第33卷 第19期 计 算 机 工 程 2007年10月
Vol.33 No.19 Computer Engineering October 2007
·软件技术与数据库· 文章编号：1000—3428(2007)19—0079—03 文献标识码：A 中图分类号：TP24

# 基于组件的网络移动机器人软件框架

刘 哲，尹怡欣，殷 路
(北京科技大学信息工程学院，北京 100083)

**摘 要：**在研究移动机器人自身特点及网络控制特性的基础上，提出一种基于组件的移动机器人程序框架，减少了编写移动机器人应用程序时的复杂性并提高了代码的复用性。通过构建统一的网络平台和控制平台，制定一组规范的模块抽象类和接口，实现组件的动态加载和连接。通过优先值控制，灵活地调整对机器人的控制方式和网络控制模式。结合中科院智能机器人平台AIM，说明如何使用Java和XML构建该框架。

**关键词：**组件；程序框架；移动机器人；Java；XML

# Component-based Software Framework for Network Mobile Robot

LIU Zhe, YIN Yi-xin, YIN Lu
(School of Information Engineering, University of Science and Technology Beijing, Beijing 100083)

**【Abstract】** This paper presents a component-based framework of mobile robot. This framework can reduce the complexity in application building, and improve code reuse. By developing a uniform control and network platform and establishing a group of abstract class and interface, the framework component can be dynamically loaded and linked. By changing component, the robot can be flexibly adjusted to work under different control manners and network control modes. An example illustrates how Java and XML techniques are used in the frame.

**【Key words】** component; software framework; mobile robot; Java; XML

随着台式计算机的发展及互联网的不断扩大，人们找到了能极大提高机器人利用率的途径，即将互联网和机器人技术结合起来，开展机器人远程操作与实验。

但移动机器人是一个复杂的系统，大多数基于网络的移动机器人的研究，只针对于网络延迟、控制方式等细节问题，缺乏专门对基于网络的移动机器人程序在整体框架和设计实现上的探讨。此外，移动机器人控制软件涉及技术面多，彼此间关联度小，使得设计和实现一个完整的基于网络的移动机器人控制系统仍是一个很大的挑战。

本文从互联网应用的特点入手，针对机器人的网络应用设计了一个软件框架，实现了此类应用程序之间的共同点，降低了其构建代价，提高了代码的复用性。

## 1 框架的设计

框架是指在一个给定的领域内一个应用程序的部分设计和实现[1]。因此，框架在设计的过程中，必须针对该领域的特点，把握领域内相似应用程序的结构，为运行一批对象提供一个有组织的环境。框架的设计必须充分考虑基于网络的移动机器人所具有的特点。

### 1.1 网络移动机器人特点

(1)基于网络的移动机器人所面临的首要问题是选择通信方式和协议。但网络延迟和数据包的丢失给系统的性能、安全性和鲁棒性带来的影响也不容忽视。

(2)网络移动机器人的控制模式对系统功能的实现起决定性作用。因此，在框架设计中必须有效地利用直接控制、监督控制和共享策略控制3种基本类型[2]，并对其做出合理的演变。

(3)由于移动机器人本体必须有一定的自主能力及自我保护能力，因此为防止因为误操作或网络延迟引起的硬件损坏，就需要解决好各个智能模块之间的协调和通信关系。

### 1.2 基本框架

基于上述网络移动机器人的特点，本文将框架分成3个部分：网络平台，控制平台及控制模式模块。结构如图1所示。

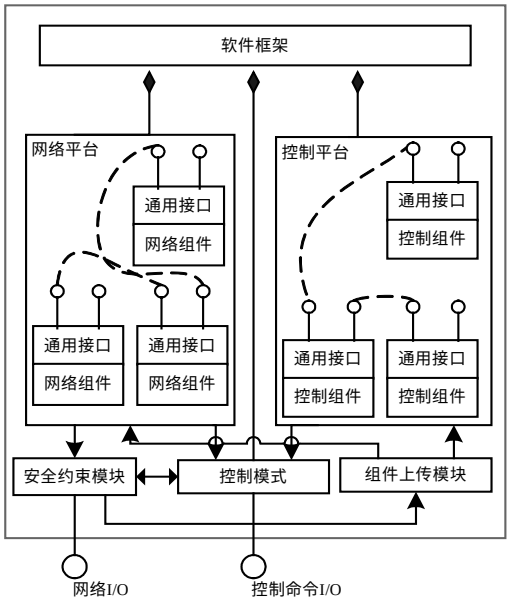


**图1 系统框架**

**基金项目：**国家自然科学基金资助项目(60374032)
**作者简介：**刘 哲(1981－)，男，硕士研究生，主研方向：智能移动机器人；尹怡欣，教授；殷 路，博士研究生
**收稿日期：**2006-10-10 **E-mail：**hieddy@gmail.com

