

基于 XML 的面向对象知识表示模式设计^①

XML - Based Object - Oriented Knowledge Representation Pattern Design

徐大庆 (中国科学院 合肥智能机械研究所 安徽合肥 230031)
(中国科学技术大学 自动化系 安徽合肥 230027)

李 森 袁 媛 (中国科学院 合肥智能机械研究所 安徽合肥 230031)

摘 要: 在传统知识表示方法的基础上,综合利用了框架、产生式规则、面向对象等知识表示方式的特点设计了一种基于 XML 的新型知识表示方法,并完成其 XML Schema 的设计。此方法将普通规则与模糊规则统一起来,充分利用了 XML 对结构化数据的表示能力,实现了面向对象与产生式规则方式混合表示知识结构化存储。兼顾了知识库的机器可读性和人可读性,实现了知识表示的跨平台、跨操作系统性能。以此为基础可开发出适于多种应用领域的交互性好、易扩展的知识获取平台及相应的推理机。

关键词: 知识表示 面向对象 XML 产生式规则 专家系统

1 引言

专家系统是一种设计用来对人类专家的问题求解能力建模的计算机程序^[1]。专家系统的研究和设计着重于知识处理,包括知识的获取、表示和应用三个核心环节。其中,知识表示是研究用机器表示知识的可行性、有效性的一般方法,是一种数据结构与控制结构的统一体,既考虑知识的存储又考虑知识的使用。知识表示方式的选择与设计不仅对知识的有效存储有关,也直接影响着系统的知识获取能力和知识的应用效率,因而,知识表示是知识工程中最基本问题之一。

在目前的专家系统中,知识表示方法主要有逻辑表示法、产生式规则表示法、状态空间法、框架表示法、语义网络表示法、面向对象表示法和神经网络表示法等。几种主要的知识表示方法各有局限,例如:①框架的固定性使得很多时候实际情况与原型不符^[2];②产生式系统难于表达过程式知识。

采用面向对象技术的知识表示方法是最近研究热点。从认识论的观点出发,面向对象的思想比较符合

人们的一般思维过程,因而很快就渗透到了计算机的多个领域,其中一个很有发展前景的分支就是面向对象的知识表示和方法^[3]。它能在不同层次表达知识,在高层次对象能封装复杂的行为,而具体细节对该层知识又是透明的,还可以构造相关信息并把它们保持在一起,灵活性好,优越性比较显著。

2 专家系统开发平台

为了提高专家系统的开发效率、质量和自动化水平,人们已经研制了多种专家系统开发平台。专家系统开发平台的一种典型结构如图 1 所示。

可见,知识库处于专家系统开发平台的核心位置,知识表示方式的选择与设计对于专家系统开发工具的研制至关重要。本文以中国科学院合肥智能机械研究所开发的农业专家系统开发平台(DET)中知识描述语言(DET 语言)为参考,设计一种基于 XML 的面向对象

① 基金项目:中科院知识创新工程重要方向项目(KGCX2-SW-511)

知识表示方法并给出 XML 模式。

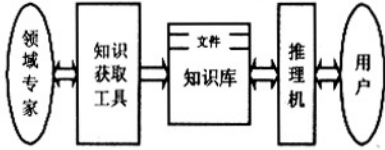


图 1 专家系统开发平台结构简图

DET 是由知识库编辑器、知识库文件、编译器、二进制代码、推理机组成的专家系统开发平台,采用的过程是“知识获取—编译—推理交互”。在多年的推广应用,已用该平台成功开发出了大量实用的农业专家系统,并且在实际的生产中起到了较大的经济效益。平台采用的 DET 语言功能相当丰富,但不易于学习掌握,这使得专家的知识较难得到全面发挥。

DET 语言是一套适用于农业的知识描述语言,它是一种类 C 的高级计算机语言,有 6 种基本数据类型,可产生 183 条指令,具有一般程序设计语言的语法结构。DET 的推理机采用对二进制代码进行解释执行的方式运行,在一定程度上加快了推理的速度,但这给平台的维护与扩展带来了较高的复杂性。

DET 采用以 kb 为后缀的文本文件存储 DET 语言知识库,经编译器编译成二进制代码供推理机使用。kb 文件表示的知识库给专家系统开发平台带来以下困难:(1)知识表示过于离散,难以表示知识库的框架结构,使得可视化知识获取界面中的知识节点定位较困难;(2)难以设计结构清晰的推理机对结构化程度低的 kb 知识库进行直接推理;(3)DET 语言的编译器及解释执行二进制代码的推理机难以维护与扩展。采用面向对象知识表示方法和 XML 文档存储知识库方式可以解决以上问题。

