文章编号: 1002-0446(2006) 06-0617-06

机器人图标化编程环境的设计及实现*

鲍贤捷[1], 陈 卫东[1], 曹其新[2]
(1. 上海交通大学自动化系, 上海 200030; 2 上海交通大学机器人研究所, 上海 200030)

摘 要: 为解决传统机器人文本编程不易于学习和使用的问题,设计了面向机器人的图标化简便编程环境. 重点研究了图标化机器人程序语言的设计及其解释执行技术. 通过图标化编程环境在工业机器人上的实现和应用, 说明图标化编程技术相对于传统文本式编程技术的易用性和高效性.

关键词: 机器人编程环境; 工业机器人; 图标化编程; 面向对象编程
中图分类号: TP24 文献标识码: B

An Icon-Based Robot Programming Environment Design and Implementation

BAO Xian-jie[1], CHEN Wei-dong[1], CAO Qi-xin[2]
(1. *Department of Automation, Shanghai Jiaotong University, Shanghai 200030, China;*
2. *Research Institute of Robotics, Shanghai Jiaotong University, Shanghai 200030, China*)

Abstract In order to overcome the difficulties in the learning and using of traditional text-based robot programming method, a simple and convenient icon-based robot programming method is designed. The emphasis is put on icon-based robot language design and its interpretive execution algorithm. The icon-based robot programming environment has been applied to and implemented in industrial robots, and the results show that this method is easy to use and highly efficient.

Keywords robot programming environment; industrial robot; icon-based programming; object-oriented programming (OOP)

1 引言 (Introduction)

随着机器人技术的发展, 机器人的应用领域已经从传统的工业领域中的应用, 扩展到教育和服务领域中的应用. 在这些新应用中, 机器人的使用者往往并不具备编写机器人程序所需的专业知识, 如何为这些机器人用户提供高效、易用的编程环境成为近年来机器人研究的重点内容之一[1~3]. 如果选取人机交互方式为对机器人编程环境进行分类的标准[1,2]. 机器人编程环境可以分为如下几类: 基于文本程序的编程环境, 图标化编程环境, 示教编程环境, 基于 3D 仿真环境的自动编程环境[4~7]和多模交互编程环境[1,2].

其中, 面向机器人的图标化编程技术是一种离线编程技术, 具有直观、交互性强等优点, 特别适用于教育和服务领域, 在工业领域也不失为一种高效的编程交互方式, 近年来在机器人编程的应用中得到了很大发展[4]. 与传统的文本编程方式相比, 图标化机器人编程环境具有程序结构直观、易于理解、交互方式简单等优点. 图形化编程环境的缺点是以牺牲文本编程环境的部分灵活性为代价的, 更适用于应用编程而非系统编程[7~9].

本文首先介绍了图标化编程环境这一软件包的设计, 而后重点研究了机器人图标化程序语言的设计, 及实现图标程序顺序执行的解释执行算法; 还介绍了机器人图标化编程环境在一种工业机器人控制系统中的应用, 并通过一个实际案例说明了图标化机器人编程技术高效、易用的特点.

2 图标化机器人编程环境的设计 (Design of the icon-based robot programming system)

2.1 软件需求和用例分析

针对图标化机器人编程方式的优缺点, 设计了

* 收稿日期: 2005-12-16

机器人图标化编程环境这一软件包. 这一编程平台采用机器人图标程序语言, 设计软件包时的两个技术着眼点是平台通用性和编程交互的简便性. 图标化机器人简便编程平台适用于各种机器人执行器的编程, 并为缺少技术背景的非专业用户提供了高效易用的图标化编程环境.

为了这两个目标, 提出了机器人图标化编程环境软件包的需求:

- 使用面向机器人执行器的图标化程序语言;
- 提供编程、程序运行的监控和信息交互的各个功能界面;
- 利用简单的人机交互手段代替传统文本程序的输入;
- 编程交互简单直观, 使用户能够在短时间内掌握;
- 使用面向对象的高级语言实现, 使之具有良好的可移植性.

与此软件需求对应, 利用 UML语言的用例图方式对图标化机器人简便编程软件的使用需求进行了分析. 图 1中的用例示意图描述了使用者对图标化机器人简便编程平台的操作及其基本功能.

