table student(stuID interger primary key,deptID interger references dept)
```

一个本体包括:概念的集合 C;谓词的集合 P;概念的继承层次 H:$H\subseteq C\times C$,$H(C1,C2)$ 说明 $C1$ 是 $C2$ 的子概念;谓词到概念的映射 domain:$P\rightarrow C$,给出谓词的主语来自的域;谓词到概念的映射 range:$P\rightarrow C$,给出谓词的宾语来自的域;公理系统 A。

近年来,研究界提出了一些本体描述语言,如 RDF(资源描述框架) + RDF schema、OWL、DAML + OIL 等。本文使用框架逻辑语言来描述本体形式语义。一个用框架逻辑描述的本体可以转换成相应的 RDF 描述[10]。无论是框架逻辑还是 RDF schema 都没有基本数据类型,一切皆是对象,属性与关联也不作区分,对象与对象之间通过谓词联系。下面给出了对应上述两个关系用框架逻辑描述的本体片断。其中,dept、student 是概念 object 的子概念。Dept 有两个谓词 deptID 和 deptName,student 有两个谓词,即 stuID 和 deptID。Student 和 dept 之间通过 deptID 联系;student 的实例在谓词 deptID 上的 range 值对应概念 dept 的一个实例。

```
dept::object.
student::object.
dept[deptID =>> integer,deptName =>> string].
student[stuID =>> integer,deptID =>> dept].
forall X,Y Y:dept<-X:student[deptID->Y].
```

假设有一个通过关系模式定义的关系数据库,需要将其转换到本体,转换后的本体与关系模式的区别在于:a)关系模型包含基本数据类型,但是在本体中一切都是概念。这里为每个基本类型创建一个对应的概念;b)关系数据库中的属性和关联在本体中并不作区分,统一用谓词表示;c)本体中的概念和谓词可以组织成层次结构,表现更深层的语义;d)本体能提供信息的形式语义,建立公理系统,进而进行机器分析和推理。

3 转换

3.1 转换过程

关系数据库是数据迁移的源,本体是数据迁移的目标。关系模式不能为关系数据库中的信息提供形式语义,该方法通过提供本体,尽量丰富关系数据库的语义信息。由于在实际应用中,关系模式以 3NF 最为常见,该方法以满足 3NF 的数据库为输入,数据迁移的过程如图 1 所示,共有四步:

a)抽取关系数据库的模式信息,如关系名、属性名、主键、外键、完整性约束等;

b)分析主键、外键、属性等信息,应用下文中给出的映射规则构造本体、创建本体概念,将概念进行组织(分类、分层等),并去掉冗余关系;

c)抽取关系数据库中的记录;

d)将关系数据库中的记录映射为本体实例,形成知识库。

其中:a)、b)是关系模式转换;c)、d)是关系数据迁移。

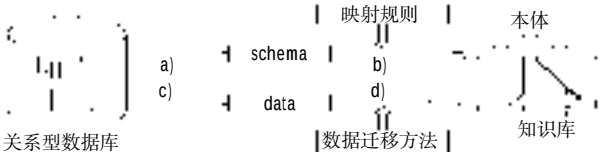

图1 数据转换过程

3.2 模式转换

模式信息的抽取可通过标准 ODBC 或 JDBC 等的 API 来获得,模式转换则通过应用一系列转换规则,对关系、属性、主外键之间的关联进行映射来完成。转换规则的应用有一定的先后顺序,进而本体也是逐步创建而成。本文引入以下辅助函数:a)ttoc:$T\rightarrow C$,表示将关系模式中的基本数据类型转换成对应概念;b)concept:$R\rightarrow C$,表示为关系模型中的关系创建对应概念;c)refer:$R\times R\times A\rightarrow$boolean,refer($r_i$, r_j, A) = true 说明关系 r_j 引用了关系 r_i 中的属性 A;refer(r_i, r_j, A) = false 说明 r_j 没有引用 r_i 中的属性 A。下面描述了一个关系模型,表 1 给出了运用一系列转换规则之后创建完成的本体。从关系映射、属

性及参照完整性映射和实体完整性映射三个方面来说明模式转换过程:

关系模式 R:

department(deptID, fullname, homepage)

student(stuID, name, sex, deptID references department)

phd(stuID references student, year, major)

excellent(stuID references student, level)

course(courseID, subject)

choose(courseID references course, stuID references student)

teacher(teacherID, stuffroom)

teacher_Info(teacherID references teacher, phone, address)

teach (teacherID references course, courseID references course, deptID references department)

表 1 为由关系模式 R 创建的本体。

表 1　由关系模式创建的本体

概念	谓词	约束和公理
