

95-58 面向对象的知识表示方法

TP18

王宽全

(西南师范大学计算机科学系 重庆 630715)

A

摘要 本文在分析了传统的知识表示方法基础上,着重探讨了面向对象的知识表示的方法、特点、推理及程序实现问题。

关键词 知识表示,面向对象,封装,推理。

一、引言

知识表示是知识工程研究的重要问题。传统的知识表示方法,诸如一阶谓词逻辑、产生式规则、语义网络、框架等,在这一学科的发展过程中,起到了非常重要的作用。但随着需要处理的知识和数据的大量增加,人们发现单一的知识表示已不能满足需要,于是提出了混合知识表示^[1]。例如,将产生式与框架结合,或者将产生式、框架、过程结合等等^[2]。这些研究促进了知识工程的发展,出现了一些含多种知识表示的专家系统及专家系统开发环境^[3]。

虽然混合知识表示增强了知识表示能力,也提高了知识表示效率,但混合知识表示仍存在问题。例如,它的结构性不强,使系统更加复杂,难于管理。对于一个大的知识系统来说,几种知识表示不易配合,使知识工程师难于维护。基于这种情况,国内外的研究者又提出了面向对象的知识表示这个概念,并作了深入的研究,取得了一些成果。例如,关于城市管理的专家系统 URBANEX^[4],面向对象的混合智能 CAD 系统^[5],将知识库与数据库结合而提出的面向对象模型 SOM^[6],以及面向对象的专家系统开发环境 X-I^[7]、Z-II^[8]。

本文拟将面向对象的知识表示与传统的几种典型的知识表示方法作比较,说明这种新的知识表示方法的特点,并且详细讨论这

种知识表示的一般方法、推理控制策略及程序实现问题。

二、知识表示的概念

什么是知识表示?这个问题到现在也没有一个大家公认的定义。笔者认为,在人工智能研究者中对知识表示存在三种定义。其一是偏向一般性描述,如 Brachman 和 Levesgue 作了如下定义,“知识表示简单地说就是用某种语言或通讯手段描述或图示某一世界的状态”^[2]。邱涤虹认为,“所谓知识表示是为描述世界所作的一组约定,是知识的符号化”^[1]。

其二是偏向数据结构方面的定义。如吴信东认为“知识表示是关于各种数据结构及其解释过程的组合”^[9]。而史忠植的定义为“知识是命题、规则等的集合,是人们对于客观世界和主观世界信息内在联系的认识。知识表示是用符号数据结构描述知识”^[10]。

其三是形式定义,如王树林对知识表示所作的形式化定义:^[11]

令 $A_i = (a_{i1}, a_{i2}, \dots, a_{in})$, a_{ij} 是知识

$U_i(u_{i1}, \dots, u_{in}), u_{ij} = 0 \mid 1$

$U_i A_i = (u_{i1} a_{i1}, u_{i2} a_{i2}, \dots, u_{in} a_{in})$

$A_i \leftarrow f_i(u_i a_i), i = 1, 2, \dots, n$ (1)

f_i 为定义在 A_n 集合上的谓词, $\leftarrow$ 表示映射,即在 A_n 集合中存在一元素与之对应,则称(1)式为知识表示。

对于知识表示通常有如下要求:①表示能力,即能否将问题求解所需要的各类知识

完全表示出来;②推理效率,即能否有效地利用知识库中的知识完成推理;③正确性,即表示方法是否具备良好定义的语义并保证推理的正确性;④结构性,即表示方法是否具有好的模块化结构,便于知识库的维护。

三、传统的知识表示方法

知识表示方法是知识工程师对领域知识的事实和关系的一种模型化。传统的方法,如语义网络,对象、属性、值三元组,框架及规则都在专家系统中得到广泛使用。每一种方法都有它自己的优点和缺点。

1. 语义网络

语义网络是一种基于广义图的表示方法,其中结点代表概念,结点之间的弧或超弧说明了它们之间的相互关系^[12]。形式地说,语义网络由二元组 (N, L) 组成, N 是用来表示对象及特征的结点集合。对象可以是物理实体,也可以是概念。 L 是结点之间链的集合,代表了结点间的关系。最常使用的链是 is-a 链和 has-a 链, is-a 链表示了类和实例之间的关系,通常有继承特性;而 has-a 链表示结点有一定特征。

2. 对象、属性、值三元组(OAV)

在 OAV 方法中,知识被表示成三元组 (O, A, V) , O 是对象的集合,它们可以是物理实体或概念, A 是属性集合,它表示对象的属性, V 则是对象属性的取值。

