

DOI:10.3969/j.issn.1001-4551.2016.04.024

工业机器人通用图形化编程软件的研究*

李煜卉, 彭勇刚*

(浙江大学 电气工程学院, 浙江 杭州 310027)

摘要:针对工业机器人控制软件中传统文本编程方式带来的操作难度大和通用性不够的问题,对图形化编程方式进行了研究,将 XML 技术和图形界面实现技术应用到编程软件中,提出了一种图形化编程软件方案,包括软件总体设计思路及实现方法;通过对工业机器人通用操作的参数分析,建立了通用操作到图标式任务的转换关系,且将图标式任务转换为下方解释器能够识别的中间文件;通过设计用于表示机器人操作的图标的数据结构以及搭建能够自行跟踪用户所有操作并且具有自结构性(本身包含顺序和父子信息)的编辑器,实现了图标式任务的绘制以及记录;定义了与图标数据结构相对应的节点,采取同样具有自结构性的中间文件对图形任务进行了保存,最后生成了逻辑正确、数据无误且能够被下方解释器识别的文件。研究表明,用户通过拖曳和连线进行图形化编程,降低了编程难度,同时生成的文件可移植性高,提高了软件通用性。

关键词:工业机器人;图形化编程;数据结构;插件开发;通用性

中图分类号:TP242.2

文献标志码:A

文章编号:1001-4551(2016)04-0502-05

Implementation of graphical programming software for industrial robot

LI Yu-hui, PENG Yong-gang

(School of Electrical Engineering, Zhejiang University, Hangzhou 310027, China)

Abstract: Aiming at the problem of the traditional text-based programming was difficult to operate and lack of commonality, the method of graphical programming was studied, the XML technology and the graphical interfaces achieve technology were applying to software. A structure design of graphical programming software for industrial robot was proposed, including the integrated design ideas and concrete implementation method. Through the analysis of the parameters for general industrial robot operation, the corresponding relation between robot manipulations and graphical symbols was established, then the flowchart-style graphics task consisted of symbols and connections was converted to intermediate file which can be explained by interpreter and hardware equipment after series communication. The data structure of symbols and implementation of self-structure editor which could record operation of users were defined, so that graphical task can be easily completed. Furthermore, the correspond realation between node and date structure of logo were defined, the intermediate file with self-structure could save graphical task so the correct file can be explained by thterpreter was generated in the end. The results indicate that it can fully ease the programming for industrial robot and improve the flexibility and practicality of robot software system.

Key words: industrial robot; graphical programming; data structure; plug-in development; commonality

0 引言

近年来,图形化方式已经在众多领域得到应用,比较熟知的图形化编程语言有 LabView, Matlab/Simu-

link, UML 和 RCX。在机器人领域,图标式(icon-based)编程方式首先在教育行业兴起,随着工业机器人行业的迅速发展,受过特定机器人语言编程培训的工人量不足以满足扩大的市场需求,因此图形化编程

收稿日期:2015-11-27

基金项目:杭州市产业链重大科技创新资助项目(20132111A04)

作者简介:李煜卉(1991-),女,湖南邵阳人,主要从事工业机器人控制软件的研究。E-mail:liyuhui20093357@126.com

通信联系人:彭勇刚,男,博士,副教授,博士生导师。E-mail:pengyg@zju.edu.cn

开始在工业机器人领域火热^[1]。1999年,欧洲合作研究计划 AMIRA(advanced man machine interface for robot system application)提出了工业机器人系统应用程序的用户界面风格规范^[2]。与传统的编程方式相比,流程图式的程序结构更为直观,用户无需考虑语法、句法的严格要求,可以将更多的精力用于如何更有效地、更优化的实现设计任务目标^[3]。同时,参数和操作的设置也更为简洁。

目前,国内外各机器人公司以及研究机构的控制软件平台开发主要集中在离线编程与仿真系统方面,主要有 Tecnomatix 公司开发的 RObCAD^[4], Robotsimulations 公司开发的 Workspace, ABB 公司开发的 Robot-Studio 系统^[5]等。国内也有不少大学、研究所等进行离线编程系统开发,其中哈尔滨工业大学现代焊接生产技术国家重点实验室是国内对离线编程研究最早的机构,多年来不断累积取得了很多突破性的进步^[6-7],实现了相关技术积累。这些系统离线编程和仿真功能强大,但是主要适合本公司或者特定结构的机器人开发和仿真,系统的开放性不够,并且大多数的离线编程语言还是基于描述性、或者半描述性半流程图式的机器人语言,从而限制了离线编程系统的使用和广泛推广。

