

CLIPS 嵌入 .NET 平台技术与实现

The Realization and Application of the Technology Embedding CLIPS within .NET Platform

王 华 李鹏波 (国防科技大学机电工程与自动化学院 长沙 410073)

摘要:本文介绍了 CLIPS 嵌入高级语言的方法,其中包括 CLIPS 嵌入 VC++, CLIPS 嵌入 C++ Builder,比较分析了 CLIPS 嵌入接口的不同,重点研究分析了 CLIPS 如何嵌入 C#.NET,进行专家系统的开发与实现。

关键词:CLIPS 嵌入 C#.NET 专家系统

1 引言

在 CLIPS 和高级语言相结合开发专家系统领域,已经有了很多的实践经验和成功案例,比如 CLIPS 嵌入 VC++, CLIPS 嵌入 Delphi 开发的专家系统都已经在广泛应用,但随着程序设计方法的发展,.NET 平台和 C# 语言的出现,.NET 是针对面向对象、Internet 开发而专门设计的全新开发平台,C#建立在 C 的高性能、C++ 的面向对象结构、Java 的安全性和 VB 的快速开发等语言的众多经验之上,是理想的用于开发以组件为基础的、多层分布式 Web 应用程序的全新语言。而专家系统发展方向之一的是分布式的专家系统,CLIPS 与 C#的结合,正适合这种发展趋势,CLIPS 嵌入 .NET 平台技术的研究与实现对专家系统的探索发展具有一定的经验积累和推动作用。

2 CLIPS 介绍

专家系统是人工智能领域的一个重要研究方向,CLIPS (C Language Integrated Production System)是由美国航空航天局约翰逊空间中心开发的一种专家系统工具,由 C 语言编写。最初的版本发布于 1985 年,当前的最新版本是 CLIPS6.23。

CLIPS 的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 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和对象实例集。

CLIPS 既支持基于规则的编程,又支持面向对象和面向过程的编程,同时它具有很好的兼容性和可扩展性。用 CLIPS 开发的程序能与 C、Ada、Pascal 等语言兼容,尤其是 6.1 以后的版本能在 C++ 编译器中编译。

由于具有兼容性好、知识表达方式灵活、运行效率高、集成性好、可免费使用且可靠性高的特点,CLIPS 已经成为目前最为广泛的专家系统开发工具之一。

3 CLIPS 嵌入 VC++ 和 C++ Builder

由于 CLIPS 可以直接在 C++ 编译器中编译、兼容性强且源代码开放,CLIPS 和 VC++、C++ Builder 程序的嵌套使用主要有两种方法:直接嵌入式的混合编程和 DLL 方式的混合编程。

3.1 直接嵌入式混合编程

CLIPS 的源代码是公开的,用户可以自由使用和根据自身需求进行修改,由于 CLIPS 在 C++ 下可编译,因此可以把 CLIPS 源代码直接添加到 VC++、C++ Builder 程序中,CLIPS 函数和功能可以直接通过对源代码中函数的调用来实现,然后和主程序一起编译、链接形成一个整体。

使用时,以 VC 为例,首先把 CLIPS 核心源文件添加

到工程中,其次 VC 程序必须提供一个主程序以代替 CLIPS 源代码中的主程序 (WinMain. c),在主程序中包含 "clips. h" 头文件。在其中通过调用 CLIPS 源代码中的 InitializeCLIPS 函数来完成 CLIPS 的初始化。而且不管 CLIPS 是否调用外部函数,UserFunctions 函数必须在主程序中被定义。这些完成后就可以和其它子程序一样进行调用。和 VC 程序一起编译、链接生成 .EXE 程序。

3.2 DLL 方式的混合编程

DLL 方式把 CLIPS 的核心命令如: Load、Reset、Run 等封装成一个动态链接库 clips. dll (CLIPS 开发者直接提供),通过调用此动态链接库中的函数可以实现 CLIPS 的绝大多数核心功能。所有程序可以共享这个动态链接库,有助于减小程序的规模,降低内存需求。

① 嵌入 VC 时,通过引入 clips. dll 文件,我们将 CLIPS 内核嵌入 VC++ 编写的主程序中,从而实现 CLIPS 和 VC++ 的混合编程。注意,这里 clips. dll 只是一个动态库,使用时不需要注册。

② 嵌入 C++ Builder 时,程序调用 CLIPS 动态链接库至少需要两个文件:库文件 clips. dll 和输入接口库文件 clips. lib。CLIPS 开发者提供的动态链接库是由 VC++ 生成的,存在 VC++ 动态库和 C++ Builder 接口的不兼容性。

首先是两者输入接口库文件的格式不统一,VC++ 使用 COFF 库文件格式,C++ Builder 使用 OMF 格式,所以需要库文件格式转换;

其次 VC++ 和 C++ Builder 连接器命名规则不同,DLL 文件中的每一个函数都有一个连接时的函数名称,连接器在编译程序时使用连接器名称处理函数。VC++ 一般使用 _stdcall 作为引出函数的类型,而 C++ Builder 使用 _cdecl 作为引入函数的类型。如果连接器不能处理某个函数,就会产生一个外部错误。因此需要创建一个对 VC++ 动态库和 C++ Builder 两者都兼容的新的输入接口库文件 clips. lib。

