



中华人民共和国国家标准

GB/T 16720.3—1996
eqv ISO/IEC 9506-3:1991

工业自动化系统 制造报文规范 第3部分：机器人伴同标准

Industrial automation systems—
Manufacturing Message Specification(MMS)—
Part 3: Companion standard for robotics

1996-12-26 发布

1997-07-01 实施

国家技术监督局 发布

目 次

前言	Ⅱ
ISO/IEC 前言	Ⅳ
引言	V
1 范围	1
2 引用标准	1
3 定义	2
4 缩略语	2
5 机器人应用描述	3
6 机器人应用专用上下文映射	11
7 机器人专用服务和协议	22
8 机器人专用标准化对象	42
9 机器人一致性类别	54
附录 A(提示的附录) 示例	58
附录 B(提示的附录) 进一步问题——高级一致性类别	65
附录 C(提示的附录) 中英对照	67

前 言

本标准是根据国际标准化组织和国际电工委员会 ISO/IEC 9506-3:1991《工业自动化系统制造报文规范 第3部分:机器人伴同标准》而制定的。在技术内容上与该国际标准等效,在编写格式上符合我国标准 GB/T 1.1—1993《标准化工作导则 第一单元 第一部分 标准编写的基本规定》。“目次”按章列出。第1章“范围”按 GB/T 1.1—1993 的要求重新进行了内容和条款的组织,并删除了国际标准中那些说明在本标准中不规定和不标识的内容的句子。第2章“引用标准”按 GB/T 1.22—1993 的规定进行编写,先列出国家标准,后列出国际标准,并按顺序编排。第3章原文定义的术语共13条,本标准删去了与 GB/T 12643 中相重复的7条。第5.2.1.2.2条中,原文有“Tool Tip”的提法,因与机器人术语及功能测试时的提法不同,故改为“TCP”(工具中心点)。第5.2.2.1条图7中坐标轴方向与 ISO 9787 不一致,本标准作了相应的修改。目前 ISO/TC 184/SC5 虽已提出了两个修改件草案,因未正式发布,故本标准其余章条的内容,此版不作更改和补充。在附录的最后,国际标准为“索引”,在本标准中改为“中英对照”,并删去一些重复的词。

通过对该国际标准的等效采用,促使我们全面理解标准的内容,更好地贯彻和实施本标准,从而促使我国机器人和机器人系统的开发和应用。使之在生产中产生更大的经济效益。

根据 GB/T 1.1 的规定,保留了该国际标准的前言和引言。

本标准是《工业自动化系统 制造报文规范》的第3部分,前两部分已转化成我国标准,其余部分正在转化中。其名称如下:

- 第1部分:服务定义;
- 第2部分:协议规范;
- 第3部分:机器人伴同标准;
- 第4部分:数控伴同标准;
- 第5部分:可编程控制器伴同标准;
- 第6部分:过程控制伴同标准;

.....

本标准从1997年7月1日起实施;

本标准的附录A、B都是提示的附录;

本标准由全国工业自动化系统标委会提出;

本标准由机械部北京机械工业自动化所归口并起草;

本标准主要起草人:胡景鏐,郝淑芬,沈重重。

ISO/IEC 前言

国际标准化组织(ISO)和国际电工委员会(IEC)共同组成世界范围标准化体系。作为 ISO 或 IEC 成员的国家团体通过这两个组织所设置的从事各自领域内技术活动的技术委员会,参与国际标准的制定工作。ISO 和 IEC 各技术委员会在共同感兴趣的领域内互相合作。其他与 ISO 和 IEC 有联系的,政府间的和民间的国际组织,也参与了此工作。

在信息技术领域内,ISO 和 IEC 已建立了一个联合技术委员会 ISO/IEC JTC1。由联合技术委员会正式通过的国际标准草案在国家团体中进行投票,至少要有参加投票的国家团体的 75%同意,才能作为国际标准出版。

ISO/IEC 9506-3 国际标准是由 ISO/TC 184“工业自动化系统和集成”联合技术委员会制定的。

ISO/IEC 9506 在总标题“工业自动化系统 制造报文规范”下由下列部分组成:

- 第 1 部分:服务定义;
- 第 2 部分:协议规范;
- 第 3 部分:机器人伴同标准。

附录 A 和 B 仅作为资料性附录。

引 言

本标准用于使用机器人和机器人系统的开放通信系统。该机器人或机器人系统是依据 OSI 模型 (GB 9387) 与通信网络互连的。本标准认可机器人对于一些装置可充当控制器(客户),这些装置包括与机器人相连接的视觉系统和夹持器等。与这些装置通信的客户一致性,本标准未进行定义,应由该装置的伴同标准来定义与这些装置通信的一致性要求,或由 GB/T 16720.1 (ISO/IEC 9506-1) 和 GB/T 16720.2 (ISO/IEC 9506-2) 来定义。

本标准定义了当机器人使用于具有多客户网络中的一致性要求。并采用在 ISO 8824 中定义的方法来描述报文。

本标准给出了几种一致性类别的说明,其中包括一个基本类别。基本类别被作为机器人的最低限的一致性来考虑,这些机器人是作为网络的主计算机或客户装置的“从设备”(Slave)或服务器而连入的。基本类别形成了机器人在这些类型网络中的一致性的“内核”(Kernel)。所有其他的一致性类别将被附加到基本类别上。本标准亦给出了包括 MMS 通用模型中未定义的协议元素的抽象语法表示法在内的机器人专用服务和协议。

本标准认可机器人对与其相连的诸如视觉系统和各种夹持器等装置充当控制器,并标识这种方式中进行通信的要求,但并不标识机器人起客户作用时 MMS 服务和协议的一致性要求。这些一致性要求由预定与机器人通信的那些装置的伴同标准来标识。

MMS 要求与某些标准一起使用,这些标准能实现 GB 9387 所定义的信息处理系统中开放系统互连所需的系统性和统一性的规定。因此,MMS 被置于 OSI 模型的应用层内。它规定了扩展信息系统网络所要求的应用服务元素和协议,该网络是供自动化工厂环境的可编程控制装置使用的。由 MMS 所定义的服务是通用的,供各个伴同标准引用。但每个伴同标准面向一个更专用的应用类别。

本标准考虑到任何时刻机器人都要安全操作。在 ISO 10218 中规定了机器人的安全要求。在本标准中所描绘的全部机器人动作都是在安全标准中许可的。

本标准的实施只需要最低限的包括在第 9 章中的 MMS 执行程序,该章引用了 GB/T 16720.1 和 GB/T 16720.2 的一致性要求。为了正确地实施本标准,机器人和机器人系统的 MMS 实施者对 MMS 应有全面的理解,并应对本标准所确定的建模、服务和协议有充分的理解。机器人及机器人系统的用户应接受本文件中有关服务和建模的章条的指导。

本标准所采用的术语“机器人”即 ISO 8373 中所定义的“操作型工业机器人”。当在本标准中使用术语“机器人”时,通常是指操作机以及为了机器人实施任务所需的控制系统和辅助设备、装置、传感器或通信链。图 1 表示了在本标准中描述的机器人系统的各个元素。由于 ISO 8373 中的定义只描述了具有单臂的机器人系统,而本标准亦考虑了可协调操作的多臂机器人,因此此术语已经通用化。

