

◎ 研发、设计、测试 ◎

结合 CLIPS 和 VC++ 实现规则推理的方法

代 荣^{1,2}, 何玉林¹, 杨显刚¹DAI Rong^{1,2}, HE Yu-lin¹, YANG Xian-gang¹

1. 重庆大学 机械工程学院, 重庆 400044

2. 西南大学 工程技术学院, 重庆 400715

1. College of Mechanical Engineering, Chongqing University, Chongqing 400044, China

2. College of Engineering Technology, Southwest University, Chongqing 400715, China

E-mail: dai_yun2002@sina.com

DAI Rong, HE Yu-lin, YANG Xian-gang. Method to realize rule-based reasoning using CLIPS and VC++. Computer Engineering and Applications, 2010, 46(3): 54-57.

Abstract: Expert system development generally has three ways, such as using advanced programming language, using expert system crust, and using expert system development kit. This article introduces respective characteristic of expert system tool CLIPS6 and advanced programming language VC++, puts forward that expert system tool CLIPS6 and VC++ advanced program language are combined together to develop the expert system for rule reasoning. The paper describes in detail the general process of CLIPS embedded into VC++, how to embed CLIPS into VC++, how to add CLIPS user definition function and transfer and return parameters, how to add CLIPS event handlers. And take the motorcycle design as example, elaborates the rule editing, facts obtaining, rule explaining, man-machine interactive function, thus realize rule-based reasoning of the motorcycle intelligence design.

Key words: motorcycle; intelligence design; rule reasoning

摘要: 专家系统开发通常有三种方式: 高级程序语言, 专家系统外壳, 专家系统工具。该文介绍了专家系统工具 CLIPS6 和高级程序语言 VC++6.0 的各自特点, 提出了将专家系统工具 CLIPS6 与 VC++6.0 高级语言编程工具结合起来开发专家系统, 实现规则推理。详细描述了 CLIPS 嵌入 VC++ 的一般过程: 如何把 CLIPS6 嵌入 VC++, 如何加入 CLIPS 用户自定义函数来传递和返回参数。并以摩托车智能设计为例, 详细阐述了规则编辑, 事实获取, 实现规则解释, 实现人机交互功能等, 从而实现摩托车智能设计的规则推理。

关键词: 摩托车; 智能设计; 规则推理

DOI: 10.3778/j.issn.1002-8331.2010.03.017 文章编号: 1002-8331(2010)03-0054-04 文献标识码: A 中图分类号: TP182; TP302.1

1 引言

专家系统的开发一直是人工智能领域的一个重要研究方向。目前比较流行的专家系统开发语言主要有以下几种工具: (1) 程序设计语言。包括通用程序设计语言(如 C/C++, PASCAL)和人工智能语言(Prolog 和 Lisp)两种。(2) 专家系统外壳。它是由一些已经成熟的具体专家系统通过抽取其中的具体知识, 而保留它的体系结构和功能演变过来的。利用它可以快速产生一个专家系统。其主要缺点在于灵活性和通用性比较差, 只局限于某些特定的域。(3) 专家系统开发工具。它介于两者之间, 为知识的表示提供了固定的模式, 使专家系统的构造更加方便。

这三种方式各有优缺点, 其中专家系统工具以其高效、简便成为最常用的工具; 另一方面, VC++ 等面向对象的可视化编程工具早已成为科研人员熟悉的开发工具。选择两者结合的方法来开发专家系统, 可以充分利用两者的特点实现优势互补。

2 CLIPS 简介

CLIPS 基本结构是产生式系统, 采用正向推理机制, CLIPS 与一般的产生式系统的不同在于其推理过程中独特的 *rete*^[1] 模式匹配算法, 极大地提高了系统的反应速度。CLIPS 专家系统主要有三大部分组成: (1) 知识库: 包括初始事实和初始对象实例以及规则库。(2) 推理机构: 控制整个规则的执行, 决定规则是否应该执行和什么时候被执行。(3) 黑板: 由事实列表和对象实例列表组成, 用于存储推理结果数据(黑板数据)。黑板数据包括开始推理以来得到的事实集对象实例集。

3 CLIPS 和 VC++ 结合的优势

利用 VC++ 实现专家系统的主要缺点是 C++ 在逻辑推理方面并不擅长, 因此采用 C++ 编写的推理机的开发程序量相对较大。CLIPS 的优点包括: (1) 逻辑推理方面的强大功能。使用

