

# 分布式自主移动机器人集成环境\*

吴春明 张友军 朱森良

(浙江大学人工智能研究所 杭州 310027)

**摘要** 本文详细介绍了一个基于异质分布式计算机系统的机器人集成环境——Robix,它提供用户简单的通讯模式,可以很方便地实现进程间通讯,同时提供对实时系统十分重要的分布式中断,Robix 已在一个室外自主移动机器人系统 ATB1 的集成中得到应用。

**关键词** 分布式共享内存,分布式中断,异质系统,自主移动机器人。

**中图法分类号** TP242.6

自主移动机器人是一个高智能、多系统的复杂工程系统,内部包括多个功能特性各异的功能模块,有实时控制部件、多种信息处理部件、通讯部件等等。要建立这样一个系统,一定要考虑到多个部件集成中可能出现的情况,并且要解决好处理器之间的通讯问题。要成功地将这些功能模块组织在一起,一方面需要一个协调各个功能模块的总体协调模块<sup>[1]</sup>,另一方面需要一个良好的集成环境。一个较好的集成环境需要满足以下几点:① 图形界面及“舒适”的开发(编程)环境;② 模块化、可扩充的结构;③ 可靠的实时响应能力。④ 提供对网络资源访问透明的分布环境。

有的机器人系统使用了一些较专门的用于并行开发环境进行开发。比如为了满足实时处理的需要,选用实时 OS(严格的时间分配),有的系统为了集成环境的统一采用 UNIX 系统。集成时一般采用一个同应用紧密相关的(专用于应用)集成环境,对于新的应用需作改进。本文提出了一种用于分布式机器人系统的集成环境,主要集成机器人系统中的 3 个部分:规划、控制、传感器。本环境可以灵活地将以上几个部分“粘合”在一起构成一个功能较强的机器人系统,研究和改进可由各个模块单独进行,新的算法和策略能较容易地加入到系统中,目的是要建立一个可扩充的、具有一定编程模式的分布式开发环境。

## 1 Robix 的功能设计

在基于分布式计算机系统的移动机器人系统中,存在着大量进程间的通讯,我们看一个典型的机器人系统实例。如图 1,二维模块、三维模块、融合模块、规划模块都是独立的进程,

\* 本文研究得到国防科工委“八五”重大预研项目资金资助。作者吴春明,1967年生,博士,主要研究领域为人工智能,智能机器人体系结构。张友军,1970年生,博士生,主要研究领域为人工智能,分布式系统,智能机器人体系结构。朱森良,1946年生,教授,博士生导师,主要研究领域为人工智能,计算机视觉,智能机器人体系结构。

本文通讯联系人:吴春明,杭州 310027,浙江大学人工智能研究所

本文 1996-06-09 收到修改稿

可以分布在不同的机器上,融合模块需要三维和二维模块的处理结果才能进行处理,并将处理结果送给规划模块,它们之间的数据交换是通过进程间通讯实现的。

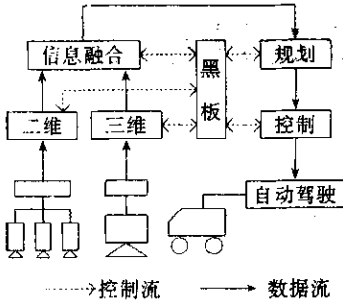


图1 自主式移动机器人结构图

为了支持进程间通讯,有的机器人系统采用统一的数据结构实现功能模块之间的通讯<sup>[2]</sup>(进程间通讯),它的特点是简单易行,但当系统中硬件复杂、处理机差异很大时无法工作。也有的系统提供多种通讯模式<sup>[3]</sup>,但需要用户考虑大量的通讯协议及异种机之间的数据格式转换,使用户代码复杂化,从而增加了系统集成的难度。我们设计了一个称为 Robix 的分布式运行环境,力图综合上述两种系统的特点,提供用户简单的通讯模式,同时又能提供较多的功能。

Robix 通过一块虚拟的分布式共享内存区支持功能模块进程间通讯服务,功能模块直接从共享内存中进行数据的发送或接收。而实际上数据通讯可能是在两台计算机之间产生的。在 Robix 中,每个功能模块有一个唯一的名称标志,如图 1 所示,融合模块就可用“Fusion”来标志,在通讯中进程用这个名称标识,不必关心该进程所在计算机网络上的地址等具体信息。用户也不必关心数据通讯具体是在哪两台计算机之间进行的。