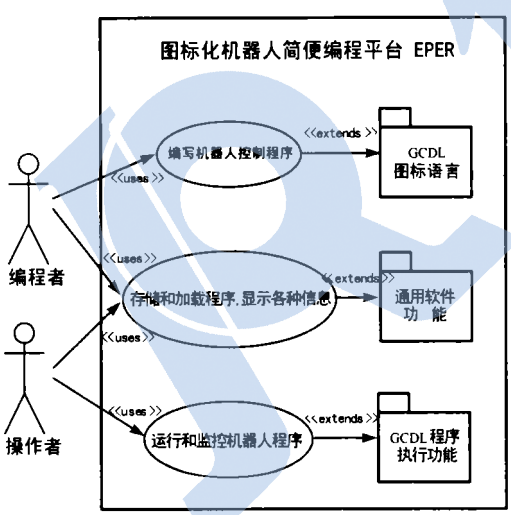

图 1 图标化机器人简便编程环境 EPER的用例示意图

Fig 1 User-case diagram of the icon-based easy programming environment for robot (EPER)

由图 1可见, 面向机器人的图标化编程环境为用户提供了两大基本功能, 分别是图标化语言的编程和程序的执行及监控功能. 图标化机器人语言是这一编程环境的核心功能, 其设计直接影响了用户编程效率和易用性. 以下两节将通过面向机器人的图标编程语言的表示和指令集这两个方面设计这一图标语言.

2.2 图标编程语言的表示

机器人图标编程语言 GCDL(Graphical Control Description Language)是图标化机器人编程语言, 具有两种基本语言元素: 图标(icon)和连线(line). 图标用于表示该语言的各种语句, 而连线则用于表示语句在逻辑上的顺序关系.

机器人图标编程语言采用类似于流程图的程序结构表示程序的逻辑结构. 由于流程图的显示在纵向上较长, 而宽度有限, 与一般计算机的显示区域不匹配, 所以为提高显示效率, 程序逻辑被分为主逻辑和分支逻辑(循环体、条件判断), 分别以垂直和水平方向的连线表示, 在分支逻辑(水平方向)执行结束后自动跳转回到上一级的逻辑图标节点执行. 通过这种改良的流程图结构, 大部分图标程序可以在一个长度和宽度相当的长方形区域中显示, 这种表示方式可以充分使用编程显示区域, 增强程序可读性.

图 2、图 3表示了一段类 C语言的程序及相应图标程序的对比.

```
Main(){
ServoOn;
MovJ(p1);
SetDo(1, Do0);
SetDo(0, Do1);
for(int i= 0; i<= 1; i++)
{
bool var1=
LogicAnd(Di0, Di1)
if(var1)
    MovLCart(p2);
else
    MovLCart(p3);
}
ServoOff
return; }
```

图 2 类 C语言机器人文本程序

Fig 2 Text-based robot programming sample in C-like language

图 2所示是一个类 C语言的机器人文本程序, 图 3是同一个程序的图标语言描述. 在此描述中, 每个图标表示一条指令, 而逻辑关系表示为直线(折线)的连线; 不同的语句以图标名称加以区别. 而连线则以颜色和方向加以区别, 定义主逻辑采用垂直方向的连线表示, 而分支逻辑(循环、条件判断等)采用水平方向的联线表示. 通过这种表示, 图标程序是按垂直和水平方向展开的树型结构, 语句是树的节点, 逻辑关联是这一树型结构的连线.

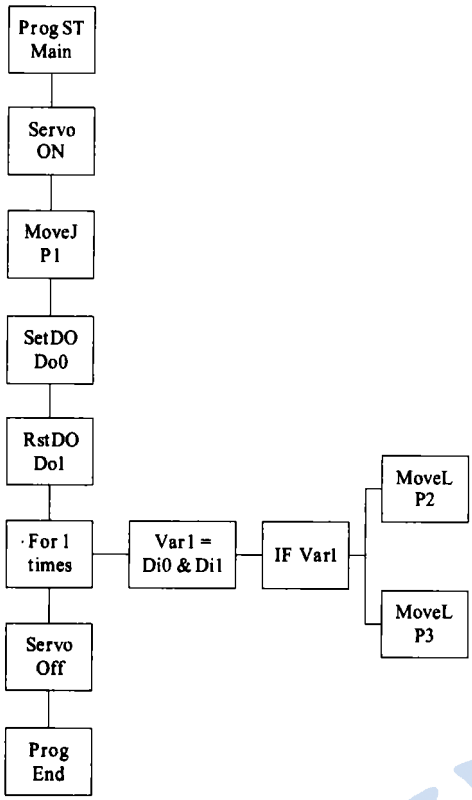

图 3　图标化机器人程序 (对应于图 2的 C程序)

Fig 3　Sample of icon-based robot program (corresponding to Fig 2)

计算机语言的研究表明, 所有的复杂程序逻辑都可以用顺序、判断分支和循环这 3种基本逻辑的排列和组合表示. 在机器人图标编程语言中, 连线表示的是顺序逻辑, 而循环和判断分支分别具有各自的图标表示, 面向机器人的图标编程语言支持 for及 do- while循环, 和 if- else条件分支, 只有通过这 3种图标才能产生分支逻辑和分支结构的显示.