OAV 方法实质上是语义网络的一种特殊形式。对象和属性的关系是 has-a 链,而属性与值的关系则是 is-a 链, OAV 比语义网络更结构化,但随着对象数量增加, OAV 系统仍然难以管理。

3. 框架

框架由 Minsky 在 1975 年提出^[13]。它是用来描述对象的,由若干个存储对象信息的槽组成。槽的功能与 OAV 的属性相似,然而框架又与 OAV 不同,它的槽能够包含缺省值、指向其它框架的指针、规则集及过程。框架和 OAV 可以认为是语义网络的特殊情况或子集。这三个系统表示知识的能力是相同

的,不同之处在于它们组织知识的结构的观念。框架的一个卓越特点是它能容易地抓住理解特殊概念或情况的知识,因此,将知识库分割成框架在很多应用系统中变成很普遍了。

4. 规则

规则有两部分,第一部分是通过 AND 或 OR 连接的前提条件,第二部分是结论。当规则的前提条件为真,其结论也为真。在某些系统中,规则可能是通过语义网络或 OAV 来实现的。例如斯坦福大学开发的医疗诊断专家系统 MYSIN。此外,规则也可以用框架来表示,如 Intelli Corp 的知识工程环境。

四、面向对象的知识表示

从数据模型来看,一个对象可能是一个人,一个地方,一件东西,一个组织,一个概念或者是用户想收集和存储的信息^[14]。或者说一个对象就是独立的一组数据及定义在上面的方法集^[15]。存在共同结构和行为的事物可以组成一个类(Class),一个类可以有若干个对象(Object)。我们使用对象的集合来表示知识。因此,知识可以形式化地用三元组来描述: $K = (C, I, A)$ 。 C 是类的集合, I 是实例对象的集合(Instance object), A 是类及对象的属性集合。

1. 类

类是用来描述一组属性相似的对象,它决定其对象的行为。每一个类在系统中有唯一的名称,以及定义其行为的属性集。一个类可以是某一个类的子类,从而可以从父类那里继承所有行为。

对类来说,可能存在以下五个方面的属性:

名称:类名,是系统中类的唯一标识符。

超类(Super Class):类的父类名。

类变量(Class Variables):该类所有对象共享的变量集,对该类对象来说属全局变量。

实例对象变量(Instance Variables):类的每一个对象所具有的私有变量,对这些变量,每一个对象可以取不同的值。

类属性(Class attributes):类的各对象共享的方法(methods)、外部方法(external methods)和规则(rules)之集。

2. 对象

对象是类的成员。他们的特性通过其类而定义。每一个对象由三个属性集组成:

名称:对象的名字,系统中用以标识该对象的唯一标识符。

类:该对象所属的类名。

对象属性:该对象所具有的属性。

3. 属性

属性通过某种类和值来决定。属性的种类在类中定义,而值在类或对象里定义,属性可分为三种类型:

①数据:这种属性用来存储简单的数据,如整数、实数、布尔变量及字符串,对这些属性可进行存取、运算等操作。

②数据库数据:这种属性与关系数据库中某一记录相对应,为了定义这类属性,知识工程师必须指明数据库名、字段名、字段类型,这些数据可以通过数据库接口来获得。

③方法:方法不是用来存贮数据,而是用来定义对象的能力。它可以是过程、外部过程、规则等,方法是处理对象中的数据,传送和接收消息的唯一途径。

五、面向对象知识表示是最结构化的知识表示方法

语义网络的主要优点是灵活性,因为新结点和链可以按要求不加限制地定义。这种灵活性在面向对象的知识表示中仍然存在而且对象间的关系还可以动态建立,这些关系和语义网络中的链有同样的能力。事实上,面向对象的结构可以看成动态的语义网络。语义网络中的 is-a 链在面向对象的表示中通过类和子类或者类和对象的关系来实现。所以,面向对象的知识表示和语义网络有相同的表示能力,但更加结构化。

语义网络、规则和 OAV,它们的一个共同缺点就是结构化不强,随着对象或规则的增加,系统变得难以管理。这是因为知识表示

不能模块化,规则和对象之间的交互变得非常复杂。当对象或属性的值被修改时,要准确指出对整个系统的影响是很困难的。因此,这种知识表示很难开发和维护,特别是对大型的知识库更是如此。面向对象的知识表示由于结构化和封装特性,与上述三种知识表示方法有明显区别。

