基于 XML 的多层次范例映射模型*)

汤 胤[1] 彭 宏[2] 郑启伦[3]
(华南理工大学计算机科学与工程学院 广州510640)

摘 要 本文以软件系统设计的 CBR 方法为例,针对以往 CBR 范例表示的缺陷,深入研究了基于 XML 的范例表示,建立了基于 XML 的多层次范例映射模型,随之探讨了在这个模型下 CBR 方法中的一些关键问题,最后给出了概念层次和概念层次树的定义和概化方法。

关键词 范例推理,范例表示,XML,概念提升

XML Based Multi-Hierarch Mapping Model of Case Representation

TANG Yin[1] PENG Hong[2] ZHENG Qi-Lun[3]
(Institute of Computer Science, South China University of Tech. GUangzhou 510640)

Abstract With an example of a CBR system of software designing, we focus on XML-based case representation method, to avoid shortcomings of previous solution of case representation. A multi-hierarch mapping model based on XML solution is proposed, accompany with the solution of several CBR key issues. At the end of the article, the definitions of concept hierarch and its generalization method are given.

Keywords CBR, Case representation, XML, Concept generalization

1 引言

基于范例推理(Case-Based Reasoning,CBR)在人工智能领域是一种非常有用的方法,引起了人工智能研究者的广泛关注,它已经应用到辅助复杂的工程设计如建筑业、工程制造等行业,显示出广泛的应用前景。

常见的 CBR 系统描述范例一般使用静态框架,这对系统的实用性有很大的局限性。首先,系统在应用中会引入许多新的范例,静态框架很难容纳新范例的新结构和特性[4];其次,CBR 最适用于具有大量值得借鉴和重用历史记录的行业如建筑设计、软件设计、法律、审计等等,从这些记录中提取信息重新构造框架表示是一项繁重而粗糙的工作;再次,CBR 系统没有比较成熟的表示和存储模型,在 CBR 的应用中将出现很多问题,往往对每个应用系统都得从头全面设计;同时,目前的 CBR 系统对范例库也缺乏完善的管理,这些都阻碍了 CBR 系统的广泛应用。

Lorcan Coyle[1]等提出了使用 CBML 语言描述范例的方法,解决了使用 XML 进行范例表示的问题,但没有给出合适的 XML 与范例映射模型。Chen Chu-Xun[4]研究了利用数据库进行范例存储和表示的技术,提出了基于关系数据库的范例模板构造方法,但由于关系数据库本身固有的结构化特性,难以描述比较复杂的 CBR 范例。本文研究 XML 与范例模板的映射问题,给出了一个新的范例表示与多级 XML 的映射模型,可以较好地解决上述问题。

2 当前范例表示的问题

2.1 静态范例模板不具备灵活性

常见的 CBR 系统对范例的描述是通过预先定义静态的范例模板来实现。以标准的静态方式描述范例可以大大简化 CBR 检索和匹配过程,而且直观。在一些小规模应用中,可以把不同角度提取的信息作为一个个子集加到范例中去,以支持对范例的多重利用。这种针对特定应用的需求来预定义范例模板的方法在大规模应用中并不适用。在大规模应用中,对知识的重用不局限于几个可以预见的需求,支持多目的、多角度、多层面的知识重用是非常重要的。范例模板扩充或改变的问题不能很好地解决,则当出现新需求时,通过对原有范例模板进行简单的修改不能满足应用要求。实际上,不同的工作对于同一批数据的观察角度常常是不同的,同一观察者的视角也有可能经常改变,因此采用静态范例模板的方式有很大局限性[4]。

2.2 基于关系数据库方法的局限性

范例通常描述了一个问题的解决,因此一个范例往往要容纳各种不同格式不同结构的数据。由于其数据种类繁多和结构复杂,擅长处理结构化数据的传统关系数据库在处理大量存在于各种媒体的非结构化数据如图形、图像、声音等时,其信息系统显得难以胜任。数据的存储有多种方式,关系数据库是最广泛使用的一种形式,适合表示相对较为结构化的内容。然而范例往往是非结构化、但又有一定规律的数据集合。本文提出了基于 XML 的范例表示,可以克服范例数据非结构化等等问题。