术语“机器人系统控制器”应包括系统的(多个)机器人臂(可能有多个)的控制程序的(单)任务程序操作。

“MMS 的各种服务”指的是在 GB/T 16720.1 中所定义的抽象服务,“MMS 协议”指的是在 GB/T 16720.2 中所定义的协议。

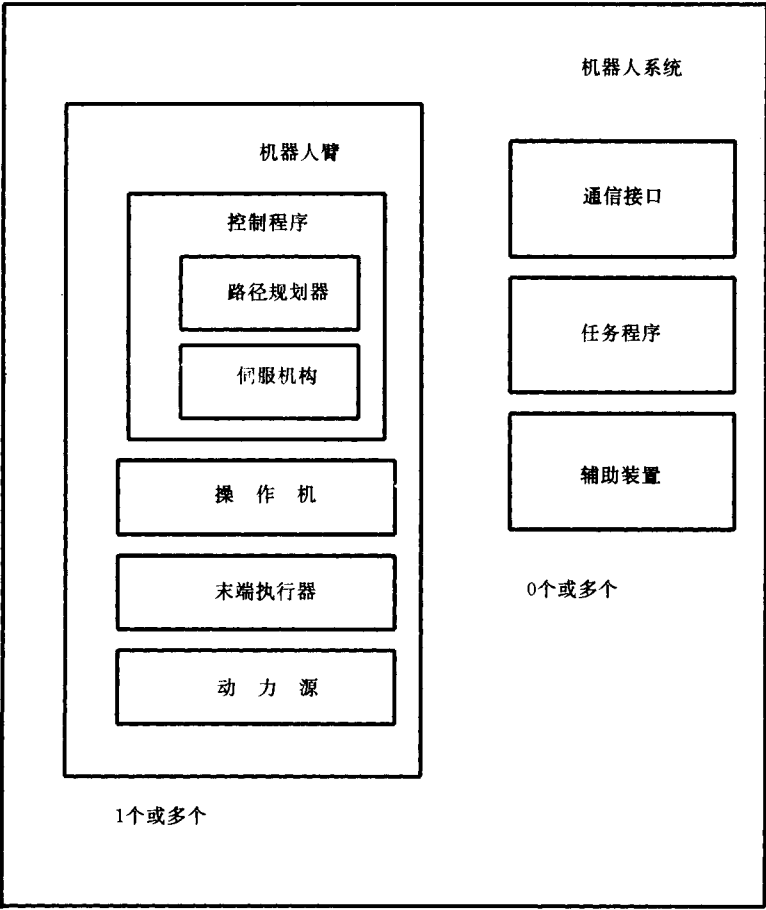


图 1 机器人系统

中华人民共和国国家标准

工业自动化系统
制造报文规范
第3部分:机器人伴同标准

GB/T 16720.3—1996
eqv ISO 9506-3:1991

Industrial automation systems—
Manufacturing Message Specification(MMS)—
Part 3: Companion standard for robotics

1 范围

本标准规定了需要由 MMS 伴同标准规定的协议元素的抽象语法记法表示的各种机器人专用服务和协议及机器人专用的标准化对象。

本标准描述了机器人的模型以及机器人的属性如何被映射到虚拟制造装置的属性上。

本标准给出了包含一个基本类别和几个增强型类别(enhanced classes)的一致性类别的说明。

本标准给出了在本标准规定的抽象语法中作为服务器操作的机器人的协议和各种服务定义。

本标准适用于使用机器人和机器人系统的开放式通信系统,而该机器人或机器人系统是依据 OSI 模型与通信网络相连的。

2 引用标准

下列标准所包含的条文,通过在本标准中引用而构成为本标准的条文。本标准出版时,所示版本均为有效。所有标准都会被修订,使用本标准的各方应探讨、使用下列标准最新版本的可能性。

GB 9387—1988 信息处理系统 开放系统互连 基本参考模型(idt ISO 7498:1984)

GB/T 12643—1990 工业机器人 术语和图形符号

GB/T 15129—1994 信息处理系统 开放系统互连 服务约定(idt ISO/TR 8509:1987)

GB/T 16262—1996 信息处理系统 开放系统互连 抽象语法记法 1(ASN. 1)规范(idt ISO/IEC 8824:1990)

GB/T 16720.1—1996 工业自动化系统 制造报文规范 第1部分:服务定义(eqv ISO/IEC 9506-1:1990)

GB/T 16720.2—1996 工业自动化系统 制造报文规范 第2部分:协议规范(eqv ISO/IEC 9506-2:1990)

ISO 8571:1988 信息处理系统 开放系统互连 文件传送、存取和管理

ISO 8649:1988 信息处理系统 开放系统互连 关联控制服务元素的服务定义

ISO 8650:1988 信息处理系统 开放系统互连 关联控制服务元素的协议规范

ISO 9787:1990 工业机器人 坐标系和运动

ISO 10218:1992 操作型工业机器人 安全

3 定义

本标准采用下列定义。

3.1 通用定义

本标准采用 GB/T 16720.1、GB/T 16720.2 第 3 章及 GB/T 12643 和 ISO 9787 中已定义的术语。

3.2 专用定义

3.2.1 本地控制 local control

是一个布尔值,它指出对于远程操作能或不能引起 MMS 服务器状态的变化。若本地控制是 TRUE(真),则远程操作不能改变服务器的状态。

注:此定义适用于 MMS 伴同标准,但与 ISO 10218 所定义的含义不同。

3.2.2 使能运动 motion enabled

是一个布尔值,若此值为 TRUE(真),则表示对一个机器人臂控制程序的有效指令的存在将导致机器人臂的运动。

3.2.3 远程操作 remote operation

通过 OSI 通信网络使用 MMS 服务进行操作的数据采集或控制。

3.2.4 机器人臂 robot arm

当在本标准中使用,指的是一个操作机和末端执行器及其动力源以及控制操作机的控制程序。

3.2.5 机器人系统控制器 robot system controller

机器人的整个控制系统,由(单个)任务程序及用于控制(单或多个)机器人臂和(单或多个)辅助装置的(单或多套)控制程序组成。

注:本标准的术语“机器人”即“工业机器人”。

3.2.6 步 step

一个任务程序执行的基本元素。它可包含也可不包含机器人的运动。

注:步的概念取决于机器人的编程语言。

4 缩略语

在本标准的正文中使用了下述缩略语。

ACSE:	关联控制服务元素
ASE:	应用服务元素
ASN.1	抽象语法表示法(1)
C:	条件参数
CBB:	一致性构造块
Cnf:	确认
CS:	伴同标准
DIS:	国际标准草案
FTAM:	文件传送、存取和管理
ind:	指示
I/O:	输入/输出
IS:	国际标准
M:	代理
MICS:	机械接口坐标系
MMS:	制造报文规范
OSI:	开放系统互连

PDU:	协议数据单元
PICS:	协议执行程序一致性语句
Req:	请求
Rsp:	应答
S:	选择
TR:	技术报告
U:	用户任选参数
VMD:	虚拟制造装置

5 机器人应用描述

5.1 制造配置

5.1.1 一般考虑

在本标准第9章中所定义的一致性类别与其被使用时的配置无关。在此条款中所描述的配置仅起指导作用,是为了对作为本标准基础的基本原理的深入了解。实际的配置能够同时展现几种配置的特性。

在远程通信环境中,一个节点被认为是客户,另一个节点被认为是服务器。一个与机器人连接的主机(host)被认为是机器人的客户。主机通常给机器人以命令并监控机器人的功能。机器人相对于主机被认为是服务器。在机器人通过 OSI 通信通道与智能周边设备(诸如夹持器、视觉系统等装置)或其他机器人相连接的情况下,此机器人相对于这些装置被视为客户。本标准所描述的各个装置之间的连接为逻辑连接。

