

# CLIPS 数值应用与 VC++ 的交互

宁志强, 陶元芳, 刘晓莲

(太原科技大学 机械学院, 山西 太原 030024)

**摘要:** CLIPS (C Language Integrated Production System) 是一种专家系统开发语言。因为其界面基于 DOS 操作系统, 所以不够友好。文中提出了一种改进的 VC++ 和 CLIPS 交互方法。采用直接从 CLIPS 动态链接库中调用内部函数的方法来读取保存在事实中的槽值。这种方法可以不必通过文本交互中介就把 CLIPS 中的多个事实同时输出到 MFC 编辑框中。文中介绍了 CLIPS 在选用数值型机械零部件方面的应用。分析了应用 CLIPS 高效的模式匹配和隶属函数理论来选取零部件的方法。通过举一个实例来说明直接调用 CLIPS 内部函数的具体实现步骤。

**关键词:** CLIPS; 专家系统; 动态链接库; 机械零部件; 内部函数

中图分类号: TP311

文献标识码: A

文章编号: 1673-629X(2013)04-0226-04

doi: 10.3969/j.issn.1673-629X.2013.04.055

## Numerical Application of CLIPS and Interoperation between CLIPS and VC++

NING Zhi-qiang, TAO Yuan-fang, LIU Xiao-lian

(College of Mechanical Engineering, Taiyuan University of Science and Technology, Taiyuan 030024, China)

**Abstract:** CLIPS (C Language Integrated Production System) is a kind of expert system language. The user interface is not friendly because it is based on the DOS operating system. It presents a method of improved interoperation technology of VC++ and CLIPS. And employ the method of directly calling the internal function in dynamic link library of CLIPS for getting slots of facts. The method can directly output some facts in CLIPS to MFC edit box instead of text file as medium. Introduce how to apply CLIPS for selecting numerical machinery parts. The meaning of using CLIPS efficient pattern matching and membership function theory to select numerical parts was analyzed. An example is given in the paper to illustrate steps of the interoperation through directly calling the internal function of CLIPS.

**Key words:** CLIPS; expert system; dynamic link library; machinery parts; internal function

## 0 引言

CLIPS 是一种专家系统开发语言, 多应用于符号推理, 如故障检测专家系统、操作型专家系统、目标识别、方案选择等。在数值方面应用较少, 笔者尝试利用 CLIPS 高效的模式匹配和隶属函数理论来选取零部件。采用直接从 CLIPS 动态链接库中调用内部函数的方法实现与 VC++ 的交互, 并介绍了部分 CLIPS 的内部函数。

## 1 CLIPS 介绍及在机械零部件方面的应用

### 1.1 专家系统工具 CLIPS

专家系统开发工具 CLIPS 是“(C Language Inte-

grated Production System)”的首字母缩写词。它是美国航空航天/约翰逊太空中心用 C 语言设计的, 其主要特点是可移植性高, 成本低和易于与外部系统集成。它的源代码是公开的, 免费的, 允许任何人修改和使用<sup>[1]</sup>。

CLIPS 中知识表示的技术的一个常用方法是产生式规则。如果在事实库中一个规则的特定前件事实存在, 就满足了该规则的模式匹配条件, 然后执行后件命令。

CLIPS 基于规则的专家系统的基本组成<sup>[2]</sup>:

- 1) 事实库: 包含推理所需的数据。
- 2) 知识库: 包含所有规则。
- 3) 推理机: 对运行进行总体控制。

### 1.2 CLIPS 在机械零部件选取时的数值型应用

选取一个部件时, 采用隶属度值来衡量该部件和实际参数要求的匹配程度。模糊命题“X 近似  $\gamma$ ”的隶属函数:

收稿日期: 2012-07-19; 修回日期: 2012-10-24

基金项目: 山西省研究生教改课题(20102034)

作者简介: 宁志强(1986-), 男, 硕士研究生, 主要研究方向为起重运输机械及其 CAD 技术; 陶元芳, 教授, 研究方向为起重运输机械及其 CAD 技术。

$$\mu_{\text{close}}(x) = \frac{1}{1 + \left(\frac{x - \gamma}{\beta}\right)^2}$$