本研究提出一种应用于工业机器人控制软件平台的图形化编程软件系统,以提高工业机器人离线编程系统的通用性和易用性。用户用对应的图形模块来表示工业机器人的逻辑操作或者运动操作,通过连线生成图形化文件,避免传统编程方式对句法、语法的严格要求,以及代码序列程序结构和问题求解的结构化策略的形象表达较差等弊端^[8]。

1 软件系统整体设计

1.1 整体框架设计

工业机器人离线编程系统总体由两层组成,系统整体框架图如图1所示。

本研究基于控制集中和功能分散的思想,将上层人机交互、运动规划、图形化编程等功能和下层运动控制实现进行有机分离。上位软件平台负责提供友好人机接口,维护零件模具库,提供参数设置接口,并可通过运动规划模块根据模具库中现有模具或通过示教自动生成运动控制程序,通过可扩展的下位机接口驱动底层控制系统。其中图形化编程软件系统为工业机器人离线编程系统的子系统,负责实现流程图式的任务绘制和中间文件的生成。

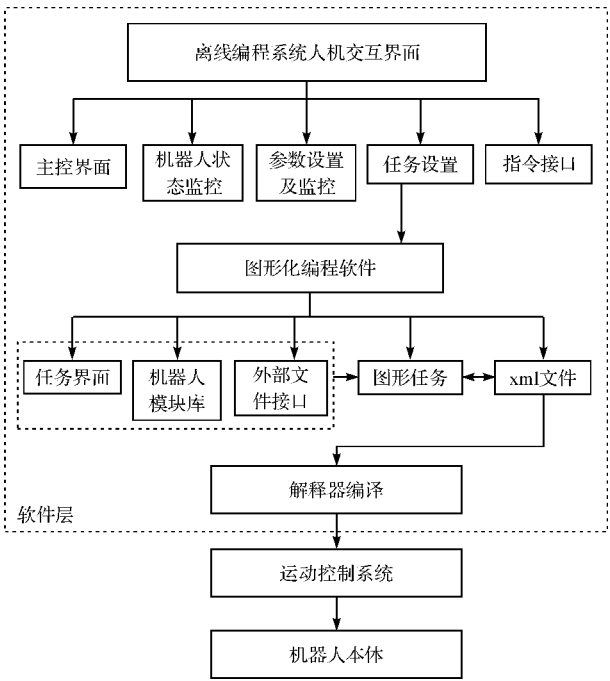


图1 离线编程系统整体框架图

1.2 图形化编程软件的整体设计思路

图形化编程软件系统设计思路总体可以划分为4个阶段:自结构的任务编辑界面的实现,工业机器人操作的模块化封装及其数据结构设计,图形任务与XML(extensible markup language)文件的互相转换,任务的完善性合法性检查。

图形化任务由封装为图标式的任务元素和表示任务流动方向的分支连线组成。这些任务元素是针对工业机器人的不同操作而封装起来的不同模块,包含运动参数、图形参数、连接参数等信息。这些模块可分为运动控制,条件控制和外部文件导入等几大类。通过 MVC(model-control-view)模式实现的任务编辑界面不仅仅是提供拖动模块、连接模块功能的界面,还能够及时响应每一步操作带来的内部变化,如任务的顺序结构变动,任务的父子结构变动,也就是说,所形成的图形化文件是有看不见的内在结构的。XML文件也是自结构化的,本身文件里就包含了父子结构和顺序结构,因此可以与图形化文件的编程元素以及位置、从属关系形成一一对应,不论是转换还是反转换。

2 实现方法和技术

2.1 任务编辑界面的实现

Eclipse 不仅仅是一个平台,同时还具有十分耀眼的插件体系结构,所有的业务功能都封装在 Eclipse 的插件

当中,Eclipse 为插件提供运行基础^[9]。图形化编程首先要实现一个可以拖动图标模块并且将模块进行连线的任务编辑界面,因此本研究选取 Eclipse 的 GEF (graphical editor framework) 插件在 Eclipse 上进行开发,最后将插件发布为脱离 Eclipse 运行的独立运行软件产品。