3 面向对象知识表示 XML 模式设计

3.1 数据类型

面向对象的知识表示方式需要涵盖高级语言的相应功能,如变量定义、控制语句、输入输出语句等。

在数据类型选择上,以知识库的可用性、简洁性及兼容性为目标,采用的基本变量类型有: int、float、date、string、char、mdate;基本扩展类型:数组类型,多

媒体类型;用户自定义类型:类、枚举、多选及模糊变量类型。其中基本类型的前 4 种简单类型直接由 W3C 的 XML 模式内置简单类型借鉴过来。mdate 用来表示月与日的组合类型。

几种类型的常量表示模式如下:

```
<xs:simpleType name = " MinDecimalType" >
  <xs:restriction base = " xs:decimal" >
    <xs:minInclusive value = " 0" />
    <xs:maxInclusive value = " 1" />
  </xs:restriction >
</xs:simpleType >
```

```
<xs:simpleType name = " MdateType" >
  <xs:restriction base = " xs:token" >
    <xs:pattern value = " \d{1,2} \- \d{1,2}" />
  </xs:restriction >
```

```
<xs:simpleType name = " ConstMultimediaType" >
  <xs:restriction base = " xs:string" >
    <xs:pattern value = " M&quot;( \s| \S) * . ((txt) | (htm) | (html) | (gif) | (jpg) | (bmp) | (rm) | (wav) )&quot;" />
  </xs:restriction >
</xs:simpleType >
```

MinDecimalType 主要用来表示可信度、阈值等常量数据信息,ConstMultimediaType 用来表示多媒常量信息。另外还有 6 种常量类型模式未在上表中列出,各种类型数据的语义错误由知识获取工具专用的检查工具进行验证。

3.2 知识库总体结构

框架是一种表示概念或对象一成不变知识的数据结构,比较适合表达结构性知识。既能存储陈述性知识又能存储过程性知识,所组织的知识具有层次性。

知识库及知识模型结构符合框架的上述特点,借鉴框架的概念与结构,结合 XML 文档表示方法,知识库根节点结构如下:

```

<知识库 名称 = "..." >
  <版权信息 >
    <版权所有 >...</版权所有 >
    <作者 >...</作者 >
    <开发日期 >...</开发日期 >
    <修改日期 >...</修改日期 >
  </版权信息 >
  <知识模型 >...</知识模型 >
</知识库 >
    
```

在本文的 XML 文档示例表示中,用“...”代表属于固定结构以外或由于篇幅原因省略的内容。

3.3 知识模型

知识库的可用知识绝大部分存储在知识模型结点中,知识模型是递归定义的,可嵌套多个子知识模型,从而形成树状结构,每一层有其相应的各种库来存储不同类型的知识。知识模型节点结构设计如图 2 所示:

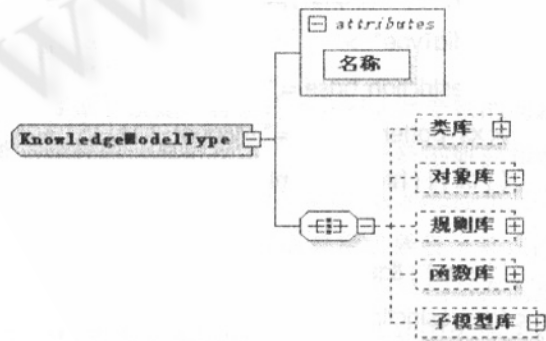


图 2 知识模型节点 XML 模式图

其中,类库由枚举类型、多选类型、模糊变量类型、类 4 种类型的任意个节点构成。对象库由多个不同类型的对象构成,规则库由多条规则构成,函数库由任意条函数和至多一条主函数组成。子模型库是由可嵌套定义的任意个子模型构成。

3.4 类库

从面向对象的角度来看,人们在认识问题和分析问题时,可以把问题分解为一些对象以及对象之间的组合和联系。面向对象的知识表示方法考虑了现实世界与面向对象表示空间的关系,可以简化现实世界的