作者简介: 代荣(1971-), 男, 博士研究生, 讲师, 高级程序员, 主要从事机械设计自动化、智能化、知识工程研究; 何玉林(1945-), 男, 教授, 博士生导师, 主要研究方向: CIMS, CAD/CAM, 计算机图形学等; 杨显刚(1975-), 男, 教师, 博士研究生, 研究方向: 知识工程。

收稿日期: 2008-12-15 修回日期: 2009-01-16

CLIPS 开发专家系统,推理机的设计可以大大简化。只需要将相关领域的知识按照给定的知识表示模型表示并按一定结构顺序组织起来即可。(2)有利于和其他语言联合使用。CLIPS 很容易实现和其他语言的接口,而且 CLIPS 写成的专家系统可以和 VC++共同编译,极大地提高了系统运行速度和可移植性。

从以上分析可以看出,VC++和 CLIPS 在开发专家系统方面都有各自突出的优势,也有其缺陷。两者具有很强的互补性:CLIPS 在逻辑推理方面的强大功能使它适合编写专家系统的知识库和推理机部分;VC++用于开发人机界面、处理与数据库和外设的数据交换。CLIPS 在推理过程中遇到的复杂数值运算则通过调用 VC++编写的外部函数来完成。这样结合 VC++和 CLIPS 编程可以实现功能强大、界面友好的专家系统。

4 CLIP 嵌入 VC++的一般过程

(1)首先将 CLIPS 的源代码拷贝到 VC++工程文件目录下。

(2)在 VC++中选取菜单:工程->添加工程->Files,将所有文件(*.C)都选上添加到工程中。

(3)加入相应头文件声明

为使在 VC++中能识别 CLIPS 源程序,并可以在用户自定义函数中使用 CLIPS 中的方法函数,必须加入 clips.h 的头文件声明,声明代码如下:

```
extern "C" {#include "CLIPSSrc\clips.h"}
```

声明可在 stdafx.h 头文件末尾加入,但在菜单工程->设置->C/C++标签下,将/yu stdafx 去掉或者在分类中将 precompiled header 选取为不预处理头文件。

(4)在 UserFunctions 函数中声明用户自定义函数

用户自定义函数必须对 CLIPS 声明,以使它们能被 CLIPS 正常访问。用户自定义函数可以通过修改函数 UserFunctions 来向 CLIPS 声明。在 UserFunctions 中,需为每个用户自定义函数调用函数 DefineFunction,这样用户的源代码才能被 CLIPS 编译和链接。例如:

extern "C" void UserFunctions();//必须在用户自定义函数的定义前引用,这样 UserFunctions 中声明的函数才能找到在外部定义的用户自定义函数。

```
void UserFunctions()
{ extern void TextOut(char*str1,char*str2);
  extern int YesOrNo(char*str1,char*str2,char*str3);
  extern void ExplainOut(char*str1,char*str2);
  DefineFunction("TextOut",'v',PTIF TextOut,"TextOut");
  DefineFunction("YesOrNo",'b',PTIF YesOrNo,"YesOrNo");
  DefineFunction("ExplainOut",'v',PTIF ExplainOut,"Explain-
  Out");
}
```

DefineFunction 的第 1 个参数是函数的名字,是一个字符串,第 2 个参数是返回 CLIPS 的值的类型。返回类型如表 1 所示^[2]:

表 1 返回类型

返回代码	返回类型	返回代码	返回类型
b	布尔型	l	长整型
c	字符串型	v	无返回值
f	单精度浮点数	d	双精度浮点数
i	整型	a	外部地址
o	实例名	w	符号

第 3 个参数是指向实际函数的指针,它与第一个参数不一

定相同,第 4 个参数是第 3 个参数的字符串表现形式,它必须与第 3 个参数相同,但得加上引号。

(5)编写用户自定义函数

在要推理的窗口中编写用户自定义函数,例如:

```
void TextOut(char*str1,char*str2)
{ str1=RtnLexeme(1);
  str2=RtnLexeme(2);
  if(strcmp(str1,"REPAIR")==0)
  {
    CCaseEvaluate CaseEvaluate;
    CaseEvaluate.m_strAdvise=str2;
    CaseEvaluate.DoModal();
  }
  else if(strcmp(str1,"STROKE")==0)
  {
    MyRepltem.m_strStroke="";
    MyRepltem.m_strStroke+=str2;
  }
  else if(strcmp(str1,"cylinder-number")==0)
  {
    MyRepltem.m_strCylinderNum="";
    MyRepltem.m_strCylinderNum+=str2;
  }
  ...
}
```

