

专家系统开发环境 NEW 的设计与实现

陈世福 潘金贵 袁峰 童立

(南京大学计算机科学系, 南京 210008)

THE DESIGN AND IMPLEMENTATION ON THE NEW EXPERT SYSTEM DEVELOPMENT ENVIRONMENT

Che Shifu, Pan Jingui, Yuan feng and Tong Li

(Department of Computer Science, Nanjing University, Nanjing 210008)

Abstract Summarizing the current situation of knowledge engineering languages and their support environments. We designed and implemented the NEW expert system development environment according to the idea that knowledge engineering languages and their environments are the evolution of traditional programming. This paper describes its design and implementation, focusing upon the architecture, the knowledge representation, the inference mechanism, the environment and the applications.

摘要 本文分析了当前知识工程语言及其开发环境的现状,并介绍了基于知识工程语言及其支撑环境应是传统程序设计语言及其支撑环境的提高和进化的思想设计和实现的 NEW 专家系统开发环境,本文主要介绍了 NEW 的系统结构、知识表示、推理机制、开发环境以及实现与应用的情况。

§ 1. 引言

由于专家系统在知识表示和自动推理方面的特殊要求,用传统程序设计语言(如 Pascal, C, Fortran 等)实现起来较为复杂。因而,一类称为知识工程语言的新型语言应运而生,例如 KEE, M. 1, Guru, INSIGHT2, ART 等。仔细分析和实际使用后,我们发现目前的知识工程语言及其支撑环境存在许多不足:

(1) 用产生式系统开发的专家系统错综复杂,甚至连开发者也不能完全了解其内部推理过程。这表明产生式系统并不像人们早期所说的具有良好的模块性,完全有必要引入更高一级的模块结构。

(2) 产生式型知识工程语言,例如 INSIGHT2, EMYCIN 通常只提供布尔型、数值型、符号型三种基本数据类型,复杂的数据结构难以表示或无法表示。为了使知识工程语言有更大的适应性,有必要引入顺序、重复、递归等控制结构和复合数据结构及继承性体系。

(3) 知识工程语言通常无严格语法、语义定义,和外设交互能力差,计算能力弱,这就妨碍了开发高质量、高性能的实用专家系统。

(4) 开发环境不理想,对国内用户而言几乎所有知识工程语言只有英文的开发环境。而汉化版

本在版面、对汉语的处理能力、联机帮助提示信息方面有较多不足。

知识工程语言及其开发环境的这些不足表明,在设计知识工程语言时并没有很好地吸收和利用传统语言以及环境的设计实现方面积累的经验、教训。AI工作者们几乎都是先从具体应用系统中提炼出一个个新语言、新外壳系统,而不管其是否具有有良好的继承性,然后,再在不断产生的问题中发现数据结构,模块结构,形式定义,外设交互等有用特性。与此同时,实际用户在现有知识工程环境不能满足其应用要求时,又设计、实现一个个专用语言和环境。

从本质上说,专家系统是传统程序的发展。因此,知识工程语言应是传统程序设计语言的提高,知识工程环境应是传统语言支撑环境的进化。基于这种继承和发展的思想,我们认真分析了知识工程对语言和环境的新要求,吸收了传统程序设计语言设计和实现方面的有益经验,在需求和实现复杂性方面作了适当折中,设计并实现了NEW(Nanjing University Expert System Workbench)专家系统开发环境。本文将介绍NEW系统的主要技术细节。

§ 2. NEW 系统结构

NEW 专家系统开发环境提供了两种界面:开发者界面和使用者界面。开发者界面供专家系统开发者使用,用于设计专家系统。使用者界面供专家系统的用户使用,用于用专家系统进行咨询。

NEW 系统的结构如图 1 所示。

开发者界面主要由编辑程序和编译程序组成。NEW 编辑程序是一个功能较强的全屏幕编辑程序,可方便地建立、修改知识库源文件。建成的知识库文件由编译程序进行词法、语法检查,翻译成一种便于推理机使用的效率较高的内部表示。