2 3　图标指令集

机器人图标编程语言 GCDL是一种面向执行器的动作级语言, 其指令集的设计必须平衡易用性和功能性这一矛盾, 通过对执行器的基本操作进行分析和简化, 根据这些操作是连续的还是离散的, 将其分为两类: 执行器运动 (连续)和 IO运算及操作 (离散). 再加上程序逻辑控制, 机器人图标编程语言包含了 3类基本图标 (指令), 分别是程序逻辑图标、机器人动作图标和 IO操作及运算图标.

表 1列出了这 3类的所有图标及其功能简介. 如表 1所示, 机器人图标编程语言 GCDL涵盖了程序逻辑控制、执行器动作、IO和变量操作这四方面, 通过这些逻辑和操作的组合, 这一指令集所涵盖的功能与一般工业机器人的指令集相近, 如: 安川公司的 Inform 机器人编程语言等.

表 1　面向机器人的图标编程语言的指令集

Table 1　Instruction set of the icon-based robot programming language

类型	名称	功能
程序逻辑	Prog ST	程序开始, 入口点
	Prog End	程序结束, 终止点
	If- Else	逻辑分支
	For	For循环
	Do- While	Do- while循环
机器人动作	Servo On	伺服启动
	Servo Off	伺服关闭
	Set Speed	设定运动速度
	Get Joint	读取当前各轴位置
	Move J	PTP运动
	Move L	Linear运动 (轴坐标)
	Move L(Cart)	Linear运动 (笛卡儿)
	Move Ic(Cart)	增量运动 (笛卡儿)
	Move with Vision	视觉标定运动
IO操作及变量运算	Grip Open	手爪开启
	Grip Close	手爪关闭
	IO Read	读取 IO至变量
	IO Write	从变量写入 IO
	Set Var	设置变量值
	Logic Op	变量逻辑运算
	Arith Op	变量算术运算

3　图标程序的解释执行方法 (Interpretive execution method of icon-based program)

机器人图标程序的执行控制是图标化编程环境实施的关键技术. 通过对编译执行和解释执行两种执行方式的对比, 在机器人图标化编程环境软件包的实现中, 选择了解释执行机制. 解释执行的优点是不生成中间代码, 程序可以直接执行, 避免了编译的开销, 而且解释器的开发和维护成本较低. 为此, 开发了图标程序解释器, 用于实现机器人图标程序的执行控制. 这一解释器有两方面的功能, 其一是执行各个图标的具体功能, 如: 机器人运动和 IO操作等; 其二是依据图标程序的拓扑结构, 控制各图标的执行顺序, 本节将重点讨论图标程序的顺序执行控制的实现技术. 以图 3中的图标化机器人程序为例, 将这一程序的执行顺序在图 4中以箭头表示, 其执行

顺序如 1到 12步所示, 6到 10步骤间循环 n 次.

通过对这一执行顺序的解析, 可以看出, 只有在循环 (loop) 和条件判断 (ifelse) 等分支图标上时, 图标程序的执行顺序有跳转; 其它图标都只有一个主、分支逻辑上的前驱图标和一个主、分支逻辑上的后继图标, 顺序执行. 同时, 由于图标程序的语法支持循环等分支图标的嵌套, 当执行到分支图标或者分支结束时, 可能会进行一次或多次的回归, 跳转到上一级的分支图标继续运行. 由于没有编译过程, 解释器不能预先知道程序的拓扑结构, 那么图标程序解释器必须按照一定的搜索算法对这一以图标为节点的树进行遍历.