虽然操作盘和示教盒能够被用于直接操作机器人,但在上述的意义中,由于它们没有通过 OSI 通信通道与机器人连接,因此不认为它们是客户。然而,可以认为它们是机器人服务器的一部分。

当客户与机器人进行通信时,本标准没有对客户的配置规定任何硬性要求。只要客户有能力把适当的请求发送到机器人和接收来自机器人的应答。

5.1.2 配置(一):机器人服务器,单客户

本配置是由一个客户(即主计算机)和一个机器人所组成(见图2),客户控制机器人或与其进行通信联络。客户或主机把请求发送到机器人或服务器,对此,机器人必须应答。机器人可包含其本身的诸如视觉和夹持器控制等子系统,这些子系统不通过 MMS 来直接控制,也不属于配置(一)的范围。

在配置(一)的简单执行程序中,仅需要一个 MMS 关联。在较复杂的执行程序中,可以有多个共存的 MMS 关联。

例如,为了增加通流量,在主机和机器人之间使用了两个 MMS 关联。

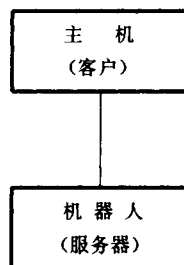


图2 机器人服务器/单客户

5.1.3 配置(二):一个机器人服务器,多个客户

在配置(二)中,机器人是服务器,但有多个客户(见图3)。由于任何客户(主机)都能初始化一个关

联,因此,要求机器人(服务器)部分能对多个共存的关联进行支持。

一个多客户配置要求有一个取得和放弃机器人控制权的机制。没有这种能力,就没有方法去防止两个以上的客户同时企图控制机器人。

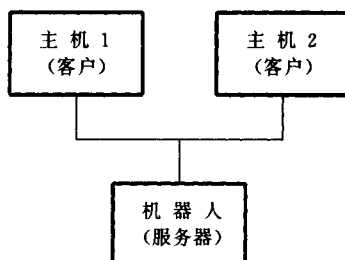


图 3 机器人服务器/多个客户

此种配置的一个例子是一个实施机器人控制的主机及一个作为更大系统一部分用以监控机器人的第二主机。

5.1.4 配置(三):机器人客户

在配置(三)中,机器人是一个或多个装置的客户(见图 4)。该机器人同时也可能如配置(一)和配置(二)所示,对一个或多个主机而言,又是一个服务器。机器人在客户身份中,要求能起请求启动器的作用,而不仅是一个应答器。

本标准仅定义作为客户的系统和各个作为各服务器的机器人之间的交互作用,并不定义与其他装置如视觉系统、夹持器等交互作用。此类交互作用只能由基于这些装置的要求来定义。此类配置的一例是一个作为客户的机器人对作为文件服务器的另一个系统。

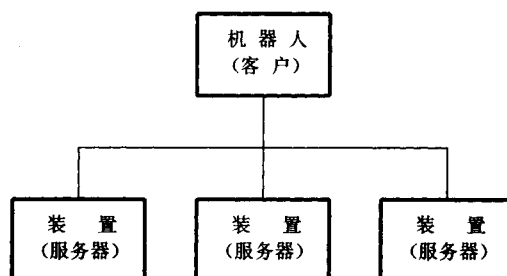


图 4 机器人客户

5.1.5 配置(四):同层对同层

在配置(四)中,几台机器人被认为是同层的,并且都能起客户和服务器两种作用(见图 5)。在此配置中,每个机器人含有客户和服务器两重身份。

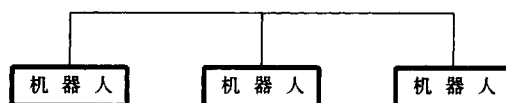


图 5 同层对同层

5.2 机器人专用模型

5.2.1 机器人物理模型

5.2.1.1 总则

本标准描述了一个机器人系统的抽象模型。为了从通信通道的角度描述机器人的活动能力,需要有该模型的各种属性。实际的机器人系统可能有比在此描述的更多的特性。但也不要求机器人系统具有在本标准中所描述的全部属性。

一个机器人系统是由一个或多个机器人臂以及一个机器人系统控制器组成。还有若干与机器人臂协调作用的辅助装置,它们是从机器人臂中物理地和逻辑地分离出来的。特别是各种安全联锁就是此类辅助装置之一,其中急停按钮可视为一个组件。

5.2.1.2 机器人臂

5.2.1.2.1 机器人臂子系统

机器人系统的中心要素是由操作机及其动力源、机器人臂控制程序和末端执行器组成的机器人臂。本标准的集中点是机器人臂的远程操作及与其一起的辅助装置的协调控制。

操作机是由一组机械连杆和关节组成,每个连杆与其相关连的关节一起组成机器人的一个轴。关节是由一个被机器人臂控制程序所控制的驱动器所驱动。机器人臂的控制程序被认为是由两个主要部分组成:一个是伺服机构,一个是路径规划器。

在本标准中,已建模的机器人臂所具有的特性,同样地影响着伺服机构和路径规划器子系统。为了实施机器人的任务,了解操作机的关节数目和每个关节的特性是必要的。为了正确地操作机器人系统,必须了解实际的校准状态(已校准、未校准或正在校准)和电源状态(臂电源开/关)。

在一个机器人臂中,信息流程可描述如下(见图6):

任务程序生成操作机的编程的位姿。

然后,编程的位姿被传送到路径规划器的子系统中,该系统生成以关节值表示的指令状态。

然后,把该指令状态输入到驱动操作机的伺服机构子系统中。

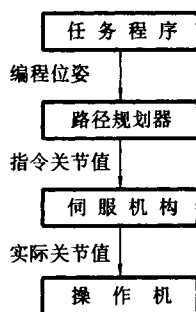


图6 信息流程

5.2.1.2.2 路径规划器

路径规划器子系统接收来自任务程序的指令(或来自其他的局部资源),并且把这些请求转成一个计时的成组的伺服机构指令使操作机运动。实际上,路径规划器是负责把所要求的操作机的路径转换成适当的伺服机构指令,而其路径是由工具中心点的速度或位置来表示的。为了以惯用的方式来表示这些运动,引入一组坐标系来描述机器人臂。这将在5.2.2中加以详细说明。

路径规划器是负责监控操作机运动的某些特性,并使用期望值来修改。其中值得注意的是工具中心点速度和加速度的控制。

编程速度通常是将指令输入到路径规划器而形成的速度。路径规划是来自机器人系统控制器的任务程序。编程速度通常是由任务程序所建立的工具中心点的速度。

速度系数是一种适用于改变操作机速度的超限增值系数。速度系数将改变操作机的编程速度,因此整个操作机的运动(由任务程序所支配的)能够以某个速度来实施,该速度可由被编程速度均匀地进行上下调节。

编程加速度通常是把指令输入到路径规划器而形成的加速度。路径规划是来自机器人系统控制器

的任务程序。编程加速度通常是由任务程序所建立的工具中心点的加速度。

加速度系数是适用于改变操作机的加速度的一个超限增值系数。加速度系数将改变操作机的编程加速度,因此整个操作机的运动(由任务程序控制的)能够以某个加速度来实施,此加速度可由编程加速度均匀地进行上下调节。

路径规划器的附加属性包括一组输入和输出值。路径规划器的输入值描述了包含位置、速度和加速度的以欧基里德空间来表示的操作机的编程状态。路径规划器的输出值描述了以关节空间表示的操作机的指令状态。