(1)网络平台、控制平台负责各自组件的加载、初始化以及组件之间、组件和平台之间的通信建立。它们分别是网络部分和本地控制部分的应用程序的入口和出口。但机器人功能的实现和网络部分的连接是由各自平台内的组件实现的，而非直接通过平台。

(2)网络组件和控制组件定义了通用接口和组件的基本行为。组件在两个平台初始化的时候被装入，通过通用接口，由网络状况和用户的意愿动态决定彼此间连接的顺序。

(3)安全和约束模块负责权限和网络速度的动态测试。其中，约束模块定时检测网络的速度并反馈至控制模式模块，以决定系统在何种策略下工作；上传模块在用户具有权限的情况下，允许其将自己编写的组件和系统的初始化参数文件上传至机器人的控制平台和网络平台，使用户即使在异地，也完全可以按照自己的意愿对机器人进行控制或者在机器人平台上试验自己的算法。

软件框架分为白盒框架和黑盒框架[3]2 大类。虽然白盒框架的复用成本稍高于黑盒框架，但是黑箱框架内部的机制被开发者隐藏了，缺少灵活性，并不适合移动机器人的复杂平台性。采用白盒框架的结构，网络组件和控制组件通过实现一组抽象类和接口，以及相应的输出输入类来获取最大的兼容性。因此，在此架构下，可以动态地加载组件，最终实现各种组件的即插即用。该软件框架允许使用者方便地分解问题、减少开发和维护代码的成本。

### 1.3 组件设计及连接方式

组件实现的基本操作是根据输入的数据，然后从输出端口把数据输送出去。其中，数据处理的过程可能很复杂，例如需要进行图像识别、障碍物检测或延迟优化算法等。组件设计的 UML 如图 2 所示。

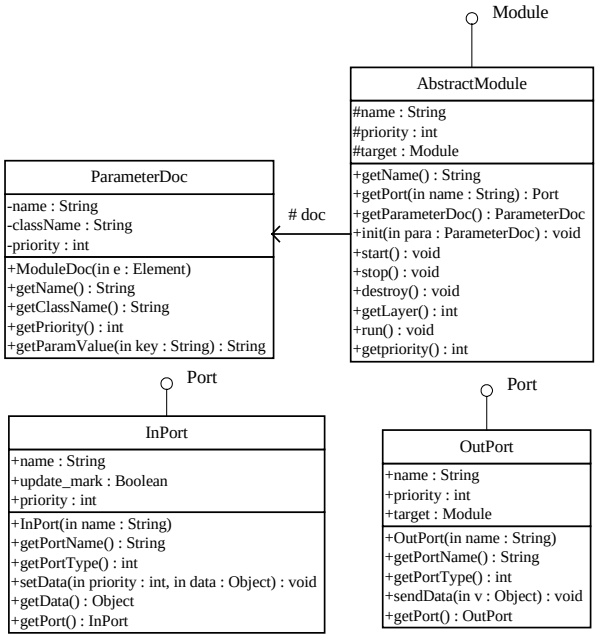


**图 2 组件部分 UML**

Module 接口定义了组件必须实现的接口，并且在抽象类 AbstractModule 中实现了部分通用方法。所有组件的核心都是从 AbstractModule 继承得到类。

ParameterDoc 是对配置文件进行操作的类。在组件初始化时，该操作类从配置文件里读取模块的各种设置参数，如通信端口号、客户机的地址、硬件通信的串口位等。

组件将输入输出部分分离出来，提供基本的输入输出类。输入输出类都实现了 Port 接口，在组件初始化时读取配置文件，根据配置文件中指定的连接关系实现组件间的动态连接。