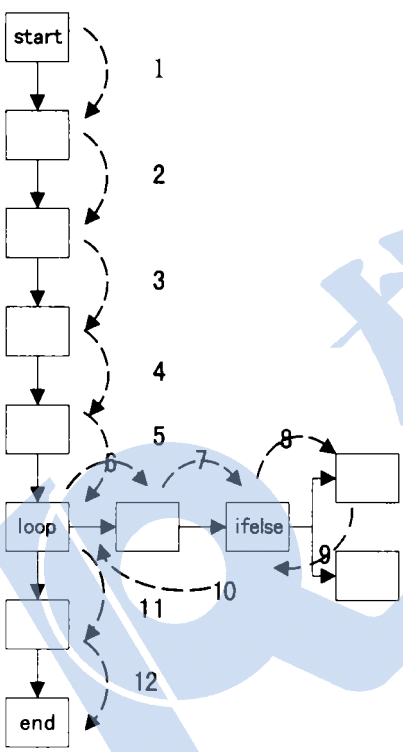

图 4 一个图标程序的执行顺序图

Fig. 4 Execution sequence diagram of an icon-based program

由于这一图标程序的解释 (遍历) 算法中包含回归运算, 因此必须对产生分支的图标予以保存, 采用栈 (stack) 结构保存分支图标, 符合其先进后出 (FILO) 的回归顺序. 可以采用图 5的形式描述这一解释算法.

这一遍历算法包含 6步:

• 初始步骤: 图标程序从开始图标开始执行, 跳至步骤一;

• 步骤一: 根据执行情况, 加载后继图标, 如果后继为程序结束图标, 则跳至终止状态; 如果后继非空, 则跳至步骤三; 如果后继为空, 则跳至步骤二;

• 步骤二: 将图标出栈, 跳至步骤三;

• 步骤三: 执行图标, 如果当前为分支图标, 则跳至步骤四; 否则, 跳至步骤一;

• 步骤四: 将分支图标入栈, 跳至步骤一;

• 终止步骤: 图标程序停止运行.

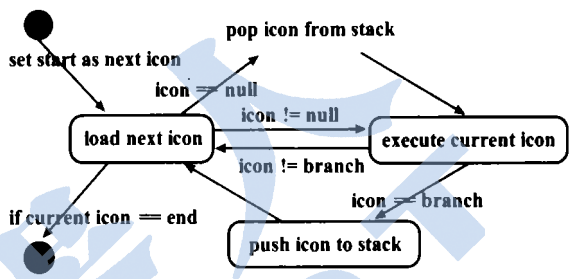

图 5 图标程序的解释算法的状态描述

Fig. 5 State description of icon-based program interpretation

特别需要指出, 在这一算法中, 分支图标不仅可以在分支条件满足时加载其后继分支图标, 而且分支图标在分支条件不满足 (如: 循环条件不满足) 时, 将加载其后继的主逻辑图标 (可以为空), 正是这一机制, 保证了回归运算的有效性.

使用这一图标程序的遍历算法, 可以实现面向机器人的图标化程序执行过程的顺序控制, 避免了程序逻辑控制的集中实现; 集中实现将不可避免地产生大量的判断 (分支的嵌套数按几何级数增长) 代码, 以分析程序的拓扑结构, 不利于图标程序的功能扩展和维护.

在面向机器人的图标化编程环境中, 这一解释器可以由各种类 C语言或者 OOP语言实现, 这种改良的遍历算法还可以用于其他具有类似拓扑结构的图标程序执行过程的顺序控制.

4 实例研究 (Case study)

采用 MotoMan-UPJ机器人作为编程对象开展实例研究. 与一般的工业机器人控制系统相比, UPJ机器人采用了 RealTime-Linux /RedHat-Linux 的组合平台作为其控制器的操作系统, UPJ机器人的实时控制软件 RTLab Functions采用 C语言实现, 运行于实时 (RTLinux Kernel) 端, 并且在非实时的 RHLinux 环境中为用户提供了调用接口 RTLab API, 机器人用户可以编写 C程序, 调用这一接口, 实现对 UPJ机器人的控制.