5.2.1.2.3 机器人臂伺服机构

机器人臂伺服机构子系统是由一组连接的伺服机构组成。操作机每个关节都有一个。操作机的关节可以是转动类型或是滑动类型。每个关节可以单独地校准,有一个行程的上下限。每个关节有一套制动机构及一组与构成机器人臂运动链中其他关节有关的连杆参数。每个关节由一个伺服机构所控制,伺服机构把控制信号送到关节驱动器,并且从关节传感器取得反馈信息。伺服机构发出指令以驱动关节到一个预定值,并且监控由它运行的实际值。

5.2.1.3 辅助装置

机器人辅助装置是与机器人任务相关的子系统,它是直接由机器人系统控制器控制的。辅助装置的控制并入到机器人任务控制器中。在辅助装置控制中使用的信息,例如辅助装置的专用参数,是机器人任务总参数不可缺少的一部分。辅助装置的控制能够被一个或多个机器人程序来实现。

注:辅助装置的实例有安全连锁、焊接控制器、喷涂系统、水喷射系统、视觉系统、传感器和夹持器等。

5.2.2 机器人坐标系

5.2.2.1 约定

本标准所使用的坐标系有绝对坐标系、机座坐标系和机械接口坐标系,在 ISO 9787 中描述了这些已标准化的坐标系。为了帮助了解本标准,在 5.2.2.2 和 5.2.2.3 中描述了另外两种机器人坐标系。对于整个制造操作,绝对坐标系建立了一个固定的参照框架,并且通常用于工位布局。此坐标系的定义一般是由用户来制定。机座坐标系建立在机器人安装平面处,通常由机器人制造厂给出。机械接口坐标系是一个依附于操作机最后杆件上的运动着的坐标系。该杆件确立了末端执行器相对于机座坐标系的位置。

除已标准化的坐标系外,引入了两个经常使用的坐标系,即工具坐标系和用户坐标系。工具坐标系用于定义机器人工具(末端执行器)的位置。用户坐标系用于替代机座坐标系供机器人臂作为参照框架。图 7 表示各种机器人坐标系的约定。

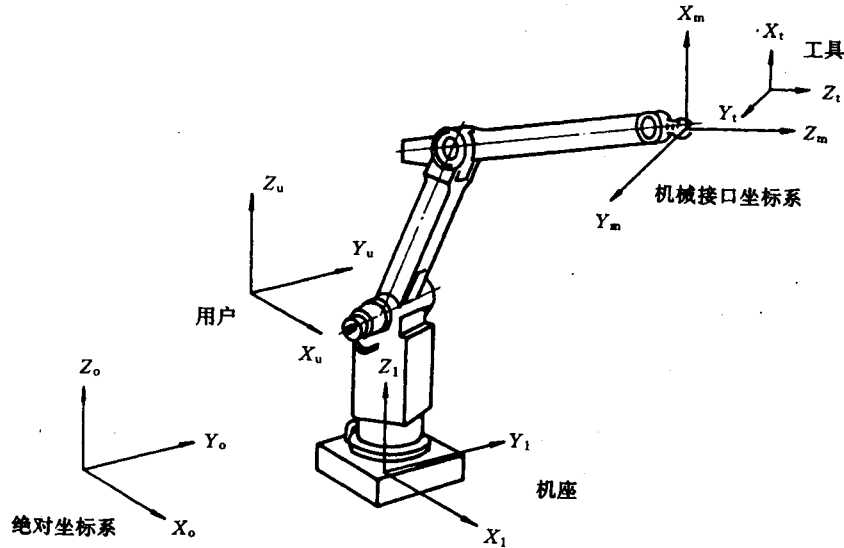


图7 机器人坐标系

5.2.2.2 工具坐标系(X_t, Y_t, Z_t)

工具坐标系的原点是由实施者所定义的位于机器人工具上的点, $+Z_t$ 和 $+X_t$ 轴线是由实施者所定义且相互垂直。 $+Y_t$ 轴线是由右手定则确定的。

在工具坐标系内的一个物体由坐标 $(x_t, y_t, z_t, a_t, b_t, c_t)$ 变换来表示。其中:

- x_t 表示沿着 X_t 坐标轴线的位移;
 - y_t 表示沿着 Y_t 坐标轴线的位移;
 - z_t 表示沿着 Z_t 坐标轴线的位移;
 - a_t 表示绕着 $+X_t$ 坐标轴反时针方向旋转;
 - b_t 表示绕着 $+Y_t$ 坐标轴反时针方向旋转;
 - c_t 表示绕着 $+Z_t$ 坐标轴反时针方向旋转;
- 坐标变换遵循右手定则正负号方向的约定。

5.2.2.3 用户坐标系(X_u, Y_u, Z_u)

用户坐标系的原点是实施者所确定的点。 $+Z_u$ 和 $+X_u$ 轴是实施者确定的, 并且互相垂直。 $+Y_u$ 轴是由右手定则确定的。

在用户坐标系内的一个物体由其坐标 $(x_u, y_u, z_u, a_u, b_u, c_u)$ 的变换来表示, 其中:

- x_u 表示沿 X_u 坐标轴的位移;
- y_u 表示沿 Y_u 坐标轴的位移;
- z_u 表示沿 Z_u 坐标轴的位移;
- a_u 表示绕 $+X_u$ 坐标轴反时针方向旋转;
- b_u 表示绕 $+Y_u$ 坐标轴反时针方向旋转;
- c_u 表示绕 $+Z_u$ 坐标轴反时针方向旋转。

坐标变换遵循右手定则正负号方向约定。

5.2.2.4 坐标系变换

在描述机器人运动的过程中,必须描述空间物体的任意位置和姿态。上面所述的任一坐标系,都能够作出这种描述。就某些坐标系而论,一个物体位置和姿态的说明相当于从参考坐标系转换成嵌入在物体中的坐标系的说明。因此,在机器人运动数学中,变换起着很大作用。

上面所描述的五种坐标系,其中四种常常是用作描述其他物体时的参照坐标系。而工具坐标系是嵌入在物体中的坐标系,而该物体需要知道位置和姿态。本标准标识了坐标系的五种变换。这些变换是用于描述机器人系统部件的动力学的相互关系。他们是:

- 机座坐标系相对于绝对坐标系的变换。
- 机械接口坐标系相对于机座坐标系的变换。
- 用户坐标系相对于机座坐标系的变换。
- 工具坐标系相对于用户坐标系的变换。
- 工具坐标系相对于机械接口坐标系的变换。

5.3 机器人专用功能

5.3.1 控制系统

5.3.1.1 机器人系统控制器

机器人系统的一个基本部件是机器人系统控制器,它代表了支配机器人操作的智能。机器人系统的控制器通常是通过采用多用途计算机运行一个或多个计算机程序来实现,但也可用其他的方法来实现。在以计算机为基础的方式中,下位计算机在使用的控制功能可能比机器人控制功能更多,但本标准只涉及计算机的与机器人操作有关的功能。在此处机器人操作的涵义包括为完成机器人功能所需的全部辅助装置的操作。

集成单元控制器是功能超出本标准范围的此类计算机系统的一例,作为其操作的一部分。它直接指挥机器人活动。

5.3.1.2 机器人控制的协调

在许多情况下,能由多点控制机器人的操作。引入通信设备则增加了另一种潜在的控制点。为了排除在控制中的冲突(conflicts),设定机器人具有唯一地对一个控制点的控制赋值方法,此控制赋值起互斥信志作用。