Robix 提供给功能模块的函数主要有以下几个:① Rob\_init()——初始化过程,获得一个访问共享内存的指针。② Rob\_send\_data()——向另外的功能模块发送数据。③ Rob\_get\_data()——读取别的模块发送的数据。④ Rob\_event()——产生一个异步事件。⑤ Rob\_get\_event()——获得一个异步事件的指针。⑥ Rob\_broadcast()——向所有模块广播。⑦ Rob\_signal\_handler()——将处理分布式中断的过程在 Robix 中声明。⑧ Rob\_signal()——产生一个分布式中断。⑨ Rob\_exit()——通知服务进程功能模块退出系统。

总的来说,提供用户两类用于通讯的函数,一种是一般的数据(Data)传送和简单的事件命令(Event);另一种是分布式中断。

## 2 Robix 的实现技术

### 2.1 Robix 的内部工作过程

机器人系统中的功能模块通过分布式共享内存进行通讯,并不直接同其它的设备打交道。实际上,不同机器上的通讯也是通过以太网进行传送的,具体的传送过程由 Robix 中的服务进程实现,Robix 内部的工作过程可以用图 2 表示。从图 2 可以看到,功能模块 B 向 D 发送数据,实际上数据先由 B 所在机器上的服务进程发送到 D 所在机器的服务进程,由服务进程接收后放入共享内存中,然后 D 可以从内存中取走数据。同样 D 向 A 发送中断信号也是在服务进程的协助下实现的。Robix 中与通讯有关的操作是由服务进程完成的。

### 2.2 通讯服务进程

Robix 中的服务进程通过对整个分布式计算机系统资源和通讯的统一管理,为机器人系统中的功能模块提供一个统一的操作运行环境,使得功能模块可以很方便地实现进程间的通讯和同步。

· 共享存储区的管理 Robix 中每个功能模块都是从一块分布式共享内存中存取数据的,这块内存区是功能模块实现信息交换的一个重要枢纽,发送数据或信号的功能模块只要将请求存入内存中即可,接收数据的模块也是从内存中取走数据,接收信号的模块也可从内存中获得信号所带的简短数据.机器人系统中每台计算机上的共享存储区都是由 Robix 的服务进程维护 and 管理的.服务进程将共享存储区中的数据通过网络发送到相应的计算机上.

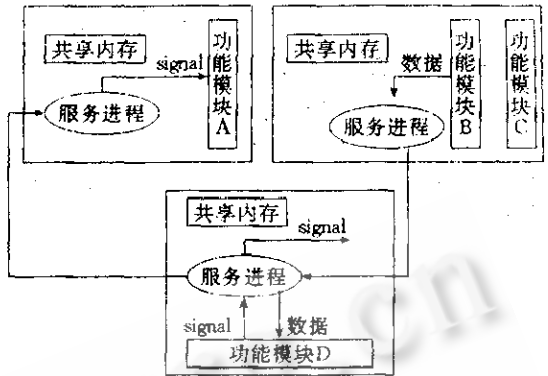


图2 Robix的内部工作过程

· 功能模块的地址管理 Robix 中提供功能模块之间的通讯服务,通讯可能是在两台机器之间进行的,需要通过网络进行传送,而功能模块中并没有这些网络地址信息,但每台机器上的服务进程内部都有一个地址表用以保存这些地址信息.地址信息由 Robix 系统服务进程自动维护、管理.功能模块初始化时,服务进程将该功能模块的地址信息保存下来,并通知其它的处理机上的服务进程.地址信息表的结构如表 1 所示.

表 1 地址信息表

App Name	Pid	Host Address
App Name1	Pid 1	Host Address 1
App Name2	Pid 2	Host Address 2
App Name3	Pid 3	Host Address 3

其中 App Name 为应用程序的名称,用一个字符串标志.在 Robix 系统中,每个应用程序的名称必须是唯一的.Pid 为应用程序的进程号,在一台工作站上唯一地标识一个进程.Host Address 为应用程序所在计算机的网络地址.

· 数据格式转换 异质机之间通讯的另一个问题是机器内部数据格式转换,不同类型的机器由于硬件体系结构不同,数据表示也不一致.以基于 Intel X86 的微机 and SUN 工作站为例. Intel X86 微机和 SUN 工作站在整数的表示方式上,字节顺序正好相反,如果直接传送将造成严重后果.例如,微机上的整数 1 在工作站上被解释成 256,同样长整数和浮点数的表示也不同.