问题。面向对象的知识表示方法采用数据抽象和信息隐蔽技术,以抽象数据类型为基础,将数据和对数据的操作放在一起,作为一个相互依存、不可分割的整体来处理,并将这种整体抽象成一种新的数据类型——类。可以较好地描述了现实世界的结构模型。

类的结构设计面向对象高级语言相似,其 XML 模式如图 3 所示:

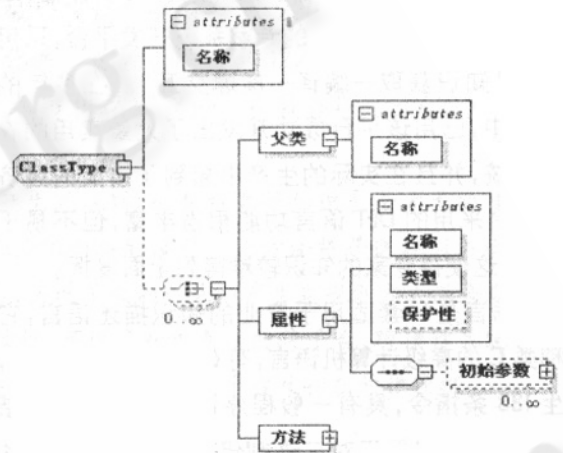


图 3 类节点 XML 模式图

其中的元素“方法”由复杂类型(函数定义类型 FunDefineType)来定义, FunDefineType 类型包括 3 个属性(名称、返回类型、保护性)、任意多个形式参数及一个函数体类型。同高级语言控制结构相对应,函数体类型由任意多个句子、分支结构或循环结构组成。

除了类以外,为增强知识表示的功能性与智能性,在类库中还设计了枚举、多选、模糊变量类型等自定义类型。模糊变量类型节点结构如图 4 所示。

模糊集合有连续型和离散型两种,本文统一定义在模糊变量类型元素节点下,用属性“连续性”来区分。

3.5 规则

规则指关联已知知识和待推导或推测的其他信息的知识结构^[1]。规则是最自然地表示专家经验性知识的表示方法,多条规则组合构成规则库。根据实际需要,规则可分为三类:普通规则、模糊规则及混合规则。普通规则是指规则前件变量和后件变量中都不包含模糊变量的规则,模糊规则指规则前件变量及后件变量

都不包含普通变量的规则,其他类型的规则称之为混合规则。为了降低推理机设计的难度,用同一规则的前件表示合取关系,用同一后件的多条规则来表示析取关系。为了同时支持精确及不精确推理,对于三种类型的规则都设置可信度及激活阈值。规则结构设计如图 5 所示。

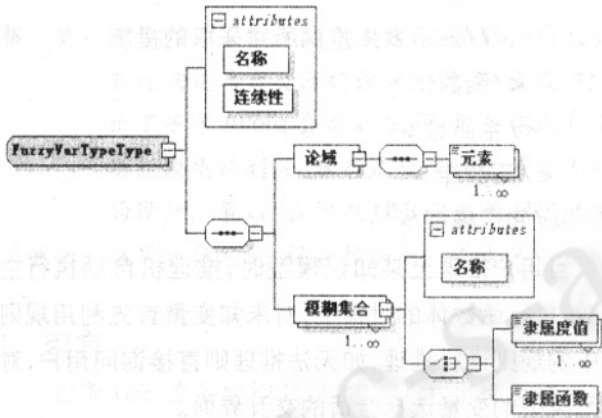


图 4 模糊变量类型节点 XML 模式图

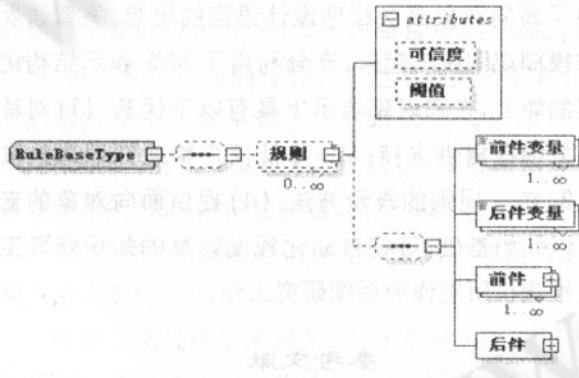


图 5 规则节点 XML 模式图

在规则表示中,除前件与后件外,还同时设计了前件变量及后件变量节点,此设计方式是为了更容易形成推理链,其中的前件变量及后件变量应由知识获取工具从规则中直接提取。前件由逻辑表达式、变量、常量或函数调用组成,后件的定义同 3.4 中提到的函数体类型。