如图1所示,该隶属函数可以表达所有近似于某个特定值 $\gamma$ 的数, $\beta$ 参数对应着交叉点0.5,由用户根据自己经验定义。当存在多个评价指标时,用权系数来描述各个方面重要程度。

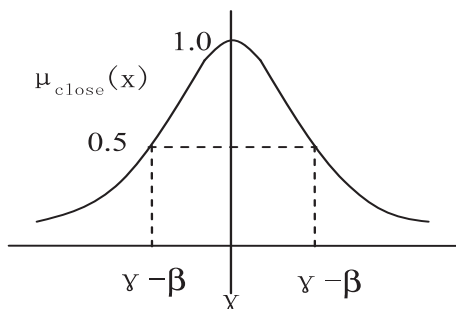


图1 模糊命题“ $X$ 近似于 $\gamma$ ”的隶属函数

### 1.3 CLIPS 和 VC++的交互方法

CLIPS 界面并不友好,是基于 DOS 操作系统,无法满足工程计算时以编辑框和对话框实现数值输入和输出的要求。而 VC++ 中 MFC 界面比较适合工程应用。此外 CLIPS 本身并不是为数据处理设计的,在这一点上 VC++ 更具优势。把 VC++ 和 CLIPS 融合可以优势互补<sup>[3]</sup>。

CLIPS 是用 C 语言编写的,可以方便地嵌入到 VC++6.0 平台,在实际运用中,常常使用以下几种方式:直接嵌入方式、加载动态链接库方式、包装类方式等<sup>[4]</sup>。

(1) 直接嵌入式<sup>[5]</sup>:直接把由 C 语言编写的 CLIPS 源代码嵌入到 MFC 中,这种方法较困难,因为 CLIPS 源代码很大,由几十个 C 文件组成,有些 C 文件会被重复编译,直接点击不能生成 exe 文件,要通过附带的 MAKEFILE 来进行编译。

(2) 加载动态链接库方式<sup>[6]</sup>:把 C 语言编写的 CLIPS 源代码做成动态链接库,然后链接到 VC++6.0 的 MFC,在 MFC 中调用 CLIPS 的底层函数。现在网上有很多 CLIPS 源程序动态链接库可供下载。

(3) 包装类式:网上现有的包装类 clipswrap 把 CLIPS.dll 的所有函数封装起来<sup>[3]</sup>,有了 clipswrap 包装类,就可以方便地在 VC++6.0 主程序中嵌入 CLIPS 作为专家系统的内核<sup>[7]</sup>,而且整个程序也更符合面向对象的程序设计风格。采用包装类式时要从网上下载 clipswrap,往往包装类必须要注意版本问题,而极少能够找到匹配的版本。

### 1.4 针对 CLIPS 数值型应用时与 VC++的交互方法

文中采用了加载动态链接库并直接调用底层函数的方法。数值型应用时,交互的重点是对保存在事实中的数值型槽值的读取。不必通过建立用户自定义函

数导出规则中的汉字,也不必借助复杂包装类。

事实保存在 CLIPS 中,当要提取事实的时候,首先要得到指向事实的指针,通过调用函数 `Void * GetNextFact( newFact )` 实现<sup>[8]</sup>;区别不同的事实要借助不同的槽名,该方法的前提是要求不同的事实模板定义时,不能定义相同的槽名,否则函数 `int GetFactSlot( newFact, "x", &theValue )` 就无法区别不同的事实。根据指向事实的指针 `newFact`,槽名 `x`,取得槽值并把它存在结构体 `DATA_OBJECT theValue` 中。读取并传递槽值到 VC 时用函数 `DOToFloat( theValue )`。

## 2 CLIPS 动态链接库和 VC++6.0MFC 交互的具体步骤

### 2.1 把 CLIPS 源代码制作动态链接库

(1) 首先启动 VC++6.0,创建 WIN32 动态链接库工程名叫 CLIPS,把 CLIPS 源代码拷到工程目录下,把所有后缀名为 .c 格式的文件全部改为 .cpp 格式。