department: : object. student: : object. phd: : student. excellent: : student. course: : object. teacher: : object. teach: : object.	department[deptID = > > integer, fullname = > > string, homepage = > > string, stuID = > > student]. student[stuID = > > integer, name = > > string, sex = > > string, deptID- > > department, courseID = > > course]. phd: : student[year = > > string major = > > string]. excellent: : student[level = > > string]. course[courseID = > > integer, subject = > > string, stuID = > > student]. teacher[teacherID = > > integer, phone = > > string, address = > > string]. teach[teacherID- > > teacher courseID- > > course deptID- > > department].	key(department, deptID). key(student, stuID). key(phd, stuID). key(excellent, stuID). key(course, courseID). key(teacher, teacherID). key(teach, teacherID, courseID, deptID). ref(department, student, deptID). ref(Student, course, stuID). ref(course, student, courseID). ref(teacher, teach, teacherID). ref(course, teach, courseID). ref(department, teach, deptID). forall $C1, C2, A$ ref($C1, C2, A$) < - forall IA exists IC2 IC2: $C2$ and (forall IA IA: $C1$ < - $C2$[A- > > IA]). forall C, A key(C, A) < - unique(C, A) and total(C, A). forall C, A unique(C, A) < - identicalValues(C, A) and singleValue(C, A). …

3.2.1　关系映射规则

关系映射是模式转换的第一步,属性映射和完整性映射都是在关系映射的基础上完成的。所有关系映射规则的形式描述如表 2 所示。

表 2　关系映射规则

规则	前提	结果
$R1$	$r \in R$, $A \in$ key(r) - $\exists r_i$(R, key(r_i) $\subseteq$ key(r) refer(r_i, r, A) = true	c: : object c = concept(r)
$R2$	$r \in R \exists r_i \in R$, $\exists c_i$ = concept(r_i) key(r) = key(r_i) $A \in$ key(r), refer(r_i, r, A) = true data(key(r)) = data(key(r_i))	无
$R3$	$r \in R \exists r_i \in R$, $\exists c_i$ = concept(r_i) key(r) = key(r_i) $A \in$ key(r), refer(r_i, r, A) = true data(key(r)) $\subset$ data(key(r_i))	c: : object c = concept(r) c: : c_i
$R4$	$r \in R$ $\exists r_i \in R$, key(r) $\supset$ (key(r_i) $\exists A$(key(r), refer(r_i, r, A) = true $\exists c_i c_i$ = concept(r_i)	c: : object c = concept(r) c: : c_i
$R5$	$r \in R$, key(r) = attr(r) $\exists r_i$, $r_j \in R$ key(r) = key(r_i) $\cup$ key(r_j) key(r_i) $\cap$ key(r_j) = $\varnothing$ $\exists c_i c_i$ = concept(r_i) $\exists c_j c_j$ = concept(r_j)	无
$R6$	$r \in R$ 规则 $R1$ ~ 5 均不适用	c: : object c = concept(r)

规则 $R1$ 首先为关系模式中的基本关系——主键不引用任何其他关系的主键的关系——创建概念。关于学校的关系模式 R 中,首先为关系 department、student、course、teacher 创建概念;规则 $R2$ 集成分散在多个关系中但表示同一事物的信息,它们共享同一概念。如关系 teacher 和 teacher_Info,不必为 teacher_Info 创建概念;规则 $R3$ 和 $R4$ 识别关系模式中的层次关系,创建子概念并创建概念层次。如规则 $R3$ 识别关系 excellent 和 student 之间的层次关系,规则 $R4$ 识别关系 phd 和 student 之间的层次关系;规则 $R5$ 处理只包含两个外健属性的关系。这种关系本身只表示另外两个关系之间多对多的关系,不必为之创建概念,如不为关系 choose 创建概念;规则 $R6$ 是缺省映射规则,当其他规则均不适用时,应用此规则,如对关系 teach 的处理。

3.2.2　属性及参照完整性映射规则

属性映射之前,本体概念已创建完成。属性的映射包括属性到本体谓词的映射及属性的参照完整性约束映射。先介绍属性到谓词的映射及参照完整性映射。

这里引入一个约束 ref,其框架逻辑定义如下:

forall $C1$, $C2$, A ref($C1$, $C2$, A)〈 - forall IA exists IC2 IC2: $C2$ and (forall IA IA: $C1$ < - $C2$[A-〉〉IA]).