3 XML 的范例解决方案

3.1 XML 介绍

XML 数据库擅长于处理半结构化的信息,特别适合表达层次型的数据,具有良好的灵活性。XML 不但可以完全替代关系数据库的二维关系存储方案,还可以表达数据结构:如

*)广东省科技攻关项目(A1020103)。汤 胤 博士生,研究方向:智能计算技术;彭宏 教授,博导,研究方向:智能计算技术;郑启伦 教授,博导,研究方向:智能计算技术,神经网络理论及应用。

树、图、序列等等多种形式。其本身所具有良好的伸缩性和表现形式的多样化解决了范例库的管理和扩充问题[1]。

3.2 核心思想

我们的想法是，底层的数据并不直接表示范例，而是用在底层 XML 数据基础上建立多级视图来完成范例表示。在 XML Case Base 的基础上建立多级概念层次结构，妥善解决索引、检索和模板引用等问题。

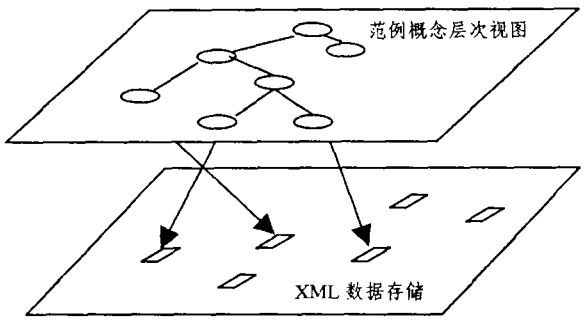

图1 多级范例概念层次视图与 XML 的映射

图1中，视图与 XML 存储间的映射是动态的，它可以使 CBR 系统有更多的灵活性，与关系数据库的表示方法的不同之处是范例按照一定的规律组织在一起，而不是像关系数据表示方法中的二维线性表，更能准确地反映范例之间的关系，同时在 XML 数据线性存储以外建立多级范例概念层次视图（如图1上方所示）。有以下优点：(1)数据共享，可以在同一个 XML 文件上相互链接多重引用，建立多个视图，建立在同一数据源上的 CBR 系统可以很轻松地实现数据共享，实现不同 CBR 系统间通信；(2)范例模板灵活，当需要建立新的特征的时候，只要添加新 XML 标签即可，比数据库添加字段要容易维护得多；(3)可移植性，当 CBR 系统移植到新的环境中时，只需重新配置范例与 XML 之间的映射，不需要修改底层数据；(4)任一级的抽象范例均可视乎需要作为设计模板使用。

4 概念分层映射模型

4.1 定义

定义1(范例偏序) 假设范例 a 是从范例集 B 中提取共同特征而得到的，则对任意范例 b∈B，定义 b≺a。

定义2(范例概念层次) 一个概念层次 H 是一个偏序集 $(h,\prec)$，其中 h 是个有限范例集，$\prec$ 是 h 上的范例偏序。

定义3(用户视图) 用户根据应用需求，有选择地从范例库中提取一定结构的相关数据就构成了用户视图。

定义4(范例概念层次树(Concept Hierarch Tree)) 如果把用户视图的内容按照抽象概念层次结构的方式来组织就构成了该范例库的范例概念层次树。

树中高层概念是层概念的概括，树根是该属性的可能属性值的最一般描述，叶是该属性的可能属性值，如果属性是连续值属性，则树叶是一些连续值范围。显然，树中，叶节点为实际范例节点，内节点为抽象节点。

现定义范例模型为：

$$S=(C,M,R_M) \tag{1}$$

其中，S：系统；C：系统的基本特征；M：功能模块集合；R：模块集合 M 上的一个关系。

显然 M 是一个递归关系，因为对于整个系统，模块本身也是从上到下逐步细化的，而 R 牵涉到 M 之间发生相互作用的所有事件。

考察 CBR 系统中一些关键的问题。

类比映射是 CBR 系统的关键步骤，反映在模型里，是比较 M 和 R 的相似性，其中结构相似性将成为主要考虑的因素，另外还有语义相似性、目标相似性。这个抽象模型可以比较恰当地反映要设计的软件系统。