函数 RtnLexeme 用于返回符号、字符串、实例名的字符型指针^[2]。其原型为 cha*RtnLexeme(argumentpositon),函数的参数需一次给出一个,每次给出参数在函数中的位置。

(6)启动 CLIPS

在 VC++中启动 CLIPS,需要在窗口中激发一按钮,在初始化 CLIPS 环境后,将规则库与事实库载入,再运行 CLIPS,进行推理,其流程见具体实现代码如下:

```
void OnRepair()
{ char*RuleName,*pFileName;
  RuleName="E:/MTCZNSJPT/MCDES/reaason/rule1.clp";//规则库
  pFileName="E:/MTCZNSJPT/MCDES/reaason/facts.txt";//事实库
  InitializeCLIPS();//初始化 CLIPS 环境
  Load(RuleName);//载入规则库
  Reset();//使 CLIPS 复位
  LoadFacts(pFileName);//载入事实库
  Run(-1L);//启动 CLIPS
}
```

5 建立专家系统

以摩托车设计专家系统(即知识库管理系统)为例说明如何使用 CLIPS 嵌入 VC++的方法实现规则推理。摩托车设计专家系统主要用于解决摩托车总体方案设计的选型与布置方面的问题。其基本框架如图 1 所示。它包含了知识获取、知识维护、知识查询和总体设计推理系统模块,其中总体设计推理模块的结构如下,如图 2 所示。

总体设计推理则采用了实例推理与规则推理的集成推理模式。根据设计需求主要特征提取相似实例并排序,选择一个实例,从而得到初始设计方案,按照方案评估确定是否满足要求,如果部分属性不满足,则对部分属性进行规则推理修改,再

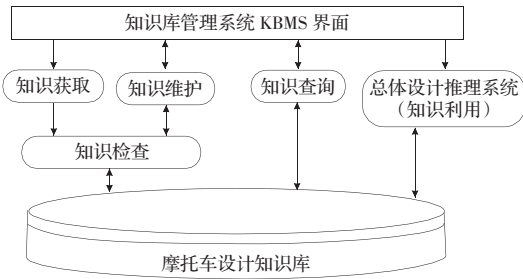


图1 摩托车设计专家系统基本框架图

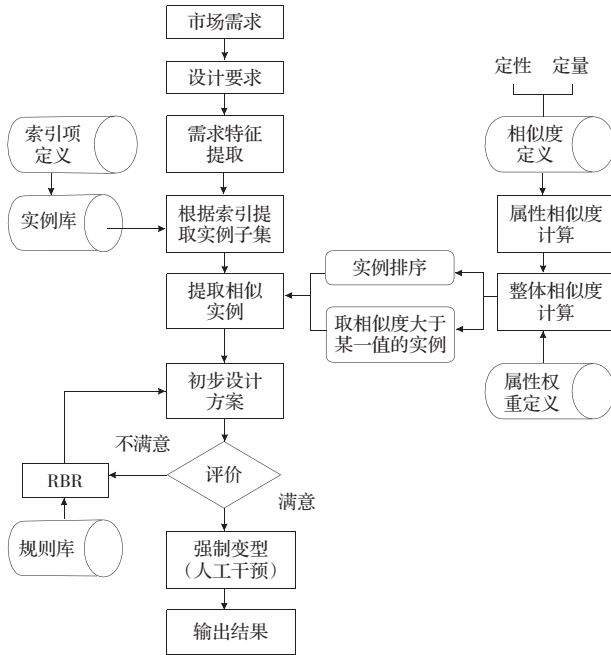


图2 CBR和RBR的集成推理图

评价,得到满意结果后输出最终设计方案。

由此可知规则推理是在实例推理得到初步设计方案之后,其关键需要解决规则编写、事实获取、解释实现和人机交互的实现等。

5.1 规则编写

规则编写按照 CLIPS 中的规范,根据领域知识的特点进行编写^[4-5]。规则库存放于名为 rule.clp 的文件中,在启动 CLIPS 时将其载入 CLIPS。如要与 VC++ 进行数据交流。则在规则中需用用户自定义函数。如下一条规则:

```
(defrule cooling-select1 "选择发动机的冷却方式"
  (repair engine)
  ?d<-(design-requirement(exhaust-capacity ?exhaust-capacity))
  (test(<=?exhaust-capacity 125))
=>
```