其含义为概念 $C2$ 谓词 A 的 range 值在概念 $C1$ 的实例中。

一般地,一个属性映射为一个概念的一个谓词,但是由外键和主/外键体现的参照完整性表示的是模型中关系与关系之间的引用。为表示这种引用关系,可能不需要创建谓词,也可能需要为多个概念创建相应谓词。映射规则的形式描述如表 3 所示。

表 3　属性及参照完整性映射规则

规则	前提	结果
$A1$	$r \in R$, $A \in$ attr(r) - $\exists r_i$, refer(r_i, r, A) = true $\exists c$ c = concept(r)	c[A = > > ttoc(type(A))].
$A2$	$r \in R$, A(attr(r) - $\exists r_i$, refer(r_i, r, A) = true - $\exists c$ c = concept(r) $\exists c_j c_j$ = concept(r_j), $\exists A_j \in$ attr(r), $A_j \cap A = \varnothing$, refer(r_j, r, A_j) = true	C_j[A = > > ttoc(type(A))]
$A3$	$r \in R$, attr(r) = key(r) $A1$, $A2 \in$ attr(r) $A1 \cup A2$ = attr(r), $A1 \cap A2 = \varnothing$ - $\exists c$ c = concept(r) $\exists c_i c_i$ = concept(r_i), $A1$ = key(r_i) $\exists c_j c_j$ = concept(r_j), $A2$ = key(r_j)	c_i[$A2$ = > > c_j] c_j[$A1$ = > > c_i] ref(c_i, c_j, $A2$) ref(c_j, c_i, $A1$)
$A4$	$r \in R$, $A \in$ attr(r) $\exists r_i$, refer(r_i, r, A) = true $\exists c$ c = concept(r) $\exists c_i c_i$ = concept(r_i), c: : c_i	无
$A5$	$r \in R$, $A \in$ attr(r) $\exists r_i$, refer(r_i, r, A) = true $\exists c_i c_i$ = concept(r_i) $\exists c$ c = concept(r) - $\exists c_j c_j$ = concept(r_j), c: : c_j	c[A_i- > > c_i] c_i[A- > > c] ref(c_i, c, A)

规则 $A1$ 和规则 $A2$ 处理没有引用关系的属性。如果它所属关系创建了对应概念,应用规则 $A1$,则为该概念创建一个谓词;如果它所属的关系对应概念不存在,应用规则 $A2$,则这个关系一定集成到了其他关系中,为它们的共用概念创建对应的谓词。如对关系 department 的三个属性应用规则 $A1$ 映射;对关系 teacher_Info 的属性 phone 和 address 应用规则 $A2$ 映射;规则 $A3$ 处理关系之间的 n:m 关系的属性映射。为所引用的关

系对应的概念创建相应谓词,同时创建约束。如对关系 choose 的属性的映射;规则 *A*4 和 *A*5 处理存在引用关系且它所属关系对应概念存在的属性。如果存在父概念,应用规则 *A*4 则不创建相应谓词;如果父概念不存在,应用规则 *A*5,则为该概念及引用指向的概念创建谓词,同时创建约束。如对关系 phd 的属性 stuID 应用规则 *A*4 映射,对关系 teach 的属性应用规则 *A*5 映射。

3.2.3 实体完整性约束映射

为尽量保持关系数据库中的语义信息,对 SQL 中的实体完整性约束(key、unique、not null 等)的映射是必不可少的。表 4 给出了一组实体完整性到框架逻辑的映射规则。

表 4　实体完整性映射

constraint	fLogic definition
key	forall *C*, *A* key(*C*,*A*) <-unique(*C*, *A*) and notNull(*C*,*A*)
unique	forall *C*, *A* unique(*C*,*A*) <- identicalValues(*C*,*A*)and singleValue(*C*,*A*)
not null	forall *C*,*A* notNull(*C*, *A*) <- forall IC exists IA IC:C and IC[*A* - > > IA]