以上两个方面的不兼容性分别可以利用 C++ Builder 提供的工具 implib. exe 进行库文件格式转换,impdef. exe 进行输入接口库文件格式转换。impdef. exe 为 clips. dll 创建一个包含所有引出函数的 DEF 文件,修改 DEF 文件使其符合 C++ Builder 的格式,然后利用修改后的 DEF 文件创建一个可被 C++ Builder 利

用的输入接口库文件 clips. lib。

3.3 CLIPS 嵌入式编程的分析比较

通过前面介绍可以看出,在以往的 CLIPS 嵌入高级编程语言中,最常用的方式就是直接嵌入式和 DLL 方式的混合编程。

① 直接嵌入式,要求高级编程语言能够对 C 或者 C++ 进行编译,这样 CLIPS 的源代码才能直接嵌入到程序中,否则直接嵌入式混合编程不能实现;

② DLL 方式的混合编程,这种方式涉及到生成 DLL 的接口和使用 DLL 的接口的兼容性问题,也就是说如果是在 VC 下创建的 DLL,在 VC 下可以直接调用,而在 C++ Builder 下则由于接口文件格式不同必须进行转换方能使用。

4 CLIPS 嵌入 Visual C#. NET 中

由于 CLIPS 在 .NET 平台下不能直接编译,所以 CLIPS 不能源代码直接嵌入 .NET 平台下,但 CLIPS 是 C 语言编写,且免费开放源代码,可扩展性和兼容性都比较强,与其它高级语言的接口比较方便,这样 CLIPS 嵌入 C# 中常用方式:一是生成 ActiveX 控件导入;二是生成 COM 组件调用;三是生成 DLL 动态链接库直接引用;四是通过调用 CLIPS 进程嵌入 C# 框架。

4.1 CLIPS 的 ActiveX 控件调用

ActiveX 控件是一个动态链接库,是作为基于 COM 服务器进行操作的,并且可以嵌入在包容器宿主应用程序中,ActiveX 控件的前身就是 OLE 控件。由于 ActiveX 控件与开发平台无关,因此,在一种编程语言上开发的 ActiveX 控件可以无须任何修改,即可在另一种编程语言中使用,其效果如同使用 Windows 通用控件一样。如:在 VC++ 中开发的 ActiveX 控件,不作任何修改即可应用于 C# 中。由此可见,通过使用 ActiveX 控件即可实现快速小型的组件重用、实现代码共享,从而提高编程效率。

一般来说,从 Internet 下载的 ActiveX 控件,或从其他途径得到的 ActiveX 控件,用户要在 Windows 中被正确使用,首先必须将控件文件 (*.OCX) 复制到硬盘中,然后将其在 Windows 中注册,未在 Windows 中注册过的 ActiveX 控件是不能使用的。注册 ActiveX 控件一般来说有三种途径:

(1) 使用 Regsvr32. exe 程序对 ActiveX 控件进行

注册;

(2) 使用安装程序制作软件: InstallShield;

(3) 在应用程序中加入注册代码。

这里我们使用方式 2, 从网上可以直接下载到 CLIPS. OCX 控件, 具体步骤如下:

① 下载 CLIPS. OCX 控件, 直接运行安装。

② 在 .NET 开发环境中打开菜单 Tools—>Customize Toolbox, 在 COM Components 选项卡中找到安装的 CLIPActiveXControl Control 对象选中即可, 点击“确定”返回, 这时在 Toolbox 面板上会出现该控件。

③ 使用时, 直接把该控件拖拽到 Form 上即可。

CLIPS 的 ActiveX 控件嵌入后, 在 C# 中调用 CLIPS 控件, 添加如下代码可实现:

```
//声明变量
private AxCLIPACTIVEXCONTROLlib. AxCLIPActiveXControl axCLIPActiveXControl1;
axCLIPActiveXControl1. Clear();
//加载推理规则
axCLIPActiveXControl1. Load("d:\whitest\checkFaultRules. clp");
//加载初始事实
axCLIPActiveXControl1. Load("d:\whitest\facts. clp");
//激活规则
axCLIPActiveXControl1. Reset();
//运行
axCLIPActiveXControl1. Run(-1);
```

通过上述语句就可以实现加载事实和规则, 启动 CLIPS, 运行所有激活的规则进行专家系统推理。

4.2 生成 COM 组件调用

CLIPS 由 c 语言编写, 可以根据自己需要对源代码进行分析、修改、编译, 在 C++ 下编译成所需的 COM 组件。在 .NET 平台下通过调用 CLIPS 生成的 COM 组件, 实现 CLIPS 嵌入 C#。通过 C 或 C++ 生成 CLIPS 的 COM 组件, 此 COM 组件具有语言无关性, 编写组件对象使用的语言与编写客户程序使用的语言可以不同; 具有进程透明性, COM 所提供的服务组件对象在实现时有两种进程模型: 进程内对象和进程外对象。进程内模型效率高, 但组件不稳定会引起客户进程崩溃, 组件可能会危及客户; 进程外模型优点是稳定性好, 组件