使用者界面提供了推理、解释、跟踪调试、方法设置、浏览知识库等多种功能。

推理机可根据不同的知识表示方式,选择不同的推理策略,以适应问题的需要。

解释机制提供了 Why、How 等解释功能,给出的解释信息有利知识库的调试。跟踪调试控制在推理过程中是否显示整个推理轨迹和实例取值情况。

方法设置可改变可信度的组合方式,设置最低可信度阈值,在提问时是否自动提供菜单等。知识库浏览设施用于查看知识库中的概念、实例和方法说明。

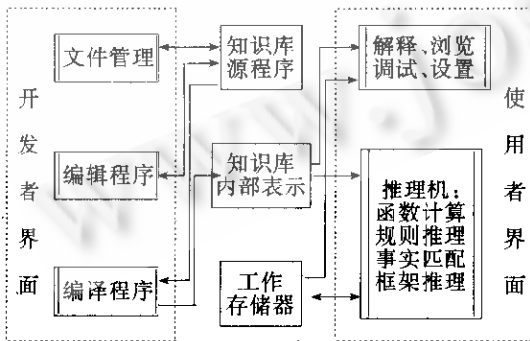


图1 NEW 的系统结构

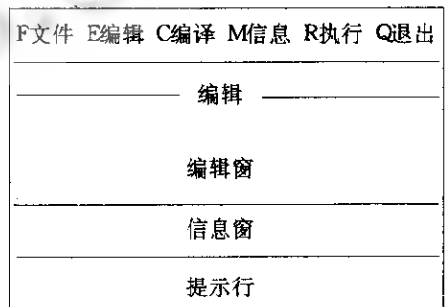


图2 NEW系统开发环境主菜单

上述界面均集成在一个基于多窗口的菜单驱动的环境之中,如图 2 所示,两个界面可以灵活地进行切换。显然,在此环境下,用户建立、修改、调试、执行自己的知识库是很方便的,用户还可通过键盘控制系统的运行。

§ 3. 知识库结构与数据表示

3.1 知识库结构

NEW 系统提供了一种用于开发知识库系统的知识工程语言,它具有多种知识表示形式,能描述并处理不同领域的问题.使用 NEW 构筑知识库系统的全部工作仅仅是建立特定问题领域的知识库.

NEW 系统的知识库由一组知识模块和有关的询问组成(如图 3 所示).每个知识模块是一个知识源,它们为概念、实例、函数和规则的说明.其 BNF 的定义为:

```

<知识库> ::= <知识模块> { ; <知识模块> } { ; <询问> }
<知识模块> ::= -knowledge base <模块名>;
                <说明> { ; <说明> }
                end
<模块名> ::= <标识符>
<说明> ::= <概念说明> | <方法说明> | <实例说明>
<询问> ::= -find <表达式> | classify <变量> | teach <概念名>

```

构筑好的知识库的推理从询问开始,NEW 提供了三种询问目标:find 用于根据知识库推导出其后表达式的值;classify 用于根据继承性体系求一实例所属的最小概念;teach 用于询问一个概念的定义.

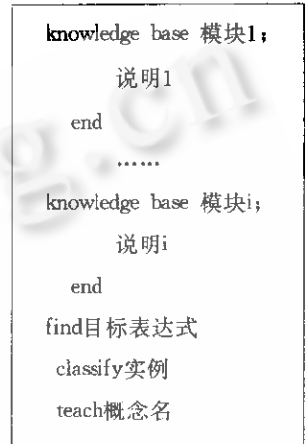


图3 NEW的知识库结构

3.2 数据表示

程序是对处理对象和处理规则的描述,而处理对象是通过语言中的数据结构刻划的.人工智能和知识工程领域的特点,对数据结构机制提出了新的要求,新的概念和方法.