5.3.1.3 机器人操作状态

5.3.1.3.1 机器人操作状态图

如图8中的描述,机器人系统由一个有限状态自动化装置来建模。

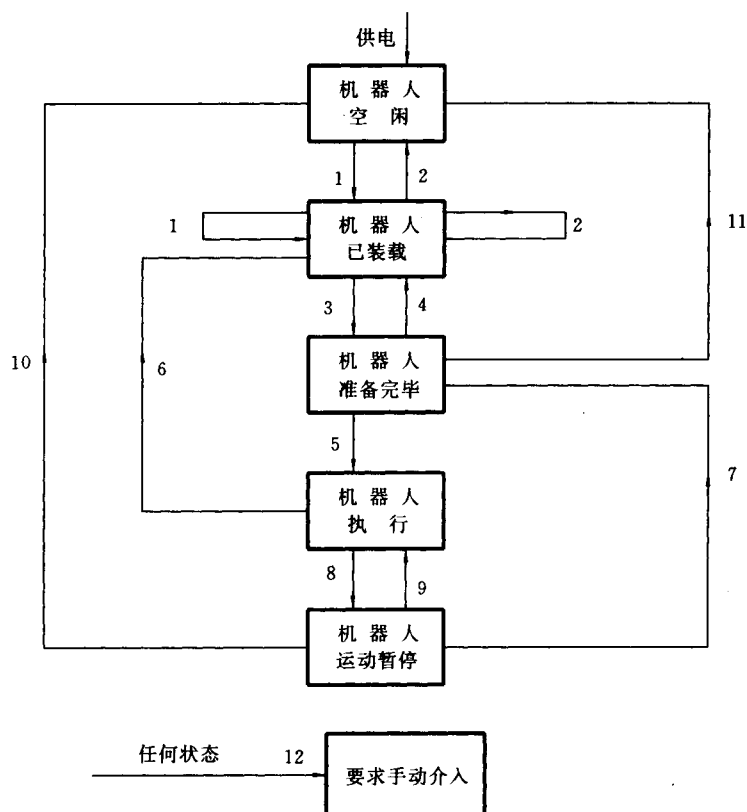


图 8 机器人操作状态

图 8 中转移编号的说明：

- 1—装载 某些任务程序被装入机器人系统控制器中。
- 2—清除 由机器人系统控制器中移去任务程序。
- 3—赋值 已准备好一个已装载的任务程序以供执行。
- 4—去值 赋值的反面。
- 5—程序开始 机器人进入 ROBOT—EXECUTING 状态。
- 6—执行结束 机器人运动被停止且机器人被返回到 ROBOT—LOADED 状态。
- 7—程序复位 一个已暂停的机器人被返回到 ROBOT—READY 状态。
- 8—暂停 机器人已临时停止运动。
- 9—继续 暂停的机器人返回到 ROBOT—EXECUTING 状态。
- 10—清除(异常终止) 除任务程序未执行完成外,其他与清除同。
- 11—清除(舍弃) 除任务程序未开始外,其他与清除同。
- 12—应急动作 将机器人置入要求手动介入的状态。

5.3.1.3.2 机器人操作状态说明

5.3.1.3.2.1 ROBOT—IDEL(机器人空闲)

ROBOT—IDEL 状态对应应在系统内机器人没有任务程序可用时的机器人状态。当机器人刚开始供电时,或当其完成一个任务程序执行和任务程序由系统内移去时,将发生此状态。

5.3.1.3.2.2 ROBOT—LOADED(机器人已装载)

ROBOT—LOADED 状态对应应在系统内有一个或多个可用的任务程序时的机器人状态。

5.3.1.3.2.3 ROBOT—READY(机器人准备完毕)

ROBOT—READY 状态表示一个可供机器人执行的任务程序已被赋值。

5.3.1.3.2.4 ROBOT—EXECUTING(机器人执行)

在 ROBOT—EXECUTING 状态中,机器人在操作,其运动是使能运动。

5.3.1.3.2.5 ROBOT—MOTION—PAUSED(机器人运动暂停)

在 ROBOT—MOTION—PAUSED 状态中,运动不是使能运动。机器人的任务程序可能在运行也可能已停止。

5.3.1.3.2.6 MANUAL—INTERVENTION—REQUIRED(要求手动介入)

在此状态中,全部运动已经停止。此时,为了完成某些其他的必要活动,需要靠机器的操作者执行某些局部动作。在此局部动作后,机器人才可进入某一其他状态。

5.3.1.3.3 机器人操作状态转移

5.3.1.3.3.1 总则

在下述对转移的说明中,假定所有这些状态的转移都可以通过本地活动出现。有些转移也能通过远程活动实现。第6章描述了使用 MMS 服务引起这些转移的远程活动。

5.3.1.3.3.2 装载

装载转移描述了把一个任务程序装入机器人系统控制器的行为。机器人系统控制器在其存储器中可以容纳几个任务程序;当然,转移只考虑能够操作和控制机器人臂运动的任务程序。若在机器人系统控制器内无此类程序,则此转移将使机器人由 ROBOT—IDEL 状态进入 ROBOT—LOADED 状态。如果存储器中至少已有一个此类任务程序,则作为转移结果,机器人将处于 ROBOT—LOADED 状态。

5.3.1.3.3.3 清除

这种转移出现在由机器人系统控制器中移去一个任务程序时,若该任务程序又是最后一个或只有该程序,则机器人处于 ROBOT—IDEL 状态。否则,余留的任务程序将继续处于 ROBOT—LOADED 状态。

5.3.1.3.3.4 赋值

在实际执行中,一个潜在任务程序赋值的过程,将使机器人置于 ROBOT—READY 状态。对于只能包含一个任务程序的机器人,装载和赋值转移可以结合成一个单个的操作。

5.3.1.3.3.5 去值

为了将机器人由 ROBOT—READY 状态运行到 ROBOT—LOADED 状态,需请求一个去值转移。这是赋值转移的反转。某些机器人可能出现允许由 ROBOT—READY 状态进行一个赋值转移。这可用一个去值转移跟随一个赋值转移来模拟。

5.3.1.3.3.6 程序开始

程序开始是机器人在运动时的基本操作。由于此转移的结果,机器人进入 ROBOT—EXECUTING 状态。

5.3.1.3.3.7 执行结束

作为机器人系统控制器内部局部活动的结果,机器人从 ROBOT—EXECUTING 状态移出进入 ROBOT—LOADED 状态。机器人再次进入 ROBOT—EXECUTING 状态之前,需要进行一次赋值转移。

5.3.1.3.3.8 程序复位

在 ROBOT—MOTION—PAUSED 状态的机器人能被程序复位转移运行到 ROBOT—READY 状态。

5.3.1.3.3.9 暂停

暂停转移引起机器人停止其运动。任务程序可以继续执行或可以停止。

5.3.1.3.3.10 继续

继续转移将机器人由 ROBOT—MOTION—PAUSED 状态运行返回到 ROBOT—EXECUTING 状态。

5.3.1.3.3.11 清除(异常终止)

有可能从 ROBOT—MOTION—PAUSED 状态进行清除转移。在此情况下,异常终止了(即不正常地结束)现行任务程序的执行,并且已赋值的任务程序被去值。从机器人系统控制器中除去所有潜在的任务程序。

5.3.1.3.3.12 清除(舍弃)

也有可能从 ROBOT—READY 状态进行清除转移。在此情况中,异常终止了(即不正常地结束)现行任务程序的执行,并且已赋值的任务程序被去值。从机器人系统控制器中除去所有潜在的任务程序。

5.3.1.3.3.13 应急动作

由于本地或远程活动的结果,可发生应急动作转移,从任何状态进入 MANUAL—INTERVENTION—REQUIRED 状态。由此状态进行恢复绝不能用自动操作,且要求操作员做明显的动作,才能将机器人转入某一其他状态。