identical values	forall *C*,*A* identicalValues(*C*,*A*) <- forall *I*1, *I*2, IA *I*1:*C* and *I*2:*C* and *I*1[*A*- > > IA] and *I*2[A- > > IA] and equal(*I*1,*I*2)
single value	forall *C*, *A* singleValue(*C*,*A*) <- forall IC, IA1, IA2 IC:*C* and IC[*A*- > > IA1] and IC[*A*- > > IA2] and equal(IA1, IA2)

3.3 数据迁移

数据迁移是在本体创建完成后,将关系数据库中的元组映射到本体实例,形成知识库。数据迁移过程的步骤如下:

a)根据元组创建本体实例,为每个实例分派一个惟一的标志,将无参照完整性约束的属性迁移到本体实例中;

b)根据参照完整性(引用关系标志关系之间的联系)创建本体实例之间的关系。

下面给出了数据迁移的例子:

dept(deptID,deptName){"3","Computer Science"}
student(stuID,deptID references dept){"2","3"}
id1:dept[deptID - > > 3,deptName - > > Computer Science].
id2:student[studID - > > 2,deptID - > > id1].

4 系统实现与验证

基于 C/C ++ 和 Redland RDF Library[11] 实现了基于规则的关系数据库到本体的转换系统。图 2 给出了系统的实现架构。其中数据源适配器负责从关系数据库中抽取模式信息和数据;关系分析器负责创建本体概念和概念间层次关系;本体概念创建完成后,属性及参照完整性分析器负责属性到本体谓词的映射;实体完整性分析器负责字段的实体完整性约束到本体属性限制的映射。这三个模块完成模式转换,输出为不带实例信息的本体结构。数据迁移模块根据获取的本体结构将数据库中的记录迁移到知识库,最终输出为 RDF 文档。

为验证转换规则的完备性和转换结果的等价性,本文设计了一组等价的 RDQL[12] 查询和 SQL 查询,分别施加在生成的 RDF 文档和原数据库上。该组查询设计覆盖了一个学校的关系模式。实验证明,该方法保持了完整的关系数据库的模式和数据信息,并挖掘了部分深层的语义信息(对象的相互关系)。

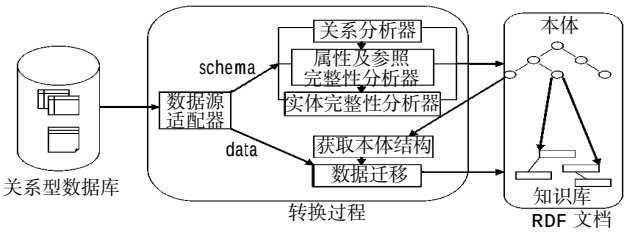

图2　转换系统结构图

下面给出一个 RDQL 查询与对应的 SQL 查询示例。

RDQL 查询:
SELECT ?x
FROM〈school.rdf〉
WHERE〈?x,〈db:deptID〉,?y〉,〈?y,〈db:deptName〉,?z〉
AND ?z = = "Computer Science"
USING db for 〈http://olcalhost:8080/schooldb#〉, rdf for〈http://www.w3.org/1999/02/22-rdf-syntax-ns#〉

等价的 SQL 查询:
SELECT stuID FROM student,dept
WHERE student.deptID = dept.deptID
AND dept.deptName = "Computer Science"

5 结束语

本文提出了一种基于规则的关系数据库到本体的转换方法,通过分析主键、属性、引用关系、完整性约束和部分数据,将关系模式映射到相应的本体结构,集成信息,创建概念层次,然后将数据库中的数据映射到本体实例。该方法最大限度地保持了关系模式的完整性,可自动化地进行转换,完成了基于本体的数据集成的第一步工作——创建本地本体。实现的系统可被用于多种语义数据集成环境中。

然而,在关系映射规则中对数据关系的分析仍然比较简单,如何通过分析关系数据的等价、包含、相交等关系确定本体概念间深层次的关系同时保证其正确性和效率,仍是需要进一步研究的问题。对所实现系统的 API 进行封装,做成可被广泛使用的中间件也是进一步的工作。

参考文献:

[1] LENZERINI M. Data integration: a theoretical perspective[C]//Proc of the 21st ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems. New York:ACM Press,2002:233-246.