传统语言中值为单一、确定的值,且要求变量在运行时刻具有某一值,引用未赋值的量常常会引起难以预料的结果.知识工程的研究表明:更多的实际问题需要表示值未定义,值带有一定的可信度以及取多个值等多种情况. NEW 知识工程语言中值定义为:

```

<值> ::= unknown | <表达式> <可信度> { and <表达式> <可信度> }
<可信度> ::= [cf<整数>]

```

其中,unknown 表示未知值,<可信度>用 0 至 100 的整数表示,and 用于联接值的多种可能.

在计算机语言中,值通常被划分为多种类型,以提高程序的可读性、可靠性和效率.

NEW 中的类型称为概念(concept).概念有标准概念(integer、real、boolean、char、string)、枚举概念、集合、序列或框架.其中标准概念、枚举概念、集合、序列对应于传统程序设计语言中的相应类型,框架概念对应于记录类型,但扩充了值的继承性.框架可以嵌套,子框架可以继承父框架概念的属性名、活动值、缺省值和限制条件.与传统语言中类型定义不同,NEW 中概念定义为名、值集、处理子程序、父概念、活动值、缺省值、限制条件和提问八个要素.其中,父概念表示继承性,活动值用于表示求值方法,缺省值表示无法用通常方法求值时所用的值,限制条件用于排除一些不合理的取值,即限制值集,提问用于向用户提示输入变量的值.

例如,可将 employee 定义为 person 的子概念,其限制条件是年龄必须在 18 岁至 65 岁之间:

```

concept employee is a person with
    age: where (age ≥ 18) and (age ≤ 65)

```

end.

定义了概念,即可定义相应的实例.概念和实例之间的关系类似于类型和变量之间的关系.实例可作为全局量、局部量和参数.实例可说明实例名、类型、计算值的方法(活动值)、缺省值、限制条件和提问,例如:

```
birth_date:string ask"what's your birth date?"
```

```
present_date:string default"1/1/1990".
```

```
ask"what's the date today?"
```

```
age:integer=diffdate(present_date,birth_date)
```

定义了三个可用于计算年龄的实例:birth_date 为出生日期,可通过提问" What's birth date?" 向用户询问;present_date 为当前日期,可通过提问"what's the date today?"向用户询问,当用户不提供时,缺省值为 1/1/1990;age 为年龄,它通过调用函数 diffdate 计算年龄.

通过概念和实例的定义,NEW 已能对知识库中的知识进行一定程度的推理.这些推理能力主要由框架概念提供.

§ 4. NEW 的知识表示及其处理

数据必须经过适当的控制结构处理才能转换成有意义的结果,传统语言通常用过程(函数)和子程序表示处理过程.

在知识工程语言中,虽然过程和函数在表示各种算法方面仍有很大的用途,但表示经验型的规则知识就显得不方便. NEW 中采用了多种方法表示数据的处理知识,对象间的简单关系可用表达式和事实表示,较复杂的算法型知识可用函数表示,经验知识可用规则表示.完成某一特定任务的处理知识的整体,在 NEW 中称为方法(method),一个方法有其名字,输入输出参数说明和方法体.方法可分为函数、规则和事实三类. NEW 系统的知识表示具有一定的特色,下面将分别予以介绍.

4.1 事实与表达式

事实和表达式可以用来表示对象间简单的关系.例如,用 NEW 开发的酒类选择专家系统中有关选择方法的事实描述为:

```
fact c; winecolor; b; winebody; s; winesweet → winelookup; string;
```

```
winelookup(red, light, _) = "valpolicalla";
```

```
winelookup(red, _, dry) = "cabernet sauvignon";
```

```
winelookup(white, full, dry) = "chardonnay";
```

```
.....
```

```
end;
```

4.2 规则

规则易于表示问题领域的经验性知识,与其它系统的规则所不同的是, NEW 系统采用分类说明方式,将求同一表达式值的规则集中在一个规则集中说明,即呈如下形式:

```
rule <接口>;
```

```
<规则>;
```

```
<规则>;
```

```
.....
```

```
end;
```