(2) 把所有 .cpp 格式和 .h 格式的文件加载到 Source Files 中,另外在 Header Files 文件中再建一个 lib.h,把 `#include "lib.h"` 加在需要导出的函数的文件中,在 lib.h 中使用关键词 `__declspec( dllexport )` 声明要导出的函数。

(3) 编译连接得到 CLIPS.dll 和 CLIPS.lib 两个文件。

应该注意的问题:动态链接库只能导出函数,而不能导出宏指令。在与 MFC 编辑框交互的地方有几个非常重要的宏指令,如 `DOToFloat` 用于在对话框输出时读取事实中的 Float 类型的槽值(在框架式知识表示方法中,槽用来表示事物的属性,槽值表示该属性的值),然后以 `m_E1 = DOToFloat( theValue )` 导出到对话框;所以要在用户对话框文件 `DLG.cpp` 中定义宏指令:

```
#define DOToFloat( target ) ( ( float ) ( ( ( struct floatHashNode * ) ( target, value ) )->contents ) )
```

该指令从源文件 `evaluatn.h` 中拷出,同时结构体 `Struct floatHashNode` 也必须从源代码 `symbol.h` 中拷到 MFC 对话框 .cpp 文件。

用户还可以直接采用外国 CLIPS 网站上做好的 CLIPS.dll 和 CLIPS.lib 的文件,这样可以省去好多时间,其中包含了近 1500 个导出函数。从 CLIPS.dll 中查看导出函数时,可以使用开始菜单,VC++ 工具 Depends 来查看。

### 2.2 DLL 的静态调用方式

把 CLIPS.dll 和 CLIPS.lib 复制到应用程序目录下,在对话框 `DLG.cpp` 文件,拷贝 `#pragma comment( lib, "CLIPS.lib" )`,对 DLL 文件的重定义信息,与

DLL 建立链接。并把结构体 struct DATA\_OBJECT 定义从 evaluatn.h 中拷出到对话框.cpp 文件,因为有的导出函数以该结构体作为参数,在声明导入函数时需要用到。声明导入函数关键词\_declspec(dllimport)。

### 2.3 从 MFC 编辑框中向 CLIPS 输入和导出数值

- (1) 从 CLIPS 中调用初始化函数: InitializeEnvironment();
- (2) 调用 CLIPS 载入函数: Load();
- (3) 调用 CLIPS 添加事实函数: AssertString(temp);
- (4) 运行推理函数: Run(-11);
- (5) 得到下一个事实地址,这个函数在编辑框输出时非常重要,因数据存储在事实的槽中,要得到槽值必须要得到指向该事实的指针,以下就是这样的一个函数:

```
Void * GetNextFact( newFact );
```

- (6) int GetFactSlot( newFact, "x", &theValue ); 根据指向事实的指针 newFact, 槽名 x, 取得槽值并把它存在结构体 DATA\_OBJECT theValue 中。函数 double DOToFloat( theValue ) 用来读取事实中的 Float 类型的槽值并传递到 VC 中,它的参数是一个结构体 theValue。如果存在指定槽名的事实返回值为 1, 否则为 0。如果要读取整型类型的槽值使用函数 DOToLong( target ), 读取字符型的槽值使用函数 DOToString( target )。

图 2 为推理机推理过程。

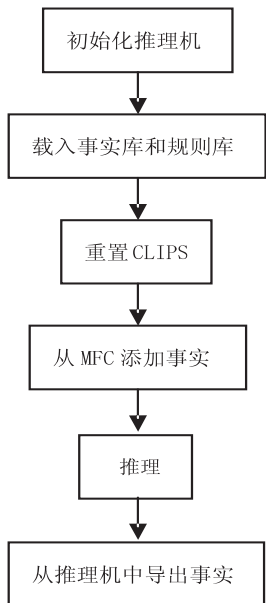


图 2 推理机推理过程

## 3 举例说明

以一个简单例子来说明,输入起重量,起升速度,吊具重量,稳态平均功率四个参数,这四个参数分别是一个事实的四个槽,以该事实的存在激活了一条规则,

规则执行部分是起升计算功率公式,得到一个新的事实,该事实只有一个槽,槽值是计算功率;以这个事实激活另一条规则,从已经存在的电动机库来选择合适的电动机,具体步骤如下:

起重机起升机构电动机功率计算公式如下:

$$\frac{(Q + G) \times v \times p_0}{m} = P$$

$Q$ —起升载荷; $G$ —吊具载荷; $m$ —吊具倍率; $v$ —起升速度; $p_0$ —稳态平均功率系数; $P$ —计算功率。

### 3.1 在 test.clp 中定义事实模板

CLIPS 中的事实模板:事实被创建之前,必须告知 CLIPS 一个给定关系名的合法槽的列表。共享相同的关系名和包含共同的信息的几组事实可以利用自定义模板结构来描述<sup>[9]</sup>。

第一个模板:

```
(deftemplate gonglv
```

```
(slot x) (slot y) (slot z) (slot w))
```

$x$ —起升载荷, $y$ —吊具载荷, $z$ —吊具倍率, $w$ —起升速度。

第二个模板:

```
(deftemplate zhi
```

```
(slot x1) (slot n1))
```

该模板定义的事实槽  $x_1$  用来保存计算功率值。槽  $n_1$  用来指定同步转速(电机级数),由列表框给出。

第三个模板:

```
(deftemplate dianji
```

```
(slot x2) (slot y2) (slot z2))
```

$x_2$ —电动机额定转速, $y_2$ —电动机额定功率, $z_2$ —电动机型号,用 A, B, C 简单代表。

为了便于读者阅读,人为给每个电动机一个编号,根据各个电机额定功率从小到大给一个数字编号。在知识库中存放电机时,以事实的方式来存储<sup>[10]</sup>:

```
(dianji (x2 950.0) (y2 50.0) (z2 A))
```

```
(dianji (x2 970.0) (y2 60.0) (z2 B))
```

```
(dianji (x2 750.0) (y2 70.0) (z2 C))
```

.....

### 3.2 在 test.clp 中定义规则

首先定义如下的一条计算功率的规则:

在符号=>之前是使这条规则激活的条件,即当存在(gonglv (x ? a) (y ? b) (z ? c) (w ? w))。? a 是变量。当从 VC++ 向 CLIPS 添加这样一个事实的时候,执行符号=>之后的语句,把计算出来的功率保存在事实 zhi 的槽中。CLIPS 中的数学表达式是按 LISP 风格表示的。在 LISP 和 CLIPS 中数学表达式必须写成前缀形式,如习惯写法 2+3 要写成(+ 2 3)。式子(/

$( (* ? w ( * ( + ? a ? b ) ? c ) ) 6000.0 )$  则是  $(Q + G) \times v \times p_0 = P$  的 CLIPS 表示方法。

```
(defrule gui1
(gonglv ( x ? a )( y ? b )( z ? c )( w ? w ))
=>(assert ( zhi ( x1 / ( * ? w ( * ( + ? a ? b )
? c ) ) 6000.0 ) ) )
)
```

取消所有功率不满足的电机型号: test 语句是测试条件元素。测试条件元素为计算规则 LHS(前件)上的表达式提供了有力的方法。可以用 test 来寻找特定的事实。以下规则是用 test 语句寻找到所有不满足功率要求的电动机,并用 retract 语句全部取消。?f 为事实地址。

```
(defrule gui2
(zhi ( x1 ? x )( n1 ? y ))
?f < - ( dianji ( y2 ? k )( z2 ? g ))
(test ( > ? x ? k ))
=>(retract ?f)
)
```