5.3.1.4 本地控制

除机器人操作状态外,机器人能够被描述为处于“正在本地控制中”或“不在本地控制中”。“正在本地控制中”意味着某些本地作用因素,如操作员正在控制仪表板、示教盒或其等同物上正在进行控制操作,以确定机器人可作哪些运动或机器人可运行哪些任务程序。若机器人“不在本地控制中”,则机器人可照常运行任务程序,此时,任何本地作用因素都没有能力停止或修改在运行的任务程序。带远程控制器的机器人可由这种远程控制器取得对该机器人的控制权。

5.3.2 任务程序执行——循环和步

机器人的操作可以被分解成循环和步。循环是表示经常重复一个操作序列。机器人操作的每次循环完成机器人所编程的任务。循环是机器人操作的自然单元。

在一次循环内,可将机器人的动作区分为更微细的动作。这些微细动作被称为“步”,并且与机器人的某些基本运动相对应,或与任务程序的某些基本活动或一个辅助装置的某些基本动作相对应。顺序操作的思想是“步”的概念的基础;机器人是由系统的各元素组成的,每个元素按照时序顺序完成基本操作。并非所有的机器人系统都允许分步控制,但对那些能分成步的机器人系统,本标准提供了分步控制的控制步骤。

当机器人开始操作时,可以设置为按一个固定循环数操作,或(若以步进方式)按在一个循环内的固定步进数操作。通常后一种方式仅用于程序排错。换言之,机器人可以在操作时不计数地重复实施同一循环,直到某个外部事件终结该过程。

5.3.3 校准

除正常的操作外,大多数机器人都具有校准机器人轴的辅助功能。此功能按类似于正常任务程序的方式操作,除由于校准功能的存在使机器人不处于 LOADED 状态外,服从图 8 的状态图。

6 机器人应用专用上下文映射

6.1 映射机器人模型到 VMD 对象

6.1.1 映射步骤的说明

本章将机器人通用模型(见第 5 章)与 GB/T 16720.1 描述的虚拟制造装置(VMD)的抽象模型建立关系。若已达到交互操作的最终目标,可将这些抽象概念应用于各种实系统中。

由于本标准无法预测正在制造的或将要制造的机器人系统的各种变化,因而只对大多数机器人都适用的抽象模型和实系统的元素间建立关联提供一般性指导。

映射过程是包括下列各步的多步步骤:

a) 首先构造一套真实物理系统的抽象模型,即第 5 章制定的模型。

b) 由抽象模型构造一组抽象对象和属性。

c) 将成组的属性与 MMS 抽象对象的一组属性相关联。如有必要,这种关联允许定义 MMS 抽象对象的扩展。

d) 最后,将装置专用的抽象对象与 MMS 抽象对象相关联。

图 9 举例说明了映射过程。

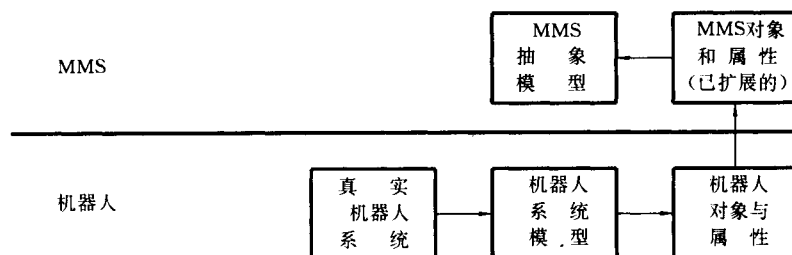


图 9 MMS 映射过程

为了适应机器人模型所有的特点,6.3 制定了程序调用模型的扩展。这些扩展允许对称作“施控”和“受控”的两类程序调用间的交互配合进行说明。受控程序调用用于模拟包括机器人臂控制程序在内的物理装置控制器的操作。施控程序调用用于模拟指挥机器人系统操作的专用程序调用。当一个 VMD 有几个施控程序调用存在时,在任何特定时间,只有一个程序调用被标识为被选操作的调用。除了与受控程序调用相关联的物理装置外,每个物理装置亦被映射到一个预定的定义域中。在 6.2 中描述了这种映射。

6.1.2 机器人系统特性

6.1.2.1 单臂机器人

从建模的观点看,最简单的机器人系统是仅有一个臂的系统。VMD 是与整个机器人系统、机器人臂、机器人系统控制器以及某些辅助装置相关联。机器人臂控制程序是被作为一个与运动硬件相关的受控程序调用来模拟的。

6.1.2.2 多个联动臂的机器人

有的机器人系统使用了一个以上的臂。若一个系统具有单个的施控智能,使几个臂的运动总是以联动方式连接,则这样一个系统可以作为单个 VMD 来模拟。在此情况中,将存在有一个控制双臂运动的单个施控程序调用。

这种模拟是与机器人系统可见的外观无关。硬件的两个分离部件能分别安装在两个场所。如果两部件的活动总是紧密地联动着,用单个 VMD 来模拟才正确。

6.1.2.3 多自主臂机器人

有的机器人系统在系统内同一场所采用了多臂,并且其臂的运动是自主的,也就是对于每个臂都有一个分离的施控智能。在这种情况下,系统应被作为具有多个 VMD 来模拟;第二种情况是对每个受控智能,都具有一个 VMD,每个 VMD 中将有一个控制着运动的单个施控程序调用。应注意上述两种情况之间的区别,它们在很大程度上是取决于机器人系统控制器的配置。被配置的同一硬件部分既可作为是一个多个自主臂的机器人,也可作为一个多个联动臂的机器人。在此情况中,对于每种配置,其对应模型将是不同的。

6.1.2.4 辅助装置

操作辅助装置的任务是同机器人臂控制程序以协调的方式来进行,并且可被认为是一单施控任务的一部分。它们被模拟为附加的受控程序调用。

6.1.3 机器人 VMD

6.1.3.1 机器人 VMD 属性模型

给定 VMD 与机器人系统的关联,即定义了描述机器人的 VMD 附加属性(除 GB/T 16720.1 给定以外)。因为 VMD 的基本特征是机器人臂和辅助装置的协调控制,因此,在 VMD 中标识了提供该控制的施控程序调用。在其执行进程中,施控程序调用将产生输出,该输出将用于对受控程序调用进行输入。

机器人操作状态取决于硬件的状况(无论运动是否是使能运动),由一组受控程序调用来表示,并取

决于被选施控程序调用的状态(见 6.3)。

对象: VMD

所有被定义的 MMS 属性

属性: 违反安全联锁(TRUE,FALSE)

属性: 机器人操作状态(ROBOT—IDEL,ROBOT—LOADED,ROBOT—READY,ROBOT—EXECUTING,ROBOT—PAUSED,MANUAL—INTERVENTION—REQUIRED)

属性: 任何物理资源通电(TRUE,FALSE)

属性: 全部物理资源已校准(TRUE,FALSE)

属性: 本地控制(TRUE,FALSE)

属性: 被选施控程序调用的引用

6.1.3.2 机器人 VMD 属性说明

6.1.3.2.1 违反安全联锁

此属性是布尔型,它指出安全联锁的状态。若该属性值是 TRUE,从机器人系统持续复位后,则说明违反了安全联锁。安全联锁复位的方法是一个本地事务。

6.1.3.2.2 机器人操作状态

此属性是在 5.1.3.1 中所描述的机器人系统的状态。在 6.3.4 中定义了该属性对物理装置以及对施控程序调用状态的相互关系。

6.1.3.2.3 任何物理资源通电

此属性是布尔型,它指出是(TRUE)否(FALSE)已经对任何物理资源供电。此属性对系统中每个物理资源是一个类似原始属性的逻辑“或”。

6.1.3.2.4 全部物理资源已校准

此属性是布尔型,它指出是(TRUE)否(FALSE)系统中具有校准属性的物理资源有等于 CALIBRATED 的值(见 6.2.2.4 和 6.2.3.4)