GEF 全称图形化编程框架,其包含的编辑器已经具备 drag&drop、undo&redo、move&delete&resize 等基本图形编辑器操作功能,并且自带可以放置各种工具的调色板和用于参数设置的属性编辑器,所以十分适合开发流程图编辑器。GEF 中,所有的图标模块对于任务编辑界面 Diagram 而言都是一个 Node,而后期定义的封装在 Node 里的机器人操作之间的区别仅仅在于生成 XML 文件时与运动相关的参数数据。即,在进行任务的逻辑处理时,所有的图形模块对于 Diagram 而言都是 Node,没有区别。

对这些 Node 的操作(如删除、添加、连线、成为前后任务或者父子任务)都采取相同的处理原则:委托,而这种委托是 GEF 基于 MVC (model-view-controller) 模式实现的,controller 是整个框架的核心,主要用于操作功能的编写,view 是轻量级图形界面显示,model 是用户设计最终被操作和编辑的对象原型,通过这种方式可以任意组合模型与视图并且配置不同操作,十分灵活。并且通过这种模式可以将用户的界面操作转化为模型的属性变化,从而为图形化文件的逻辑建立一个看不见但是准确迅速的内在结构。任务编辑过程时 MVC 模式下的 GEF 工作原理如图 2 所示。

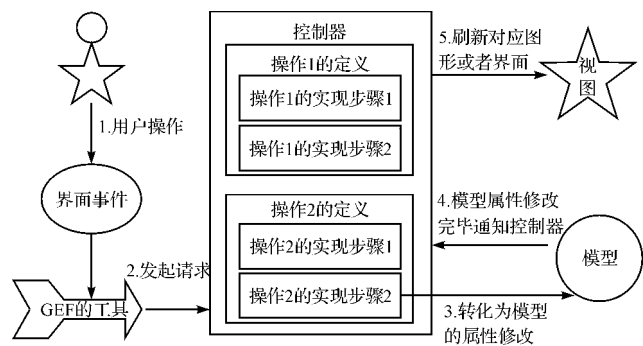


图 2 MVC 模式下 GEF 的工作原理

2.2 工业机器人操作的模块化封装

在实现了图形化编程的编辑界面之后,需要将在图形化界面的属性编辑栏里为每个模块所设置的运动相关参数到 XML 文件中,接着传递至下方解释器、控制系统,最终传递至硬件层,所以需要对机器人操作进行封装,并抽象为 GEF 中的模型类(Model)。

这套离线编程系统的图形化编程软件是在与杭州新松公司合作的基础上建立,因此对模块的定义

是在新松机器人的机器编程语言的类别上,进一步通用化完成,大致分为 5 类:

- (1) 执行模块库:初始化,结束任务,子任务开始和子任务结束;
- (2) 功能模块库:等待触发信号,延时,输出信号;
- (4) 控制模块库:组合运动,while 条件循环,for 条件循环,if-else 条件判断;
- (4) 运动模块库:圆弧运动,直线运动,样条曲线,轴旋转运动,轴平移运动;
- (5) 文件导入模块库:cad 文件导入,示教曲线导入。

由于 java 面向对象设计的原则,模块日后的拓展也十分灵活,一整套控制器的实现不变,只需要定义好新模块的运动参数添加进 Model 库即可。

为了适应更多的复杂曲线需求,本研究在基本运动的基础上增加了两个接口,用于导入 cad 文件与示教曲线文件。因为导入的文件只包含运动信息,并没有模块化和图形信息,首先要将文件进行解析读取运动参数,然后按照标准数据结构模块化成标准模块库中文件,最后进行保存。