Robix 内部有进行数据格式转换的函数,所有的格式转化都是在 Robix 内部完成的,对用户来说不必担心数据格式的不一致.

· 分布式中断 它允许一个功能模块被一个类似于 UNIX 软中断所中断,发生事件的功能模块可能是另一台处理机上的,并且可能是不同类型的机器.它的主要作用是提供一个中断机制,使系统能对突发紧急事件及时采取相应的措施,具体实现的过程如下:① 接受模块调用 Rob\_signal\_handler(),将处理异步事件的过程在 Robix 中登记,以便于调用.② 发送模块所在机器的服务进程将异步事件信号及相关数据通过以太网发送到接受模块所在的服务进程.③ 接受模块所在计算机的服务进程向对应的中断处理过程发出一个中断信号,并将相应的数据送到该过程.如果接受模块和发送模块在同一台计算机上,则只需要通过该机上的服务进程即可.

由于 Robix 提供了分布式中断这种机制,对提高系统的实时性有重要的作用,一旦出现紧急事件或发生意外情况,通过分布式中断可使机器人及时作出反应.

### 2.3 应用程序的一般模式

下面以融合模块(FUSION)为例,看 Robix 中一个功能模块的程序结构模式.

```

#include<Robix. h>          /* 将机器人系统有关的头文件包括进来 */
#include<Robot. h>
main()
{
    void Fhandler( );
    Event_data event_data;
    int event;

    Rob_init(FUSION);      /* 初始化,获得用于通讯的共享存储区 */
    Rob_signal_handler(Fhandler); /* 将处理突发事件的处理过程向 Robix 登记 */
    /* * * * * * main loop * * * * * */
    while((event=Rob_get_event(&event_data, BB))!=QUIT_EVENT){ /* 获得黑板调度器的调度信号 */
        switch(event){
            case INIT_EVENT:
                fusion_init( ); /* 融合工作前的初始化 */
                Rob_event(BB,FUSION_INIT_OK);
                break;
            case OA_EVENT:
                /* 进行避障情况下的处理,并向黑板发送处理完成的信号 */
                /* * * * Process here * * */
                Rob_event(BB,FUSION_PROCESS_OK);
                break;
            .....
        }
    }
    Rob_exit() /* 运行完毕,退出 */
}

```

上面的例程是 Robix 中功能模块的典型模式,首先完成初始化工作,获得用于通讯的共享内存,并将处理异步事件的处理过程向 Robix 登记,然后进入处理主循环.

融合模块在黑板调度器的协调下工作,每个处理周期从黑板获得一个调度信号,根据调度信号进行相应的处理.运行结束退出 Robix.

### 3 应用实例

本集成环境在自主移动机器人 ATB1 集成中已得到成功应用. ATB1 是面向室外准结构化道路环境的移动机器人,ATB1 上共安装了 2 台 Sun Sparc Station 10,6 台 PC486,2 个彩色摄像头,1 套激光测距仪及 1 套定位装置.基本结构如图 1 所示,系统集成分 2 个阶段进行.

#### 第 1 阶段:系统仿真

将各个功能模块按上节的例子进行封装,输入输出接口改为 Robix 提供的通讯函数,机器人系统是一个实时系统,不允许功能模块间存在过多的通讯,而 Robix 提供的接口形式非常简洁,使封装工作很快就完成了.

在我们的仿真系统中,用一个车辆模拟模块,来仿真机器人系统的工作.

#### 第 2 阶段:真实环境中的试验

## A QUANTIFICATIONAL LOGIC OF CONTEXT AND ITS CIRCUMSCRIPTION THEORY

LIU Haiyan CHEN Huowang WANG Bingshan

*(Department of Computer Science National University of Defense Technology Changsha 410073)*

**Abstract** A new QLC (quantificational logic of context), is defined in this paper, which embodies more logical properties of context. Then cases when QLC has equality and when domain-specific axioms are stated in different context sequences are discussed, McCarthy's circumscription theory is also introduced into QLC.

**Key words** Context, artificial intelligence, many-sorted logic, nonmonotonic reasoning, circumscription theory.

**Class number** TP18