(ExplainOut "cooling-select1" "风冷构造简单,不存在泄漏、积垢、沸腾、结冰等问题,冷却系工作可靠,使用维修方便,特别是自然风冷不需要维修。但随着发动机功率不断升高,风冷常显得冷却不足,风冷发动机温度变化大,热负荷高,影响了可靠性和耐久性,运转噪声也较大。125 mL 以下的发动机大都采用风冷")

```
(if (YesOrNo "选择发动机冷却方式" "风冷" "水冷")
then
(TextOut cooling "风冷")
else
(TextOut cooling "水冷"))
```

这是一个选择发动机冷却方式的规则,当发动机排量不大于 125 时,可选用风冷或水冷。上面用到了 3 个用户自定义函数: ExplainOut 用于规则解释及与用户交互时给用户提示, YesOrNo 用于与用户交互, TextOut 将推理结果传送给 VC++。

5.2 事实获取

从 VC 应用程序界面录入数据,然后按照规则库里的模板结构,将数据按照一定格式存放到事实文件 fact.txt 里。

```
void OnRepair() {pFileName="E://MTCZNSJPT//MCDESS//reason//
facts.txt"; //调出一个写字板
if (! MyFile1.Open (pFileName, CFile::modeCreate | CFile::mode
eReadWrite | CFile::typeText)
{
MessageBox("Unable to open file! ");
}
strOutputFormat("design-requirement(moto-use \"%s\")(moto-type \"%
s\")(exhaust-capacity %f)(max-velocity %f)(max-power %f)(hold-
pace %f)(huihuangR %f)(max-torque %f)", MyReqItem.m_strUse-
Name, MyReqItem.m_strFrameName, MyReqItem.m_fCapacity, MyReqItem.
m_fSpeed, MyReqItem.m_fPower, MyReqItem.m_fHoldPace, MyReqItem.
m_fHuiZhuang, MyReqItem.m_fTorque);
MyFile1.WriteString(strOutput);
MyFile1.Close();
}
```

5.3 实现解释

解释通过记录推理的路径实现。首先用自定义函数 ExplainOut 将每条规则的规则号与解释文字传送给 VC++, 然后定义两个全局数组来分别记录推理中所用到的规则号与解释。数组定义如下:

```
Char*strExpalin[100];
Char*strRuleNo[100];
void ExplainOut(char *str1,char *str2)
{
str1=RtnLexeme(1);
str2 =RtnLexeme(2);
int n=MyRepItem.i;
MyRepItem.strExpalin[n]=str2;
MyRepItem.strRuleNo[n]=str1;
MyRepItem.i++;
}
```

MyRepItem 是封装结果数据的对象,被定义为全局对象变量。

5.4 实现人机交互

良好的人机交互是系统性能优越的体现。在推理过程中,通过人的参与,可以进一步明确问题的性质,解决规则之间的冲突。有些规则可以有多个结果,系统给出结果的评价和建议,由用户根据实际情况去选择。人机交互主要由用户自定义函数来实现,其定义如下:

```
int YesOrNo(char*str1, char*str2, char*str3)
{
CRepairUser RepairUserDlg;
str1=RtnLexeme(1);
str2=RtnLexeme(2);
str3=RtnLexeme(3);
RepairUserDlg.m_strShow1=str1;
RepairUserDlg.m_strShow2=str2;
```