2.3 图形任务文件与 XML 文件的转换

在应用图形化编程技术的过程中使用 XML 技术是可行的,因为 XML 文档具有独特的自描述性,且自结构化,既能自定义节点信息以实现图形化编程元素的 XML 格式表示,又能很好地对应图形文件的逻辑信息^[10]。并且 XML 是编程技术中十分通用的语言,利用它作为图形化编程的标准接口语言,可以增加软件的通用性。由前文可知,图形文件中的模块都拥有自己的模型数据结构,在将图形文件转换为 XML 文件时,需要将数据结构中的连线信息和父子信息读取出来,然后转换为 XML 文件中的前后信息与父子信息,反之同理。其中,XML 的读写是用 Dom4j 完成,Dom4j 是一个 Java 的 XML API,它封装了读写 XML 文件的主要操作,开发时只需完成核心操作比如添加属性进去即可。

图形文件与 xml 文件的相互转换示意图如图 3 所示。为了简化图形,数据结构中的尺寸与坐标信息略去。

2.4 任务的完善性合法性检查

图形文件与 XML 文件在图标元素和逻辑上正确对应,并且运动参数也正确传递,在这之后还需要提供任务的完善性与合法性检测。在图形化编程过程中,用户所生成的任务文件有可能是不完整的、临时保存的或者是非法的,所以需要提供一个标志给解释器,在 XML 文件中标记出该任务文件的状态,因此本研究为 Diagram 增加两个属性 finish 与 valid,值为 true 或者 false。

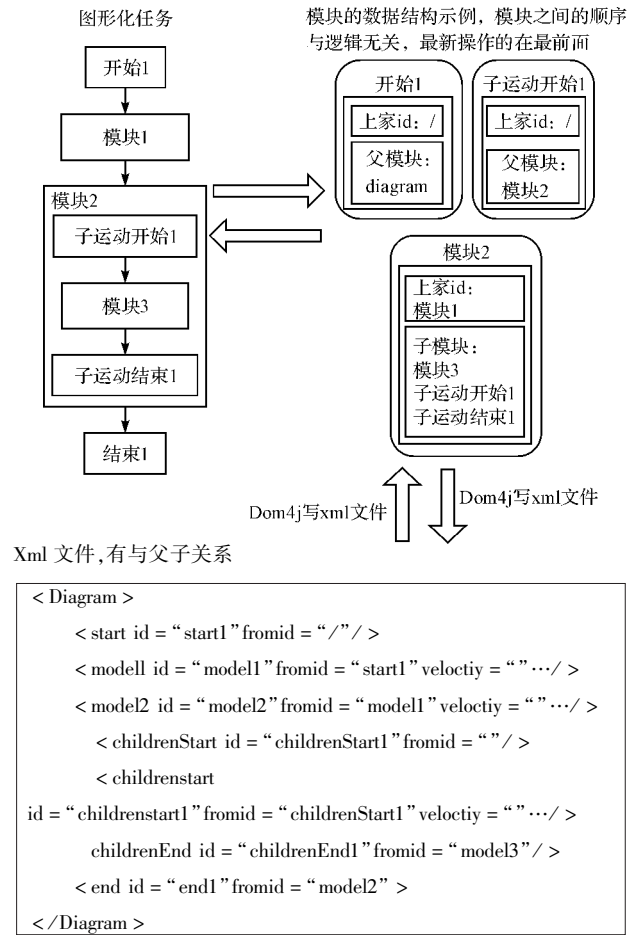


图 3 图形文件与 xml 文件的相互转换

首先是任务完善性检查,可能存在以下几种 finish = flase 的情况:从任务的开始到结束,至少有一条支路是不完整的;从任务的开始到结束,所有的支路都完整,但是存在未有任何连接的独立模块等。然后是合法性检查,可能存在以下几种 valid = false 的情况:存在闭环(循环任务都放置在了循环模块内,因此正常情况下不会出现闭环),条件判断支路不完整、支路个数超额或者支路条件相同等。

完善性检查和合法性检查都在用户点击保存时同步进行,更新 finish 和 valid 的属性值到 xml 文件中。前面已经说明过,模块的数据结构的定义和编辑器的自结构性,图形化文件是具有内在结构的,所以通过前序遍历图形化文件,如果出现从任务开始模块(仅此一个)经过某条支路无法抵达任务结束模块(仅此一个),则判定任务不完整。而未有任何连接的独立模块会在遍历后位于 XML 文件最前方,这是由于 Dom4j 的读写方式决定的,解析器在解析 xml 文件时,需要将任务开始模块作为起始读取点。其中在遍历过程中遇见条件判断模块,需要单独进行对分支进行合