模型分析和特征提取用于分析目标范例，从中识别和抽取检索源范例库的信息。在模型里，可以将目标范例进行多级概化，然后根据抽象属性，按需要分别匹配范例库每一级的范例。在较为具体的层次往往难以匹配到合适的范例，可以将目标范例进行概化，特征提取就变得十分方便。

对于范例表示题，同样地定义 XML 系统模型为：

$$X=(f,m,d) \tag{2}$$

其中，X：XML 系统；f：特性；m：模块；d：DTD 辞典，对 f，m 进行约束，并定义 m 中元素关系。

则映射模型可用三元组表示为：

$$F=(S,X,D) \tag{3}$$

其中，F：XML 的范例映射；S：范例概念层次树；X：XML 系统；D：范例概念树与 XML 数据项的映射集合。对范例的 XML 映射过程可以以下面式子来定义：

$$F:S(C,M,R_M)\xrightarrow{R_M}X(f,m,d)$$

4.2 几种语义关系的 XML 映射方法

建立了模型之后，有必要研究 R_M 内部的问题，对一些基本的语义关系进行定义，下面给出了一些语义关系的 XML 映射方法：

(1)类别关系 $A_{\underline{ISA}}$ B：

〈tag name = "A" type = "B"〉

(2)集合关系 $A_{\underline{Subset\ Of}}$B：

A 作为 B 的子树，如：〈B〉〈A〉〈/A〉〈/B〉

(3)类别关系 $A_{\underline{AKO}}$B：

B 作为类型，A 作为其名〈B〉A〈/B〉

(4)成员关系 $A_{\underline{A\ Member\ Of}}$ B：

〈B〉A〈/B〉

(5)部分关系 $A_{\underline{A\ Part\ Of}}$ B：

〈B〉A〈/B〉

(6)拥有关系 $A_{\underline{Have}}$ B：

〈A〉B〈/A〉

(7)相似关系 $A_{\underline{Similar\ to}}$ B：

〈similar〉〈item〉A〈/item〉〈item〉B〈/item〉〈/similar〉

(8)因果关系 $A_{\underline{Infer}}$ B：

〈infer〉〈reason〉A〈/reason〉〈result〉B〈/result〉〈infer〉

4.3 范例模板映射过程

引用一个软件系统设计的例子来说明模板的构建。

STEP 1 构建领域知识辞典

(1)确定 DTD 文件，选择语义关系对应的 XML 表示方法；

(2)为属性间相似性和属性内部值提供度量方法；

(3)将系统特征和模块根据知识辞典进行转换。

STEP 2 范例概化过程

在 XML 所包含的原始信息中，有的部分可以直接映射到范例中，有的信息则需要加工提炼后才能被利用，这个过程就是概化过程，经过多次概化的范例构成范例概念分层。建立抽象的范例层次结构的目的是将烦杂的原始信息分类，使得同一类原始信息具有某种共同的特征，这个特征就是从原有信息中提取的抽象信息。

```
<case name="知识管理系统" type=2 行业="审计">
  <系统描述>... ...<系统描述>
  <父范例>... ...</父范例>
  <用户部门>
      <部门>市场部</部门><部门>研发部</部门><部门>财务部</
      部门> ... ...</用户部门>
  <技术选型>
      <技术架构 type="三层">
        <feature id="表示层">ASP</feature>
        <feature id="业务层">VC</feature>
        <feature id="数据服务器">SQL 2000</feature>
      </技术架构>
      <服务器>
        <操作系统>windows 2000 server
        </操作系统>
            <运行服务 type = "web" >Apache</运行服务>
            <运行服务 type = "ftp" >IIS</运行服务>
            <运行服务 type = "web" >... ...</运行服务>
      </服务器>
      <客户机></客户机>
  </技术选型>
  <系统结构>... ...</系统结构>
  <物理拓扑图>
      <设备 type = "服务器" id="server1"></设备> ... ...
      <连接><from>server1</from><to>server2</to></连接> ... ...
  </物理拓扑图>
  <功能模块>
      <模块 id=234 name="">... ...</模块>
      <模块 id=235 name="">... ...</模块>
      <模块 id=236 name="">... ...</模块>... ...
  </功能模块>
  <模块关系图>
      <关系 type="" ><from >... ...</from>
          <to>... ...</to></关系> ...
  </模块关系图>
</case>
```