由于不同组件可能会同时对机器人发出控制指令，或者不同的传感器组件可能同时得到互相抵触的信息，因此本文采取Brooks提出的基于行为的包容体系结构[4]保证组件输入输出间的连接，该结构采用分布决策模式取代了传统的中心决策模式设计控制器，提高了移动机器人的机动能力和鲁棒性。在图 3 中，下标越小其输出优先级越高。当存在有意义的输出，即优先级高的组件处于激活状态时，即使低优先级的组件输出也处于激活状态，它也将被抑制。只有当优先级高的组件不在激活状态时，优先级低的行为才取得控制权。具体模块的优先级，可以在配置文件中指定，也可以在编写模块时就硬编码到模块中去。

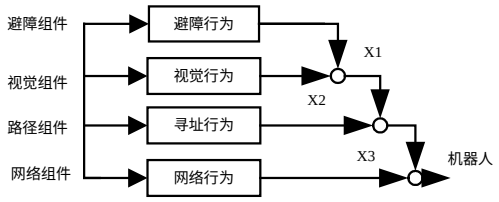


**图 3 组件优先级工作示意图**

## 2 框架的实现

第 1 节中的框架在中科院自动化所高创中心开发的智能机器人平台 AIM 上，采取 Java 语言进行了具体的实现。

### 2.1 基于 XML 的配置文件

可扩展的标识语言(extensible markup language, XML)采用自描述性的中立数据为结构，可以表示复杂的数据并使其可读。在该框架中，XML文档被用来作为平台的配置文件并储存了各组件的初始信息和参数。两个平台启动时都会从XML文件读取需要加载的模块的信息和参数，并动态地建立起连接。在读取XML文件中的配置时，由于配置文件很小，因此可采用DOM分析器来分析文件，一次性将文件读入 内存[5]。

通过分析器对配置文件的分析，将获得的元素作为引用传递给专门储存参数信息的 ParameterDoc 类。利用该类已定义好的方法，平台只需要调用这些方法即可获得要加载类的各种信息，如储存位置和连接对象等。读取配置的 XML 文件获取信息的代码如下(有删减)：

```
001: Reader r = new FileReader(file);
002: DocumentBuilderFactory factory
= DocumentBuilderFactory.newInstance();
003: DocumentBuilder builder
= factory.newDocumentBuilder();
004: Document xmldoc = builder.build(r);
005: Element root = doc.getRootElement();
006: robot_name = root.getAttributeValue("name");
007:            if(root.getName().equals("config"))
008:    {List ele_list = root.getChildren("component");
009:            for(int i=0; i<ele_list.size(); i++)
010:                    {Element e = (Element) ele_list.get(i);
011:                    ParameterDoc para= new ParameterDoc(e)}}
```

### 2.2 平台组件的动态加载和连接

在获得具体的组件信息及其参数后，利用 Java 的反射机制，可以实现对这些组件类的动态加载。由于在加载过程中必须把配置文件中给定的参数，如优先级、传递给组件的输入输出的连接方式，因此在加载时必须使用带参数的方式。因此，不能直接调用 Class 的 newInstance( )，而应调用 Constructor 的 newlnstance( )。

代码片断如下：

```
001: class c = class.forName("dodge");
002: class[ ] pTypes
= newClass[ ] {int.class, Abstractmodule.class};
003: Constructor ctor = c.getConstructor(pTypes);
004: object obj = null;
005: object[ ] arg = new object[ ]{3,control};
006: obj = ctor.newInstance(arg);
```

在组件初始化后，组建输出端口的连接对象作为参数已被传递入组件内。因此，在需要输出信息时，通过直接调用目标组件的 setData()，可将信息写入目标组件。但信息输入方式是被动的，即在每次有新的信息输入后，组件内的标志位立即会被改动，通知组件有新的信息进入。这样，各个组件在装载时就被动态地连接了。系统组件的最终连接状态如图 4。