不会危及客户程序, 一个组件进程可以为多个客户进程提供服务可, 但进程外组件开销大, 而且调用效率相对较低一些。

COM 与 C# 的互操作是通过早或迟绑定实现的, 早绑定允许使用编译时知道的类型编程, 而迟绑定要求使用通过动态发现的类型编程, 在 C# 一侧使用反射, COM 一侧使用 IDispatch。当从 C# 中调用 COM 程序时, 早绑定通过为 COM 对象及其接口提供配件形式的元数据实现。Tlbimp. exe 获取 COM 类型库并在配件中生成等价的元数据。使用已生成的配件, 就可能实例化并调用 COM 对象, 就像其他 C# 对象一样。当实例化 COM 对象时, 实际上是在处理一个称为 RCW (Runtime Callable Wrapper, 运行时可调用包裹器) 的代理。RCW 负责管理 COM 对象的生存期需求, 并将其上调用的方法转换为 COM 对象上的相应调用。

4.3 生成 DLL 动态链接库直接引用

CLIPS 直接生成 DLL 动态链接库, 在 .NET 平台下通过 P/Invoke (Platform invoke facility, 平台调用设施) 可以允许 C# 访问未管制 DLL 的函数、结构和回调。如果想调用 dll 中函数, 可以写一个带 DllImport 属性信息的 static extern 方法。PInvoke 然后会找到并载入所需 DLL, 并解析所请求的函数入口点。CLR 中有一个列集器, 知道如何在 .NET 类型和未管制类型之间转换参数以及返回值。

例如, 如果想调用 clips. dll 的 InitializeCLIPS() 函数方法如下:

```
Using System. Runtime. InteropServices;
[DllImport("clips. dll")]
Static extern void InitializeCLIPS();
```

在具体的实现中, 根据自己的需要声明、调用函数, 然后在 C# 中添加在推理模块中, 以便进行专家系统推理。

4.4 通过调用 CLIPS 进程嵌入 C# 框架

这种方法是在 C# 框架中直接嵌入 CLIPS 进程, 也就是把 CLIPS. EXE 进程嵌入 C# 的框架中直接使用 CLIPS 提供的用户界面进行推理。

在 C# 中通过使用 Process 类, 用户可以访问本地或者远程机器上的进程, 并且可以启动和终止本地系统中的进程。这个类位于命名空间 System. Diagnostics 中, 并且通过程序集 System. dll 来实现, 所以使用

它之前,必须使用 using System. Diagnostics 语句导入命名空间,并且把程序集 System. dll 添加到项目的 References 文件中。Process 类中包含一些静态方法,可以使用这些方法来执行获取系统中当前运行的所有进程、获取特定进程和启动进程等操作。在使用进程对象的属性和方法之前,必须获取或创建 Process 对象实例。用户可以使用“工具箱”面板的“组件”部分把 Process 添加到窗体中。具体实现可如下进一步完善。

例如:Process p = new Process(); //创建对象实例

```
//设置进程对象的属性
Process.StartInfo.FileName = "d:\CLIPSW32.
EXE";
...
p.Start(); //启动进程
p.WaitForInputIdle(); //等待进程初始化完毕
...
p.CloseMainWindow(); //结束进程
p.WaitForExit(); //等待进程结束
```

5 结束语

本文比较了 CLIPS 嵌入高级语言的方法,重点介绍了 CLIPS 嵌入 .NET 的方法,根据 CLIPS 嵌入平台的不同,嵌入方式要进行相应的接口变换,这种嵌入式编程

的优势在于,两者相结合开发专家系统有利于发挥 CLIPS 强大的逻辑推理功能,C#等高级语言擅长的图形图像、数据库交互、数值运算、数据采集、与外设数据通信等重要功能,实现两者优势互补、扬长避短,为开发分布式专家系统的发展、开发与实现提供了切实可行的方案,并且本文的理论方法为相关领域的实践提供了一定的引导启发。

参考文献

- 1 专家系统原理与编程,(美)Joseph Giarratano Gary Riley 著,印鉴等译。
- 2 专家系统工具 CLIPS 及其应用,吴鹤龄 编译。
- 3 C#程序设计 Jesse Liberty 著,刘基诚 译。
- 4 C#精髓 Ben Albabari, Peter Drayton, Brad Merrill 刘基诚 译。
- 5 COM 原理与应用,潘爱民 著。
- 6 CLIPS Reference Manual Advanced Programming Guide Version 6.23 January 31st 2005.
- 7 CLIPS Reference Manua Basic Programming Guide Version 6.23 June 1st 2005.
- 8 基于 DLL 方式的 C++ Builder 和 CLIPS 混合编程开发专家系统,钟明、王钦友,机电工程 2004 年第 21 卷,第 5 期。