本文以该工业机器人为对象, 开发了一套图标

化编程环境 EPER. 采用了 Java作为图标化编程环境的开发环境, 机器人图标化编程环境软件及 UPJ机器人控制系统的框架见图 6

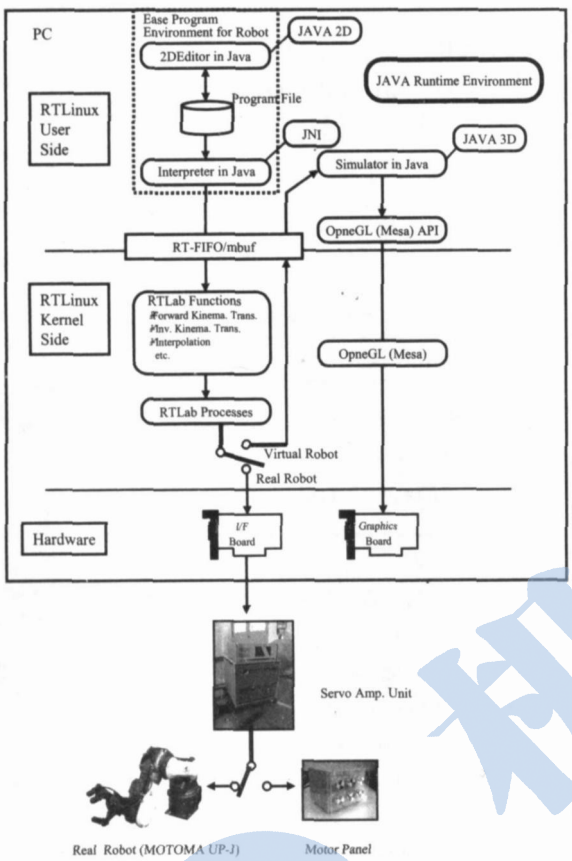

图 6　机器人图标化编程环境的实施框架及其与工业机器人控制系统的关系

Fig. 6　Implementation structure of EPER and its relation with robot control system

通过引入图标化编程技术及其实现技术, 为 UPJ机器人的使用者提供了基于图标语言 GCDL的编程手段, 替代了该机器人原有的文本编程模式, 缩短了用户学习机器人编程的时间, 并为这些用户提供了高效的图标化编程手段.

机器人图标化编程环境软件包 EPER采用 Java语言作为其开发环境, 从实现技术上保证了良好的扩展性和移植性. 图标化编程环境与 RTLab API的底层接口采用 JNI技术实现.

在用户界面的设计上, 根据 2.1节中对图标化编程环境的用例分析及机器人图标语言的特点, 采用了多窗口的界面风格, 其图形化用户界面 (GUI)包含 3个基本窗口界面 (如图 7所示), 分别是图标化编程窗口、程序执行控制窗口和信息显示窗口. 在对编程操作的考虑上, 也做了相应的优化, 基本的编程操作使用鼠标即可完成, 只有在输入数据时, 用户需要使用键盘, 交互方式非常直观, 可以降低用户学习掌握图标化编程的时间开销.

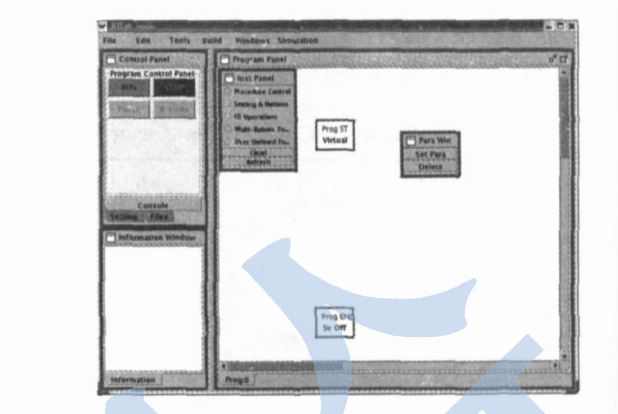

图 7　图标化编程环境的用户界面

Fig. 7　Graphical user interface of EPER

在面向机器人的图标化编程环境软件包的软件结构设计中, 广泛地采用了 OOP的设计方法, 由于篇幅所限, 本文对此内容不作深入探讨.

5　机器人图标编程和文本编程的对比分析 (A comparison analysis between icon-based robot programming and text-based robot programming)

在机器人图标编程环境中, 使用图标语言编写了一个典型机器人应用的工程例程. 任务是让机器人从工作台上抓取工件, 如果抓手中握有工件, 机器人就将其放置到传送带, 如果没有握有工件, 那么机器人执行器会回到工作台位置重新尝试抓取工件. 机器人执行器在工作台和传送带之间作门字型往复运动. 如图 8所示, 这一程序包含约 20个图标.

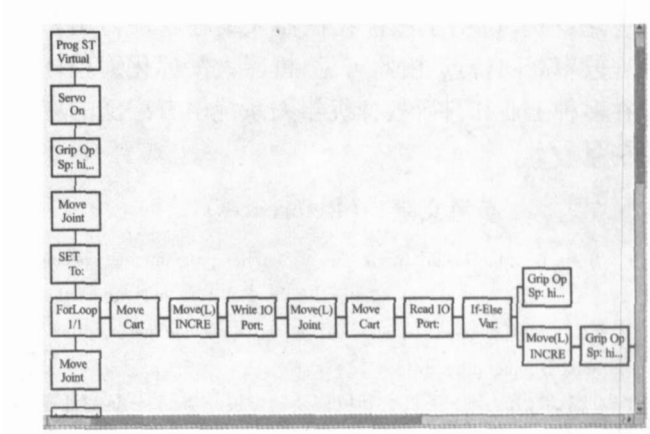

图 8　利用图标语言实现工件搬运任务

Fig. 8　An icon-based program for material handling application

为了方便进行对比, 采用了文本编程方式实现了同一个应用, 文本编程采用 C 语言编程, 通过调用控制器的 API所提供的函数接口, 实现对机器人运动的控制. 表 2列出对图标和文本两种实现方式的对比.