图2 一个软件系统设计范例

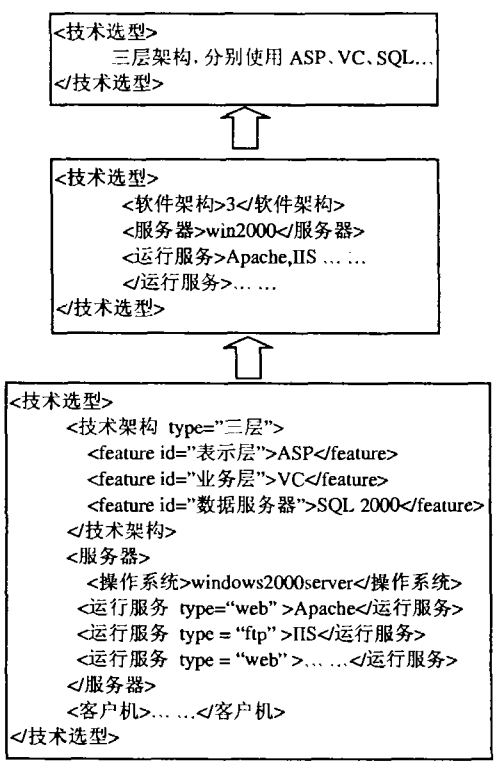

图3 原始范例及其概念提升

概念提升应包括：特征的提取概略、模块的简化（结构性描述转为辅助语言描述）。

原始设计案例中，可以以关键字形式刻画范例中所描述的特征，以便进行范例的检索。范例概化的目的是构造简洁抽象的范例模板来支持高效的检索并且提供不同程度抽象的范例模板，范例概化遵循下面原则[4]：

(1)刻画所有与检索有关的范例特征，但不包含那些对设计师有价值而与检索无关的信息，如设计图片，辅助说明等；

(2)所刻画的信息足以区别不同的设计范例。

STEP 3 范例概化后，将这样形成的范例作为原范例的父节点，保存父节点指针，由此逐步提升，最终形成范例概念层次树。

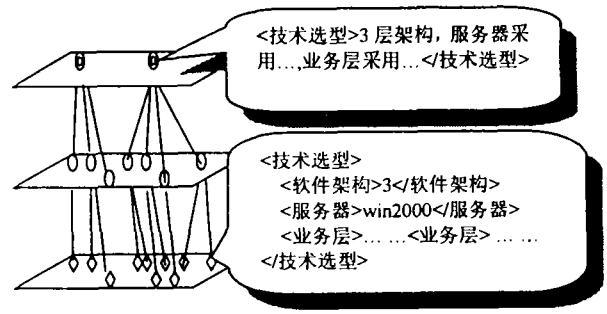

图4 范例按照属性、概念、案例类别分别进行索引

依照这种原则，在范例树中选取那些与检索有关的具体节点连同其上层节点构成一棵子树，称之为多层次多索引的概念树（包括约束、类别、情景、因果、相似、相反、归属关系等等），这样的概念提升，既完成了范例的索引过程，每个抽象范例又可以作为模板使用，可以在概念树的任一层次选择一个模板进行泛化(materialization)。

结论 在软件系统设计 CBR 系统开发过程中，结合 XML 技术是 CBR 方法以及其他一些人工智能思想走向应用的重要手段。通过范例的多级 XML 视图，随时可以引用不同等级的抽象模板，提高索引和检索效率，解决了范例模板的扩充修改问题，为 CBR 范例和 XML 映射提供了一个成熟可用的模型，使得 CBR 系统更加通用。

参 考 文 献

1 Cole L, Hayes C, Cunningham P. Representing Cases for CBR in XML

2 Hayes C, Cunningham P. Shaping a CBR view with XML

3 Leake D B. CBR in Context: The Present and Future. AAAI Press/MIT Press, 1996

4 CHEN Chu-Xun, WANG Ying-Lin, ZHANG Shen-Sheng. Case Storage Based on Relational Database. Journal of Shanghai Jiaotong University, 2000, E-5, (2): 65～69

5 史忠植. 知识发现. 清华大学出版社, 2002

6 韩嘉伟. 数据挖掘概念与技术. 机械工业出版社, 2000