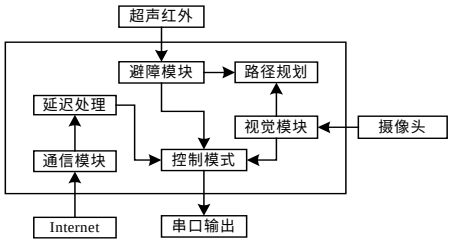


**图 4　组件连接方式**

### 2.3　控制模式模块的实现

控制模式模块决定机器人在何种策略下工作，即在收到远程控制的请求后，系统自动根据数据的丢包和延迟情况，决定用户能决定的机器人工作的方式。当二者之间的网络延迟小于 100ms 时，用户可以选取全部的 3 种控制方式；当网络延迟在 100ms~300ms 之间时，用户不可以使用直接控制方式；当网络延迟大于 300ms 时，用户只可以选择监督控制。

(1)系统每 5min 会重新测量一次网络速度，根据最新的延迟结果重新规定用户可以选择的模式。控制模式通过调整各模块的优先级来实现，系统的优先值规则如下：

1)所有来自网络平台的控制默认优先级为 1。

2)在监督控制、直接控制、共享控制策略条件下对网络控制命令的附加优先级分别为-1, 0, 1。网络平台最终的输出优先级为默认值和附加优先级之和。

3)本地的避障模块具有最高的优先级 3，视觉追踪和路径规划优先级为 1。

(2)控制模式模块负责接收来自两个平台对机器人的控制命令。为了使平台具有通用性，所有输出控制命令的组件的输出都定义为标准的控制字符串。控制模式模块接收到控制字符串后，转化为对应的控制命令，从串口发送至下位机，通过 DSP 对移动机器人电机进行控制。针对不同的硬件，只要对控制模式模块内的命令转化部分作对应修改即可，不需要修改其他组件。

### 2.4　网络通信方式及组件上传

在网络通信组件中，通过 Socket 接口，在客户端和服务器端之间建立起基于 TCP 的连接。由于网络延迟的不确定性，为防止因为网络阻塞使服务器端在很短的间隔内连续收到控制命令而影响控制效果或损坏机器人电机，因此应在通信模块后连接一个延迟处理的模块，并在该模块内建立一个根据网络延迟动态的初始化大小的缓冲区(buffer)，以充分保证数据传输的连续性。用户通过 Internet 发出命令，该命令首先由机器人端的通信组件接收，然后传送到延迟组件内，当缓冲区充满后就会把处理后的数据以相同的时间间隔传给机器人，同时控制端发出的命令又会源源不断地通过通信组件存储到缓冲区中。通过在缓冲器中对时间延迟的处理，实现了相同时间延迟的远程机器人控制。

上传组件部件最常用的是 FTP 方式，用户在选择组件和配置文件对应的平台后，自动使用不同的用户名登录，使文件自动上传到对应的目录。这样，平台在加载时自动搜索对应的目录就可以得到用户上传的配置文件和组件。

## 3　结论

基于组件的移动机器人程序框架可帮助编程者方便地实现移动机器人的控制。将该框架应用到现有的移动机器人平台的程序中，可将避障、视觉追踪和路径规划独立出来交由专人编写，从而提高工作小组效率，使得研究人员可以专注于其研究方向。通过结合 Java 和 XML，该框架能够在各种系统下顺利运行，具有较强的通用性。

随着组件数量的不断增加，将会使彼此间的连接复杂度不断加深，平台缺乏一个对组件进行有效管理的机制。同时在组件编写时，必须考虑到连接对象，将传递信息的向下转型硬编码在组件内，这些问题都是今后本课题后续研究的范畴。

## 参考文献


1 Bosch J, Molin P, Mattsson M, et al. Object-oriented Frameworks-problems and Experiences[EB/OL]. (2004-10). http://www.citeseer.ist.psu.edu/bosch97objectoriented.html.

2 郭亚宁, 陈一民. 基于 Web 的机器人遥操作[J]. 计算机工程, 2003, 29(18): 154.

3 Johnson R E, Foote B. Designing Reusable Classes[J]. Journal of Object-oriented Programming, 1988, 1(2): 22-35.

4 Brooks R A. A Robust Layered Control System for a Mobile Robot[J]. IEEE Journal of Robotics and Automation, 1986, 2(1): 14-23.

5 Horstmann C S, Cornell G. Core Java 2 Volume II Advanced Features 5E[M]. Beijing: China Machine Press, 2005.