[2] 王渊,卢鼎正,陈玉. 一种基于 Ontology 的数据集成方法[J]. 计算机工程与科学,2005,27(6):67-69.

[3] CHIANG R,BARRON T,STOREY V. A framework for the design and evaluation of reverse engineering methods for relational databases[J]. Data and Knowledge Engineering,1996,21(1):57-77.

[4] ASTROVA I. Reverse engineering of relational databases to ontologies [C]//Proc of the 1st European Semantic Web Symposium. Heidelberg:Springer-Verlag,2004:327-341.

[5] STOJANOVIC L,STOJANOVIC N,VOLZ R. Migrating data-intensive Web sites into the semantic Web[C]//Proc of the 17th ACM Symposium on Applied Computing. New York: ACM Press, 2002: 1100-1107.

[6] KASHYAP V. Design and creation of ontologies for environmental information retrieval[C]//Proc of the 12th Workshop on Knowledge Acquisition, Modeling and Management. Alberta:[s.n.],1999:1-18.

[7] DOGAN G,ISLAMAJ R. Importing relational databases into the semantic Web [EB/OL]. (2002). [2006-11-05]. http://www.mindswap.org/webai/2002/fall/Importing_20Relational_20Databases_20into_20the_20Semantic_20Web.html.

(下转第 785 页)

件中是没有的。

2.2.2　多维分析

使用 OLAP 工具,多维多角度分析数据。可以借助切片、切块、上卷、下钻等操作多维分析数据仓库中的数据,异常数据容易被发现,而通常此类信息凭经验是很难发现的。比如在银行存款业务审计时,分析该行某一年的存款数额及利息关系,以时间作为分析的维度,发现某一季度存款额提高,但是利息收入却在下降,因而怀疑该季度有异常存款发生,审计人员继续深入调查该季度的存款数据,通过下钻操作,分析该季度下每个月份的数据,确定是否有异常存款存在。

2.2.3　有指导分析

使用数据挖掘工具发现审计规则,积累审计经验,并且利用审计规则指导审计操作。

数据挖掘可以在任何类型的数据存储上进行,本模型指数据仓库基础上的数据挖掘,通常数据挖掘过程需要的数据提取、数据清洗、数据转换等操作已经在数据仓库建设阶段完成,数据仓库中的数据是没有噪声的、一致的、高质量的数据,为后续的数据挖掘提供了诸多的便利。尽管如此,进行特定的数据挖掘之前,仍需深入地分析。例如在选择某种数据挖掘算法后,要根据需求,筛选出感兴趣的属性,或者对不同的属性赋予不同的权值等。

挖掘审计规则:如图 3 所示,从数据仓库中抽样提取部分数据,作为样本集数据,采用一定的数据挖掘算法,发现某些隐含的规则,将那些有价值的规则更新到审计规则库中。此外,发现规则的过程可以采用多种数据挖掘算法相结合的方式,追求规则尽量准确、完善。

规则指导审计:审计规则库中的规则是审计人员积累的审计经验以及通过数据挖掘发现的具有一定可信度的审计规则的集合,它们可以用来指导审计,以快速发现审计线索,进而重点审计。这样,一方面提高了审计的效率;另一方面,审计规则库不断扩大,审计经验得到积累,并且实现了共享,提高了审计的质量。

现行的贷款五级分类制度将贷款分为正常贷款、关注贷款、次级贷款、可疑贷款和损失贷款。对商业银行贷款业务进行审计时,可以利用数据挖掘技术,发现合理的贷款分类规则,快速定位可疑的不良贷款,进而追踪审计。例如利用决策树方法发现信用贷款占不良贷款的多数,这样审计人员可以明确审计重点,提高审计效率。

2.3　审计深入开展

在前面数据分析的基础上,定位重点审计对象,利用先进的计算机技术或其他方式追踪线索,重点审计该类数据,分析审计结果,提出审计报告。

3　结束语

本文提出的新型的审计模型——审计数据仓库模型,可以指导审计人员高效地开展审计工作。该模型利用数据仓库技术,有效地解决了面向海量数据的审计;利用 OLAP、数据可视化等技术,多维、方便、快捷、直观地分析数据,快速发现异常数据,提供审计线索,提高了审计效率;并且利用数据挖掘等技术发现有价值的审计模式,结合审计人员的经验,实现了审计经验知识的积累与共享,并指导审计人员快速定位审计重点,提高了审计的效率及质量。

该模型的提出,为审计软件今后的发展提供了思路,同时对智能审计作了初步探索,智能审计未来的发展需要借鉴数据挖掘、人工智能等先进技术的发展,有待进一步的研究。

参考文献:

[1] 胡荣,陈月昆. 数据挖掘——现代审计处理数据的新方法[J]. 中国审计, 2004(7):38-40.