而其内的各条<规则>与其它系统中的规则格式类似.例如,动物分类系统中的判定规则可表示为:

```
rule → 该动物是: string;
```

```

if 哺乳动物 and 食肉动物 and 是黄褐色的 and 有暗斑点
then"豹";
if 哺乳动物 and 食肉动物 and 是黄褐色的 and 有黑色条纹
then"虎";
.....
end;
```

这条规则说明把对“该动物是”求值的多条规则集中起来用一条规则说明来表示,其中〈接口〉部分一般是关于一组规则(或事实)的参数及其取值类型的一个说明,它的引入对规则(或事实)进行语法检查提供了方便。

4.3 函数

NEW 中引入函数概念可用于表示较复杂的算法型知识。NEW 中的函数可以是在知识库外部用其它语言(例如 C 语言)定义的,也可以是用 NEW 知识工程语言在知识库中定义的。函数说明中提供了顺序、选择和递归三种主要控制结构,因而具有很强的定义能力,可以方便地进行各种较复杂的数学计算与表处理。例如阶乘函数可说明成:

```

function n;integer→fac;integer;
    if n=0 then 1
    else let X;integer=fac(n-1)
        in X * n
end;
```

事实说明、规则说明和函数说明在 NEW 系统中又统称为方法说明,用于说明处理知识的方法,类似于传统语言中的子程序,这使得习惯于传统程序设计语言的程序员在使用 NEW 知识工程语言开发专家系统时感到自然和方便。

然而,事实、规则和函数三类方法说明本质上定义了一个函数式语言,这对于完成外设交互等任务是不够的。为了能够引用传统语言的多种成份,NEW 提供了一个和传统语言 C 的接口,使得用 NEW 编写的知识库可充分利用 C 语言的强大功能和极高的效率。这一接口是通过所谓的外部函数提供的。当函数的体说明为 external 时,相应的体可用 C 语言实现。在咨询时它和 NEW 系统合作完成推理任务。

4.4 框架

框架是一种表达概念及其相互关系的结构型知识表示方法,其最大优点是结构性强,层次分明,可以融描述性知识、过程性控制知识及缺省值于一体,便于将大问题化小,分而治之。

NEW 系统中的框架由继承性链、属性及其侧面值构成,其侧面包括类型、活动值、缺省值、约束条件。

使用框架概念,可以描述复杂的、大的或动态的知识,而这一点正是规则型系统所不易表示的。

§ 5. NEW 的推理机制及主要算法

NEW 系统具有多种推理策略,以适应各种不同的知识表示的推理和用户的需要。

1. 对规则通常进行反向推理,但用户可方便地用 Let 语句引入临时变量控制求解次序,使推理过程为正向推理;
2. 对事实通过匹配进行推理;
3. 对函数计算求出(递归)表达式的值,函数计算的主要算法如下:

```

value eval(expr e)
switch (class(e)) /* 对表达式 e 分类 */
{case call:op=get_op(e); /* call 类,取保留字 OP */
  switch (class(op)) /* 判断 OP: */
  {case func:rslt=eval(body(op));break; /* 是 function,计算函数体 */
  case fact:rslt=fact_infer (body(op),para(e));break; /* 是 fact,进行事实推理 */
  case rule:rslt=rule_infer (body(op));break; /* 是 rule,进行规则推理 */
  }
  return rslt;
case dot:return frame_infer(e); /* dot 类,进行框架推理 */
case ass:.....
case sub:.....
case if:.....
.....
}

```

4. 对框架系统可提供:

(1)活动值推理:设概念 C , 有属性 A , 其活动值为 V , 若说明 I 为 C 的实例, 则可得出 $I.A=V$.

(2)缺省值推理:设概念 C , 有属性 A , 其缺省值为 D , 若说明 I 为 C 的实例, 则在无其它信息的情况下, 可得出 $I.A=D$.