法性检查。

但是如果存在闭环情况,前序遍历会陷入死循环,因此在此之前要进行是否存在闭环这种非法情况。如果存在闭环,也就是说某个结点的关系网里会包含自己,因此首先要为每个节点建立关系网,然后进行判定。这里的关系网不是用来还原文件整体架构,所以不需要建立复杂结构。从前面的数据结构也可以看出,模块的定义和链表的定义是一致的,包含两个信息,value 和 next,问题就转换为如何判定一个链表中是否存在环。设置两个指针,开始都指向链表头,然后其中一个指针每次向前走一步,另一个指针每次向前走两步,如果走的快的遇到 NULL 了,证明该链表中没有环,如果有环,快的指针每次都要比慢的多走一步,最终两个指针会相遇(这里快指针不会跳过慢指针而不相遇,因为它每次都只比慢指针多走一个单位)。

3 实验及结果分析

本研究将基于 Eclipse 及其 GEF 插件开发的插件程序发布为独立产品,最后生成的图形化编程软件图形界面如图 4 所示。

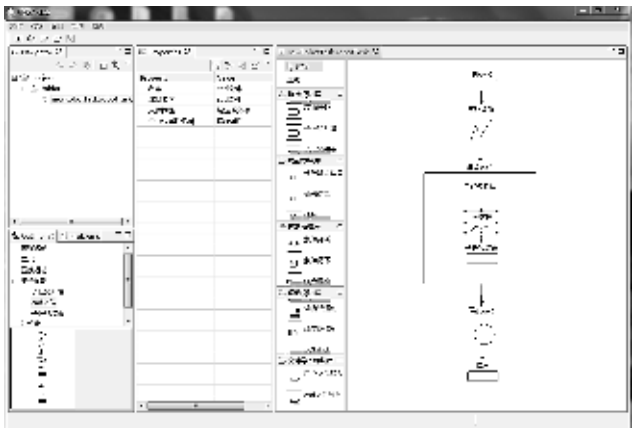


图 4 编程界面

该界面分为 4 个部分,工程区可以看见自己创建的所有文件;大纲区可以看见已经添加的所有模块以及层次结构;属性栏可以对模块的运动属性进行设置、修改、查看;最后是带有调色板的编辑区,调色板内为模块和连线,编辑区内为正在生成的任务。

保存的 XML 文件如图 5 所示。

经过多次实验测试表明,该图形化编程系统操作简单,生成的任务逻辑正确,在整个项目所研发的机器人控制软件平台中与解释器接口匹配,并且传递至底层的运动参数能够满足底层硬件的运动规划等需求。

```

<?xml version="1.0" encoding="GBK" ?>

<Root>
  <Diagram Name="Diagram" Finish="true" valid="true">
    <Start ModelName="Start" ID="81910QsUa6" Name="Start" LocationX="221"
    LocationY="19" FromID="no input" flag="null" bendpoint="null"
    message="4.1.1.2.8.8:-1.1.-2.2.-3.3.-4.4.0.1.2.3.1;D:\important\test\file
    \test8.dxf" number="4" shift="5 4" rotate="1 2" startPos="0 1 2 3"/>
    <LinearMotion ModelName="LinearMotion" ID="589XLLGnF" Name="LinearMotion"
    LocationX="218" LocationY="04" FromID="81910QsUa6" flag="null"
    bendpoint="null" CoorType="0" Velocity="0.0" points="0.0,0.0,0.0" ox="0.0000"
    oy="0.0000" oz="0.0000" fz="1"/>
    <Combination ModelName="Combination" ID="GaTp2r11" Name="Combination"
    LocationX="141" LocationY="180" FromID="589XLLGnF" flag="null"
    bendpoint="null" Finish="true" width="212" height="228" CoorType="-1">
      <ContainerStart ModelName="ContainerStart" ID="qg73fapMK"
      Name="ContainerStart" LocationX="77" LocationY="37" FromID="no input"
      flag="null" bendpoint="null"/>
      <Cad ModelName="Cad" ID="0C72NkTKsq" Name="Cad" LocationX="18"
      LocationY="93" FromID="qg73fapMK" flag="null" bendpoint="null" button="please
      choose the file" edit="0"/>
      <ContainerEnd ModelName="ContainerEnd" ID="9ELF52E38k"
      Name="ContainerEnd" LocationX="80" LocationY="151" FromID="0C72NkTKsq"
      flag="null" bendpoint="null"/>
    </Combination>
    <CircleMotion ModelName="CircleMotion" ID="V5pha71938" Name="CircleMotion"
    LocationX="221" LocationY="458" FromID="GaTp2r11" flag="null"
    bendpoint="null" CoorType="0" Velocity="0.0" points="0.0,0.0,0.0|0.0,0.0,0.0"
    ox="0.0000" oy="0.0000" oz="0.0000" fz="1"/>
    <End ModelName="End" ID="v1R11D79i5" Name="End" LocationX="223"
    LocationY="859" FromID="V5pha71938" flag="null" bendpoint="null"/>
  </Diagram>
</Root>