6.1.3.2.5 本地控制

此属性是布尔型,它指出是(TRUE)否(FALSE)某些局部作用因素正在对系统中任何物理资源“保持控制”。保持控制表示具有引起改变物理资源动作的能力及表示该资源的属性。“本地控制”能反映出是操作员还是某些自动化步骤,哪一个远程地掌握了控制要求。

如果本地控制是 FALSE,控制可以归于某些远程作用因素。这种状况并能反映出机器人的运行除任务程序外,没有其他的直接控制条件。

本地控制属性值、机器人 VMD 状态、VMD 逻辑状态三者是相互联系的。表 1 举例说明了这些属性之间的相互关系。

表 1 本地控制

本地控制	机器人操作状态	VMD 逻辑状态
是	任意状态	NO—STATE—CHANGES—ALLOWED 或 LIMITED—SERVICES—PERMITTED 或 SUPPORT—SERVICES—ALLOWED
否	要求手动介入	NO—STATE—CHANGES—ALLOWED 或 LIMITED—SERVICES—PERMITTED 或 SUPPORT—SERVICES—ALLOWED
	任何其它状态	任 意

注: 当在本地控制时,VMD 状态能继续变化,然而,如何发生状态变化是一个本地事务。

6.1.3.2.6 被选施控程序调用

该属性标识控制机器人操作的被选程序调用。该程序调用表示已赋值的任务程序(见 5.3.1.3.3.4)。该程序调用的控制属性等于 CONTROLLING(见 6.3.2.3)并已通过使用“选择操作”(见 7.4.4)被选择。若没有这种被标识的程序调用,则此属性应具有值 NONE。

6.2 映射到域的机器人特定对象

6.2.1 总则

按本标准的模型,机器人是由一组表示机器人臂(或多个臂)和辅助装置的物理资源组成。每个物理资源或资源组应被映射到表示该资源的域中。每个域应该有与资源的控制、资源的电源状态相关的属性,并且可以有一个指示资源校准状态的属性。

机器人臂的标准化域应与机器人臂的资源相关联,或在多臂的情况下,一个域与每个臂均相关。域,在其定义内应包括全部周边元件。这些元件通常是机器人臂的一部分。

由单独程序控制的辅助装置应该由单独的预定义域来模拟。特别是安全联锁需由一个此类域来模拟。对于所有数据报告服务(即读服务),这些域应仅限于基本资源。对于影响控制或装置状态变化的服务,其关系应由 VMD 的本地控制属性来调节。

对于 MMS 域,不定义扩展;机器人特定对象的属性应该由在域内定义的特定域的有名变量对象来实现。在第 8 章中描述了这种有名变量对象的命名标准。

6.2.2 机器人臂资源

6.2.2.1 机器人臂属性模型

一个机器人臂包含臂的全部元素,臂通常由机器人控制程序所控制。机器人臂对象的属性可从运动学和控制的观点描述机器人臂。

该属性模型仅包含对机器人进行远程操作所要求的属性。任何实际的机器人具有很多未包含在本标准中的属性。

对象: 机器人臂

属性: 本地控制(TRUE,FALSE)

属性: 装置供电(TRUE,FALSE)

属性: 装置校准(CALIBRATED,NOT—CALIBRATED,CALIBRATING)

属性: 使能运动(TRUE,FALSE)

属性: 关节数——整数

属性: 机座坐标系——绝对坐标系——位姿

属性: 伺服机构

属性: 机械接口坐标系——机座坐标系——位姿

属性: 有序关节描述表

属性: 关节类型(REVOLUTE,PRISMATIC)

属性: 校准(CALIBRATED,NOTCALIBRATED,CALIBRATING)

属性: 关节制动器(TRUE,FALSE)

约束: 关节制动器=TRUE

属性: 制动器合(TRUE,FALSE)

属性: 上限——浮点

属性: 下限——浮点

属性: 关节伺服

属性: 实际关节值——浮点

属性: 路径规划器

属性: 用户坐标系——机座坐标系——位姿

属性: 期望的工具坐标系——用户坐标系——位姿

属性：速度系数——浮点

属性：编程速度——浮点

属性：加速度系数——浮点

属性：编程加速度——浮点

属性：末端执行器

属性：标识符编号

属性：工具描述符

属性：工具坐标系——机械接口坐标系——位姿

注：通常机器人术语认为一个轴表示一个关节和一相关联的连杆。连杆信息是本地事务。对本抽象模型而言，关节和轴是同义的。

6.2.2.2 本地控制

此属性是布尔型，它指出机器人臂是(TRUE)否(FALSE)处于本地控制中。

6.2.2.3 装置供电

此属性是布尔型，它指出是(TRUE)否(FALSE)已对机器人臂供电。

6.2.2.4 装置校准

此属性表示机器人臂作为整体是否已校准。若全部关节已被校准，则该属性有值 CALIBRATED，若任何关节正在校准，则有值 CALIBRATING，否则就是 NOT—CALIBRATED。

6.2.2.5 使能运动

此属性是布尔型，它指出若一个有效的指令信号出现在路径规划器时，机器人臂(或其任何部分)是否将产生运动。仅当使能运动属性是 TRUE 时，机器人臂才可以自主地运动。

6.2.2.6 关节数

此属性是整数型，它指出机器人臂关节数。

6.2.2.7 机座坐标系——绝对坐标系

此属性是位姿型，它指出机座坐标系相对于绝对坐标系的变换值。

6.2.2.8 伺服机构

6.2.2.8.1 机械接口坐标系——机座坐标系

此属性是位姿型，它指出机械接口坐标系相对于机座坐标系的变换的实际值。它取自于实际的关节值属性和执行程序专用参数。

6.2.2.8.2 有序关节描述表

6.2.2.8.2.1 关节类型

此属性指出关节机构是转动还是滑动。

6.2.2.8.2.2 校准

此属性指出关节是 CALIBRATED, NOT—CALIBRATED 还是 CALIBRATING。

6.2.2.8.2.3 关节制动器

此属性是布尔型，它表示制动器对该关节是(TRUE)否(FALSE)存在。如果制动器存在(关节制动是 TRUE)，则属性“制动器合”存在。“制动器合”属性为布尔型，它表示制动器是(TRUE)否(FALSE)使用于制止运动。