(3)错误检查:设概念 C , 带有属性 A , 其限制条件为 R , 若说明 I 为 C 的实例, 并将 V 赋给 $I.A$, 则 $I.A$ 必须满足 R , 否则系统将给出适当的出错信息.

(4)匹配分类:设一组概念 C_i 构成一以 C_0 为根的概念体系, 若说明 I 为 C_0 的实例, 则 NEW 系统将能根据 I 的各属性值推出 I 为某一 C_i 的实例, 且不存在 C_k 为 C_i 的子概念, 使 I 为 C_k 的实例. 例如, 动物分类体系以 animal 为根概念, 若说明 tiger:animal, 系统能推出 tiger 为 vertebrate 的实例, 为 mammal 的实例, 直至无法细分. 可见这种自动匹配分类推理功能十分有利于分类型专家系统的编制.

框架推理的主要推理算法表示如下:

```

value frame_infer(expr e)
if(getval(e) != null) /* 计算表达式的值 */
then {attr=lastattr(e); rslt=null; /* 为空,取最后一个属性 */
  if active(attr)! = null then rslt=eval(active(attr)); /* 活动值不为空,则计算此值 */
  if(rslt == null) then rslt=ask_user(e); /* 询问用户 */
  if(rslt == null) then rslt =eval (default(e)); /* 用户未提供,则使用缺省值 */
  if(! constraint(e,rslt)) then {error("constraint fail"), rslt=null}; /* 限制条件检查 */
  return rslt;
}
else return getval(e);

```

§ 6. 结 语

整个 NEW 系统采用 C 语言编程, 源程序约 10000 行, 可在各档 IBM-PC 系列机和兼容机上运行, 同时提供了英文版本和 CCDOS、UCDOS 等多种中文版本. 和通常汉化版本不同的是, NEW 的多个中文版和英文版完全兼容, 且汉字不仅可以作为字符、字符串出现, 还可以作为各种标识符出现. 基于 SUN 工作站的版本已移植成功.

NEW 的知识工程语言提供了事实、规则、函数、框架等多种知识表示形式, 以及丰富的数据结构和控制结构, 表达能力强, 能方便地表示复合数据结构, 描述和处理不同领域的问题类型, 且具有

良好的模块性. NEW 的概念和框架概念以及方法稍加改进,即可方便地表示对象(object),从而可方便地扩充为面向对象的知识工程语言. NEW 系统的推理机制灵活,提供了较强的函数计算,规则推理,框架推理,事实匹配,可信度计算,提问菜单自动生成能力和多种推理策略,用户可按不同的知识表示方法选择相应的推理方式,并具有传统程序设计语言的风格,NEW 的开发环境和用户接口友善,与 C 语言的接口自然,与外设交互灵活. 这些功能特性与同类系统相比具有一定的特色.

NEW 系统实现以后,我们用该系统开发“动物分类专家系统”、“选酒专家系统”和“微机故障诊断专家系统”等实例,同时在国内进行推广应用. 目前国内已有 6 个用户在使用该系统. 用它开发了“紫外—可见光谱定性分析专家系统”、“水稻害虫决策管理专家系统”和“植物害虫鉴别专家系统”等实用专家系统. 为用户提供了一个功能较强的知识工程语言,得到用户好评.

参考文献

- [1] D. Waterman, A Guide to Expert System, Addison—wesley, New York, 1985.
- [2] F. Hayes—Roth, Eds. , Building Expert Systems; Addison—welsey Publishing Company, 1984.
- [3] Yuan Feng, Chen Shifu, On the Design of the NEW Expert System Workbench, Proc of the 3rd Pan Pacific Computer Conference, 1989. 8.
- [4] 陈世福、潘金贵等,《知识工程语言与应用》,南京大学出版社,1988. 2.
- [5] 刘大有等,《知识工程原理》,吉林大学出版社,吉林,1989.
- [6] 史忠植,《知识工程》,清华大学出版社,北京,1987.
- [7] 何志钧、俞瑞钊等,专家系统工具剖析,《计算机研究与发展》,1989. 9.