```

图 5 对应与图 4 的 XML 文件

4 结束语

本研究提出了基于流程图式的应用程序设计思想,在软件的实现过程中使用了现今十分成熟的 Eclipse 插件开发方式,并且采取 XML 文件这一常用的编程表达方式,通过封装机器人的通用操作和定义对应数据结构,将编程过程简化,增强了软件的通用性。同时本研究定义了图形化文件与 XML 文件的转换协议,通过 MVC 模式将其联系起来,辅以检测技术,提高了正确性和可移植性。最后生成的图形化编程软件经过多次操作和检测,可以证实其操作简单且逻辑正确。

综上所述,本研究中所阐述的系统结构设计思路 and 实现方案是可行的,能够实现一个稳定可靠的图形

化编程软件系统。由于该系统从设计实现上采用的始终是模块化设计思想,为系统的易维护性、可重用性奠定了很好的基础。

参考文献 (References):

- [1] LEES D, LEIFER L J. A Graphical programming language for robots operating in lightly structured environments [C]// Proceeding 1993 IEEE Robotics and Automation. Piscataway: [s. n.], 1993: 648-653.
- [2] KAZI A, SEYFARTH M. The MOPRH a style guide for icon-based programming [J]. VDI Berichte, 2002, 1679 (1): 41-46.
- [3] 邱长伍, 曹其新. 机器人图形化编程与三维仿真环境 [J]. 机器人, 2005, 27 (5): 436-440.
- [4] 邓伟俊. 基于 Robcad 软件的机器人基于软件的机器人点焊离线编程系统研究 [J]. 金属加工: 热加工, 2014 (10): 62-63.
- [5] 谭定, 李亮玉, 王天琪. 基于 RobotStudio 的双机器人协调仿真 [J]. 焊接技术, 2011, 40 (6): 45-48.
- [6] 刘圣祥, 高洪明, 张广军, 等. 弧焊机器人离线编程与仿真技术的研究现状及发展趋势 [J]. 焊接学报, 2007, 28 (7): 21-25.
- [7] 何广忠. 机器人弧焊离线编程系统及其自动编程技术研究 [D]. 哈尔滨: 哈尔滨工业大学材料学院, 2006.
- [8] CREMERS D, TISCHHAUSER F, WEICKERT J, et al. Diffusion snakes: Introducing statistical shape knowledge into the Mumford-Shah functional [J]. International Journal of Computer Version, 2002, 50 (3): 295-313.
- [9] 张鹏, 姜昊, 许力. Eclipse 插件开发学习笔记 [M]. 北京: 电子工业出版社, 2008.
- [10] 韩冰. XML 技术与图形化编程技术的结合 [J]. 辽宁师专学报, 2008, 10 (1): 33-34.

[编辑: 周昱晨]

本文引用格式:

李煜卉, 彭勇刚. 工业机器人通用图形化编程软件的研究 [J]. 机电工程, 2016, 33 (4): 502-506.

LI Yu-hui, PENG Yong-gang. Implementation of graphical programming software for industrial robot [J]. Journal of Mechanical & Electrical Engineering, 2016, 33 (4): 502-506.

《机电工程》杂志: <http://www.meem.com.cn>