4 应用实例

以生猪诊断专家系统为例,利用本文设计的知识表示方法可得出知识库片段的简略表示如下:

```
<知识模型 名称=" 生猪常见疫病诊断" >
  <类库 >
```

```
    <多选类型,名称=" 精神状况及运动症状列表" 元素类型=" string" >
```

```
      <常量>" 懒散、嗜睡、怕冷" </常量>
```

```
      <常量>" 肌肉颤抖或发寒颤" </常量>
```

```
      <常量>" 惊厥、抽搐、头向后仰" </常量>
```

```
      ...//其他症状常量
```

```
    </多选类型 >
```

```
    <多选类型 名称=" 疫病列表"
```

```
      元素类型=" string" >
```

```
      <常量>" 猪乙型脑炎" </常量>
```

```
      <常量>" 猪伪狂犬病" </常量>
```

```
      ...//其他疫病
```

```
    </多选类型 >
```

```
    ...//其他自定义类型,如其他症状列表
```

```
  </类库 >
```

```
  <对象库 >
```

```
    <对象 名称=" 精神状况及运动症状"
```

```
      类型=" 精神状况及运动症状列表" / >
```

```
  </对象 >
```

```
  <对象 名称=" 其他症状"
```

```
    类型=" 其他症状列表" / >
```

```
  <对象 名称=" 疫病" 类型=" 疫病列表" / >
```

```
  </对象库 >
```

```
  <规则库 >
```

```
    <规则 可信度=" 0.9" 阈值=" 0.7" >
```

```
      <前件变量 >精神状况及运动症状 </前件变量 >
```

```
      <前件变量 >其他症状 </前件变量 >
```

```
      <后件变量 >疫病 </后件变量 >
```

```
    <前件 >
```

```
      <函数调用 名称=" 精神状况及运动症状.已选中" >
```

```

<参数>
  <常量>"惊厥、抽搐、头向后仰" </常
量>
  </参数>
</函数调用>
</前件>
<前件>
  <函数调用 名称="其他症状.已选中">
    <参数>
      <常量>"流产、死胎、畸形胎或木乃伊
胎" </常量>
      </参数>
    </函数调用>
  </前件>
<前件>
  <函数调用 名称="其他症状.已选中">
    <参数>
      <常量>"睾丸肿大" </常量>
    </参数>
  </函数调用>
</前件>
<后件>
  <句子>
    <函数调用 名称="疫病.选择">
      <参数>
        <常量>"猪乙型脑炎" </常量>
      </参数>
    </函数调用>
  </句子>
</后件>
</规则>
…//其他规则
</规则库>
<函数库>
  <函数>… </函数>
…//其他函数
<主函数>
  <函数体>

```

```

<句子>
  <函数调用 名称="DISP">
    <参数>
      <变量>疫病 </变量>
    </参数>
  </函数调用>
</句子>
…//显示发生疫病所需采取的措施
</函数体>
</主函数>
</函数库>
</知识模型>
当用户推理至某知识模型时,推理机自动执行主
函数,执行函数体的过程中,对未知变量首先利用规则
库中的规则进行推理,如无法推理则直接询问用户,对
不同类型的变量选择合适的交互界面。

```

5 总结与展望

本文设计的基于 XML 的面向对象知识表示方法采用了面向对象高级程序设计语言的思想,将普通规则与模糊规则统一起来,充分利用了 XML 表示结构化数据的能力,在知识库表示上具有以下优势:(1) 对于对象编程提供支持;(2) 降低用户构建知识库的难度;(3) 统一规则的表示方法;(4) 提供面向对象的支持。以此为基础,可将自动化程度较高的知识获取工具及推理机研制作作为后续研究工作。

参考文献

- 1 蔡自兴、(美)约翰·德尔金、龚涛,高级专家系统:原理设计及应用[M],北京:科学出版社,2005.
- 2 周涛、夏天,基于 XML 的面向对象的知识表示[J],黔南民族师范学院学报,2003(6):34-37.
- 3 彭琳、杨林楠、张丽莲,基于面向对象知识表示的农业专家系统的设计[J],农机化研究,2007(2):166-168.
- 4 张学学、朱乃立,知识表示在牡丹栽培技术专家系统中的应用[J],计算机工程,2005(9):210-214.