表 2 图标化程序和文本程序的对比

Table 2 Comparison of icon-based robot program and text-based robot program

	利用图标语言编写	利用文本语言编写
代码量	约 20个图标	约 200行代码
输入方式	鼠标为主, 图标参数采用键盘输入	键盘输入
编程耗时	约 30min	约 3h
执行所需处理	无预处理, 直接执行	需要 GCC编译, 编译参数设定较复杂
运行期程序调试	动态显示程序指针, 动态修改程序数据	无调试环境
程序运行时机器人运动的紧急停止	图标编程环境支持程序的暂停或者紧急终止; 或使用硬件急停	终止整个程序运行, 或使用硬件急停
程序的保存和加载	特殊编码方式保存, 只能在图标编程环境中再现和修改	文本格式, 可以使用文本编辑器修改

通过以上对编程开销等的对比, 可以看出对于同一个应用, 图标编程代码量少, 易于编写, 而且具有较好的运行和调试环境, 编程效率高于文本编程. 而且, 对于图标编程, 编程者只需了解 UPJ机器人的基本操作特性, 经过半天左右时间的图标编程培训即可, 无需文本编程所要求的 C 语言的背景知识, 也不必掌握 RTLab API接口函数库. 无疑, 图标编程对于非专业的机器人使用者更为友好、高效.

6 结论 (Conclusion)

本文介绍了机器人图标编程环境软件包的设计和实现情况, 并着重探讨了图标化机器人程序语言的设计及图标程序的解释执行技术. 本文还通过图标化编程软件在工业机器人系统中的实现和一个编程应用案例, 说明了图标化机器人编程软件易用和编程效率高的特点. 由此可见, 机器人图标化编程环境在多种工业和科研教育机器人系统中有广泛的应用前景.

参考文献 (References)

[1] Biggs G, MacDonald B. A survey of robot programming systems [A]. The Australasian Conference on Robotics and Automation [C]. http://www.ele.auckland.ac.nz/~macdon/general/files/acra03-gblm.pdf, 2003.

[2] 马卫娟, 方志刚. 人机交互风格及其发展趋势 [J]. 航空计算技术, 1999, 29(3): 16-20.

[3] 戴齐, 姚先启. 机器人程序设计语言 [J]. 机器人, 1997, 19(5): 390-400.

[4] 赵东波, 熊有伦. 机器人离线编程系统的研究 [J]. 机器人, 1997, 19(4): 314-321.

[5] Lauria S, Bugmann G, Kyriacou T, *et al*. Training personal robots using natural language [J]. IEEE Intelligent Systems and Their Applications, 2001, 16(5): 38-45.

[6] Wong R K. Advanced object-oriented techniques for modeling robotic systems [A]. Proceedings of the IEEE International Conference on Robotics and Automation [C]. Piscataway, NJ, USA: IEEE, 1995. 1099-1104.

[7] Bischoff R, Kazi A, Seyfarth M. The MORPHA style guide for icon-based programming [A]. Proceedings of the IEEE International Workshop on Robot and Human Interactive Communication [C]. New York, NY, USA: IEEE, 2002. 482-487.

[8] Cox P T, Smedley T. Using visual programming to extend the power of spreadsheet computation [J]. Proceedings of the Workshop on Advanced Visual Interfaces [C]. New York, NY, USA: ACM, 1994. 153-161.

[9] 李瑞峰, 吕开元. 基于图形编程技术的服务机器人人机交互系统的研究 [J]. 制造业自动化, 2003, 25(3): 40-43.

作者简介:

鲍贤捷 (1978-), 男, 硕士生. 研究领域: 机器人图标化编程技术.

陈卫东 (1968-), 男, 教授. 研究领域: 移动机器人, 多机器人学, 机械臂的控制与规划.

曹其新 (1960-), 男, 教授. 研究领域: 工业机器人, 机器视觉, 智能控制.