然后根据隶属度大小选取电动机:  
综合考虑同步转速和功率,权重系数 0.5:0.5。在满足的电动机中寻找和需求最匹配的电动机。

篇幅有限,且规则很长,不使用 CLIPS 写,笔者简单表达。对于任意两个电机,计算出隶属度  $\mu_{close}(f_1)$ ,  $\mu_{close}(f_2)$ , 取消隶属度小的那个电动机  $f_2$ , 该规则不断匹配,到最后只剩一个电动机,它的隶属度最高。

```
(defrule gui3
(zhi ( x1 ? x )( n1 ? y ))
?f1 < - ( dianji ( x2 ? a )( y2 ? k )( z2 ? g ))
?f2 < - ( dianji ( x2 ? b )( y2 ? j )( z2 ? h ))
(test ? [  $\mu_{close}(f_1) > \mu_{close}(f_2)$  ])
=>(retract ?f2)
)
```

$$\mu_{close}(f_1) = 0.5 \times \frac{1}{1 + (\frac{?k - ?x}{\beta_1})^2} + 0.5 \times \frac{1}{1 + (\frac{?a - ?y}{\beta_2})^2}$$
$$\mu_{close}(f_2) = 0.5 \times \frac{1}{1 + (\frac{?j - ?x}{\beta_1})^2} + 0.5 \times \frac{1}{1 + (\frac{?b - ?y}{\beta_2})^2}$$

3.3 从 MFC 对话框向 CLIPS 添加事实

初始化 CLIPS;载入事实库和规则库 test. clp 并启

动 CLIPS;在 VC++中用 sprintf 把事实(gonglv ( x %s )( y %s )( z %s )( w %s ))转换成字符串,调用 CLIPS 中函数 AssertString ( temp );添加到 CLIPS 中,运行 CLIPS。

3.4 从 CLIPS 导出计算功率和选定的电动机

计算功率保存在槽名为 x<sub>2</sub> 的事实中,读取槽名为 x<sub>1</sub> 的事实用以下函数:

```
while( newFact! =NULL)
{
if( GetFactSlot( newFact, " x2 ", &theValue )! =0)
{ double c = DOToFloat( theValue );
m_E5 = c;
}
GetNextFact( newFact );
}
```

同理可以提取出 y<sub>2</sub>, z<sub>2</sub> 的值。

4 结束语

CLIPS 因为高效快速的 Rete 算法和较好的可移植性在国外有着广泛的研究和应用,在国内会对 CLIPS 的应用逐步增多。笔者正在探索用 CLIPS 构建起重机专家系统。

参考文献:

[1] Giarratano J C, Riley G D. 专家系统原理和编程[M]. 北京:机械工业出版社,2010.

[2] 华珊珊,谢铨洋. XML 表示 CLIPS 知识库的研究[J]. 计算机技术与发展,2009,19(10):93-95.

[3] 谢晓方,姜 震. 一种结合 CLIPS 和 VC++开发专家系统的方法[J]. 计算机系统应用,2004(12):61-63.

[4] 吴 坚,蒙艳玫,黄云奇. CLIPS 在数控机床故障诊断专家系统中的应用[J]. 装备制造技术,2008(4):91-92.

[5] 潘全文,艾弘飞,房振旭. 专家系统的基本原理和基于 CLIPS 的专家系统设计与实现[J]. 飞机设计,2004(4):78-80.

[6] 邓海平,何玉林,杜 静,等. CLIPS 嵌入 VC++技术的实现与应用[J]. 计算机工程与应用,2005(15):88-90.

[7] 崔 莹,王华军,姚雪峰. Visual C++数据库实用编程 100 例[M]. 北京:中国铁道出版社,2007.

[8] NASA. CLIPS Advanced Programming Guide [M/OL]. 2008-03-22. <http://www.ghgcorp.com/clips/CLIPS.html>.

[9] NASA. CLIPS Basic Programming Guide [M/OL]. 2008-03-22. <http://www.ghgcorp.com/clips/CLIPS.html>.

[10] Giarratano J C. CLIPS User's Guide [M/OL]. 2007-12-31. <http://www.ghgcorp.com/clips/CLIPS.html>.

## CLIPS数值应用与VC++的交互

作者: [宁志强](#), [陶元芳](#), [刘晓莲](#)  
作者单位: [太原科技大学 机械学院, 山西 太原030024](#)  
刊名: [计算机技术与发展](#)  
英文刊名: [Computer Technology and Development](#)  
年, 卷(期): 2013(4)

本文链接: [http://d.g.wanfangdata.com.cn/Periodical\\_wjfz201304057.aspx](http://d.g.wanfangdata.com.cn/Periodical_wjfz201304057.aspx)