[2] 陈明秀. 浅议数据仓库[J]. 科技资讯, 2006(12):247-248.

[3] INMON W H. Building the data warehouse[M]. 2nd ed. 王志海,等译. 北京:机械工业出版社, 2000:1-214.

[4] 梅伟恒,康晓东,江玉彬. 基于数据仓库的 OLAP 技术的研究综述[J]. 中国科技信息,2006(14):134-135,138.

[5] 张丽. 数据仓库与数据挖掘[J]. 贵州民族学院学报:哲学社会科学版,2006(2):204-206.

[6] SIRIKULVADHANA S. Data mining as a financial auditing tool [D]. [S. l.]:The Swedish School of Economics and Business Administration, 2002.

[7] KOTSIANTIS S, KOUMANAKOS E, TZELEPIS D, *et al*. Forecasting fraudulent financial statements using data mining [J]. International Journal of Computational Intelligence, 2006, 3(2):104-109.

[8] HAN Jia-wei, MICHELINE K. Data mining concepts and techniques [M]. 范明,梦小峰,等译. 北京: 机械工业出版社, 2001:1-332.

[9] WRITTEN I H, FRANK E. Data mining practical machine learning tools and techniques [M]. 2nd ed. 北京: 机械工业出版社, 2005: 1-483.

[10] 王春梅. 基于数据仓库的数据挖掘技术[J]. 西安邮电学院学报, 2006,11(5):99-100.

[11] 景波,刘莹,文巨峰. 关联规则在计算机辅助审计中的应用[J]. 计算机工程与应用,2006,42(25):210-212.

[12] [EB/OL]. [2007-01-19]. http://blog. csdn. net/truexf/archive/2006/09/05/1180313. aspx.

[13] 黄松英,何绍木. 金融审计 OLAP 模型技术分析与设计[J]. 现代计算机:专业版,2004(5):46-49.

[14] 王忠,武哲. 数据挖掘在审计信息分析中的应用[J]. 计算机应用研究,2005,22(2):167-169,193.

[15] 易仁萍,陈耿,杨明,等. 数据挖掘技术及其在审计风险管理中的应用[J]. 审计与经济研究,2003,18(1):3-6.

(上接第 770 页)

[8] 任保锋,肖卫东,唐九阳,等. 关系模式到 OWL 的映射研究[J]. 计算机应用研究,2006, 23(9):33-35.

[9] KIFER M, LAUSENL G, WU J. Logical foundations of object- oriented and frame-based languages[J]. Journal ACM, 1995, 42(4):741-843.

[10] DECKER S, BRICKLEY D, SAARELA J, *et al*. A query and inference service for RDF [EB/OL]. (1998-12-03). http://www. w3. org/TandS/QL/QL98.

[11] BECKETT D. Redland RDF libraries[EB/OL]. [2006-10-15]. http://librdf. org/.

[12] SEABORNE A. RDQL:a query language for RDF[EB/ OL]. (2004-01-09). [2006-10-15]. http://www. w3c. org/Submission/RDQL.