框架比语义网络、规则及 OAV 更结构化,因为相关的属性和规则可以形成框架层次结构。然而,框架知识表示的模块性不能清楚地定义,而且缺乏灵活性。在框架系统中,框架间的关系可能是成员关系或子类链,而且不是唯一的。进一步说,在某些系统中,规则可以通过一个框架连接到另一个框架,这些因素大大削减了框架系统的结构性。面向对象的知识表示,很类似于框架,知识可以使用类按一定层次形式来组织。然而,两个类之间唯一可能的关系是子类链,类和例子之间的关系也只有 is-a 链,而规则定义成类中的方法,不可能出现一个规则跨越两个对象的事情。这种方法也可以说是一种基于消息的方法^[16]。数据被专门保存在一个类中,要提取或修改一个数据项,向该对象发送一条消息。在类之外的代码从不直接操作存在类中的任何数据,能够修改数据的唯一途径是类中的方法,这就是面向对象的知识表示的一种封装特性。带来的好处是更加模块化和结构化,特别适合大型知识库的开发及维护。

封装是一项通过为每一模块定义严格的外部接口从而减少模块间的依赖性的技术。如果一个知识模块只能通过预先定义的接口来存取,则称这个知识模块被封装。因此,封装是与消息管理机制密切相关的。如果发送消息者权限不够,消息将受到阻塞,这一机制可以防止越权存取其它对象中的数据。一个对象的属性只能通过它自己的方法来存取。封装的效果是更加模块化。因为封装,对象的内容只能通过特定的方法来存取,各模块只能通过预定的接口而交互。所以,各模块可以分别测试和修改。使用这种形式组织起来的

知识是相当结构化,而且很容易理解和管理。

六、面向对象知识表示的推理

面向对象知识表示的推理可以从三个方面来考察。

1. 继承

一个类可以从其父类那里继承其属性。这一特点允许将知识组织成层次结构,从而它使知识的表示更结构化。系统采用类似于 Smalltalk 的继承规则:

①如果 A 类继承了 B 类,则 A 类的对象支持 B 类对象的所有操作。

②如果 A 类继承了 B 类,则 A 类的属性是 B 类属性的超集。

2. 方法

方法是属于对象的一种属性,用它来操作对象中的所有数据及传送或接收其它对象发来的消息,它也是面向对象知识表示的一种推理形式。在推理过程中,方法可以被调用而不能被修改。一般来说,方法又可分为三种形式:规则、内部方法和外部方法。

规则用来对属性变量求值;内部方法是以过程形式定义的知识;外部方法是系统提供或用户使用任何程序语言编写的过程,它们可以看成是对象的工具。

3. 消息传输及控制

系统控制可分为类内控制和类间控制。类内控制通过类自己定义的方法而管理。也就是说,类内控制依靠领域知识,它决定系统的微观行为。类间控制指通过通讯而完成的类间协调交互。消息传输(Message passing)是类间通讯的唯一方式,它决定系统的宏观行为。正因为消息传输是激活对象的唯一办法,所以决定了知识的模块性和封装性。

消息可由四个部分组成:发送消息的对象名称;接收消息的对象名称;选择消息的方法名称以及传送给某一方法的参数。

当一个消息创建后,先发送给消息管理模块检查其合法性。一般情况,消息管理模块对每一个消息作出四种可能裁决。第一,原始消息不加修改地直接传输给接收者;第二,消息内容及接收者可能被修改,而传给其它接收者;第三,消息可能受阻,不传给任何对象,这可能因为消息非法、接收者不存在,要求的方法没找到或者发送者权限不够等情况;第四,消息可能以广播方式发到某类的所有对象中。

七、结束语

作为本文的结束语,简单讨论一下面向对象这种知识表示方法的软件实现问题。许多面向对象的系统采用了 LISP 或 PROLOG 这类常用于人工智能程序设计的语言。虽然这些语言有许多优点,但笔者认为直接采用带 OOP 特性的语言更适合这类系统的编程。

Borland 公司的 C++ 提供了类的封装,继承及多态性这些强有力的 OOP 特性^[17],从而可作为首选的程序设计语言。

C++ 提供了一个关键字 CLASS 专门用于表示类,它可以封装类的属性(私有数据)和方法(成员函数)。要存取类中数据,必须通过类的成员函数调用,这完全符合面向对象知识表示的封装要求。

Borland C++ 提供了 Turbo Debugger 调试软件,它可以在源程序级调试程序,并特别配置了对类的调试以及对类的继承性调试。

我们使用 Borland C++ 2.0,对面向对象的知识表示方法进行了具体实现,在软件开发过程中,我们发现面向对象这种知识表示方法的封装性和模块性,给软件开发和维护带来极大的方便,因此面向对象的知识表示方法是一种很有价值的方法。

(参考文献共 17 篇略)