表 1 实例推理结果

用途	车型	排量/mL	最高车速 ($\text{km}\cdot\text{h}^{-1}$)	最大功 率/kW	最大转矩 ($\text{N}\cdot\text{m}$)	经济车速 ($\text{km}\cdot\text{h}^{-1}$)	最低稳定车 速($\text{km}\cdot\text{h}^{-1}$)	爬坡能力 ($^{\circ}$)	最小回转 半径/m	整车成本 /元	整体相 似度值
设计要求	轻便型	坐式	45	45	4	6	32	5	7	1.7	0
相似实例	轻便型	坐式	49	60	4.12	6.66	0	0	1.8	14 502	0.868 718
默认权值			0.38	0.15	0.2	0.05	0.05	0.05	0.05	0.05	

```

RepairUserDlg.m_strShow3=str3;
RepairUserDlg.DoModal();
if(RepairUserDlg.selectValue==1)
{return(TRUE);}
else
{return(FALSE);}
}

```

5.5 运行实例

由于摩托车智能设计系统包括了专家系统,参数化建模,有限元分析模块,该文只是编译、运行单机上的专家系统模块,点击树形目录中第一行总体设计推理系统(其他是知识库中知识获取,查询,维护,数据库维护),输入用户需求如下,如图3所示。整车成本:0(为0表示没有输入)。得到摩托车整体相似度计算结果,从大到小排序,可以任选1到9中任一方案,选中方案1,得到实例推理结果如表1。相似实例中有的数据项数据为0,表示数据库搜索到的实例中该字段缺少或未输入该数据,从上面数据可看出,所提取的相似实例与设计要求相似。



图3 专家系统界面图

相似实例中的排量、功率、转矩等属性不相符,对不相符属性进行规则推理,将激发上一章中的规则,并弹出人机交互窗口,见图4。

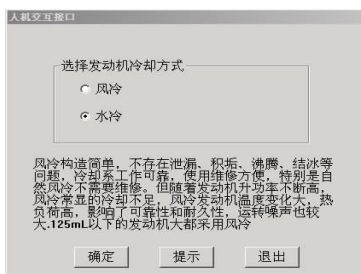


图4 人机交互界面

选择水冷,点击确定,得到规则推理后的最终方案,如图5。

用户如需要了解规则推理路径,则可查看规则推理解释器,系统将输出相关的提示信息给用户参考(如刚才在第4章的规则解释与图6中相同),见图6。

从实例推理的初始方案中,发动机冷却方式为四冲程、风

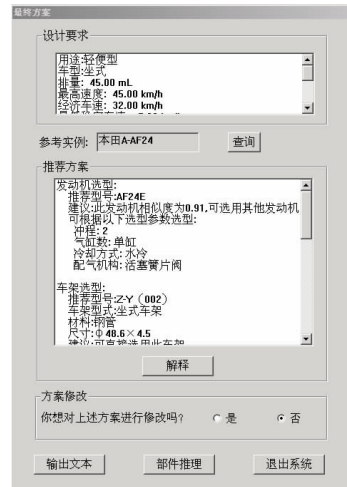


图5 规则推理后的最终方案

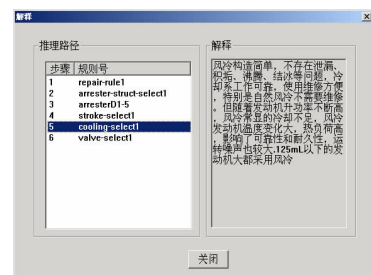


图6 规则追踪与解释

冷,到图4人机交互界面,选择水冷,到图5最终方案里出现二冲程、水冷,以及推理路径解释,说明了程序运行的有效性。

6 总结

采用专家系统工具 CLIPS 嵌入 VC++实现规则推理的方法,不仅可以有效利用知识库中的规则知识,实现基于规则的推理,还可以构建出友好的人机交互界面,节约大量开发推理机程序的时间。实践证明,该方法可节约大量的时间和人力,提高开发效率。

参考文献:

- [1] Giarratano J, Riley G. 专家系统原理与编程[M]. 印鉴,译.北京:机械工业出版社,2000.
- [2] 叶雪梅,付光远.用 C++ builder 4 调用专家系统工具 CLIPS 6 实现故障诊断[J].计算机应用软件,2000(10):50-55.
- [3] STB of NASA. Advance programming guide[Z]//Clips6.21 reference manual I.From Intelnet,2003.
- [4] STB of NASA. Basic programming guide[Z]//Clips6.21 reference manual I.From intelnet,2003.
- [5] 吴鹤龄. 专家系统工具 CLIPS 及其应用[M].北京:北京理工大学出版社,1991.