6.2.2.8.2.4 上限

此属性是浮点型，它指出关节行程的上限。

6.2.2.8.2.5 下限

此属性是浮点型，它指出关节行程的下限。

6.2.2.8.2.6 关节伺服——实际关节值

此属性是浮点型，它指出关节参数的实际值。该值是直线位置或是角度位置则取决于关节类型。

6.2.2.9 路径规划器

6.2.2.9.1 用户坐标系——机座坐标系

此属性是位姿型,它指出用户坐标系相对于机座坐标系的变换值。

6.2.2.9.2 期望的工具坐标系——用户坐标系

此属性是位姿型,它指出工具坐标系相对于用户坐标系的期望的位姿值。

6.2.2.9.3 速度系数

此属性是浮点型,它标识路径规划器编程速度输入所采用的超限速度系数。

6.2.2.9.4 编程速度

此属性是浮点型,它标识在机器人程序中已设定的机械接口坐标系相对于机座坐标系的编程速度的幅值。

6.2.2.9.5 加速度系数

此属性是浮点型,它标识路径规划器编程加速度输入所采用的超限加速系数。

6.2.2.9.6 编程加速度

此属性是浮点型,它标识在机器人程序中已设定的机械接口坐标系相对于机座坐标系的编程加速度的幅值。

6.2.2.10 末端执行器

6.2.2.10.1 标识符编号

此属性是对象标识符型,它唯一地标识在一定的集合内所有末端执行器中的一个末端执行器。

6.2.2.10.2 工具描述符

此属性提供了末端执行器的内容描述。

6.2.2.10.3 工具坐标系——机械接口坐标系

此属性是位姿型,它指出工具坐标系相对于机械接口坐标系的变换值。

6.2.3 机器人辅助装置资源

6.2.3.1 辅助属性模型

从机器人臂资源分离出的所有物理资源均作为辅助装置资源来模拟。辅助装置对象描述了物理资源的属性。

对象: 辅助装置

属性: 本地控制(TRUE,FALSE)

属性: 装置供电(TRUE,FALSE)

属性: 装置校准(CALIBRATED,NOT—CALIBRATED,CALIBRATING)

6.2.3.2 本地控制

此属性是布尔型,它表示辅助装置资源是(TRUE)否(FALSE)为本地控制。

6.2.3.3 装置供电

此属性为布尔型,它表示是否已对辅助装置供电。

6.2.3.4 装置校准

此属性指出作为一个整体,辅助装置资源是否被校准。若实际辅助装置无须进行校准,则此属性不存在。

6.3 映射到程序调用的机器人特定对象

6.3.1 机器人专用程序调用

程序调用模型的概念可扩展为包容在同一 VMD 中两个相关程序调用的相互配合。在这样一对相互配合的程序调用中,一个程序调用被称为施控程序调用,另一个被称为受控程序调用。将两个程序调用模拟为具有一个交互处理通信通道,通过该通道,它们传送报文以提供数据及过程的同步。除引起程序调用状态改变外,此报文对远程 MMS 用户是不可见的。这两种程序调用类型间的关系是“多”对“一”

的关系。一个施控程序调用能控制多个受控程序调用；一个受控程序调用在任何特定时间只能被一个施控程序调用控制。

6.3.2 程序调用对象模型的扩展

GB/T 16720.1 给出的程序调用模型需由下列属性来扩展。

对象：程序调用

属性：全部 MMS 属性

属性：错误代码——整数

属性：控制(CONTROLLING, CONTROLLED, NORMAL)

约束：控制=CONTROLLING

属性：受控程序调用引用表

属性：程序定位——字符串

属性：运行方式(FREE—RUN, CYCLE—LIMITED, STEP—LIMITED)

约束：运行方式=CYCLE—LIMITED

属性：保持循环计数——整数

约束：运行方式=STEP—LIMITED

属性：保持步进计数——整数

约束：控制=CONTROLLED

属性：施控程序调用的引用

6.3.2.1 错误代码

此属性是整数型，它标识程序调用执行的最终错误记录。零表示无错误记录。与此属性复位的方法相同，其他值的意义也是本地事务。

6.3.2.2 控制

此属性表示程序调用“CONTROLLING”是否被规定为与其他的程序调用相耦合而正在控制中，或耦合后程序调用“CONTROLLED”是否正常地接收由其他程序调用(施控程序调用)来的控制信息。若两种情况都不适用，则此属性值应为 NORMAL。

当两个程序调用以此方式耦合时，对被施控程序调用接收的“开始”和“恢复”服务请求，服务指示能够引起与施控程序调用耦合的受控程序调用的状态转移(见 7.3.3.3 和 7.3.3.5)。施控程序调用的报告结果能够包含反映在受控程序调用上服务请求效果的信息。

目的是将一个指挥机器人活动的任务程序的执行与施控程序调用相关联，并将一个物理装置的硬件控制程序操作与受控程序调用相关联。这就允许在两类程序调用间操作的逻辑异步，这是机器人操作所需的一个特征。可有三种方式去停止机器人，即停止机器人臂控制程序；当没有移动指令生成时，停止能直接停止臂的任务程序以及停止整个系统(见 7.3.3.4)。

一个施控程序调用能够控制几个受控程序调用，例如，R_ARM 是一个预定义的受控程序调用。工具管理或一个正在使用中的辅助装置可用另一个受控程序调用来模拟。这两个程序调用可能同时被同一施控程序调用所控制，该施控程序经常是一个用户写入程序。

注：也可以不使用施控及受控程序调用对象。在某些应用中，有关机器人臂可使用在建立正常程序调用过程中的引用域表内包含 R_ARM 域来表示已经足够了。可是，这样做就使停止任务程序和停止机器人臂运动间的区别能力受到妨碍。

6.3.2.3 受控程序调用的引用表

当程序调用的控制属性等于 CONTROLLING 时，此属性才存在。这是一个对其他程序调用的引用表，该程序调用具有其控制属性值等于 CONTROLLED，并且其控制程序调用的引用属性为本控制程序，此表可以为空。

6.3.2.4 程序定位

此属性是字符串型,只有当程序调用具有其控制属性等于 CONTROLLING 时才存在。此属性的使用是执行程序的首选;当使用时,程序定位属性的格式在 PICS 中(见 9.4)描述。若该属性存在,它标识当程序调用处于 RUNNING 状态时正被执行或将被执行的程序调用源代码的行。

6.3.2.5 运行方式

此属性只有当程序调用具有其控制属性等于 CONTROLLING 时才存在。若存在,它表示程序调用执行是如何管制的。如果此属性的值是 FREE—RUN(自由运行),程序调用将保持在 RUNNING 状态,直到某些本地的或远程的事件出现而引起其停止执行。若此属性的值是 CYCLE—LIMITED(有限循环),则显式计数器保持有一个包含被执行的循环数。当该计数器达到零时,程序调用停止执行并返回到 IDLE 状态。对于支持这种方式的执行程序,此属性可以取值 STEP—LIMITED(有限步数)。而当程序调用开始或恢复时,应给定执行步数。当被执行的步数已被减少到零,则施控程序调用应自动转入 STOPPED 状态。此方式通常只用于排除故障。在 PICS 中(见 9.4)需表明执行程序是否支持 STEP—LIMITED 运行方式。

6.3.2.6 保持循环计数

此属性为整数型,它是程序调用保持被执行的循环数。若程序调用不是循环计数方式,则此属性不必规定。

6.3.2.7 保持步进计数

此属性是整数型,它是当运行方式处于有限步数时,保持被执行的步数。除非程序调用为 STEP—LIMITED(有限步数)运行方式,则此属性不必规定。

6.3.2.8 施控程序调用的引用

只有当控制属性值是 CONTROLLED 时,此属性才存在。若存在,它指出对该程序调用充当施控程序调用的程序调用。被引用的程序调用应具有控制属性值等于 CONTROLLING,并且应在其受控程序调用属性的引用表中包含对此受控程序的调用的引用。如若没有这样的程序调用存在,此属性值应是 UNCONTROLLED(未受控)。

6.3.3 程序调用状态图的扩充

对于控制属性等于 CONTROLLING 的程序调用,机器人调用状态图(见 GB/T 16720.1 中 11.1.3)应被扩充。IDLE 状态被细分以说明施控程序调用的可被或不被选择。在图 10 中给出了 IDLE 状态图。

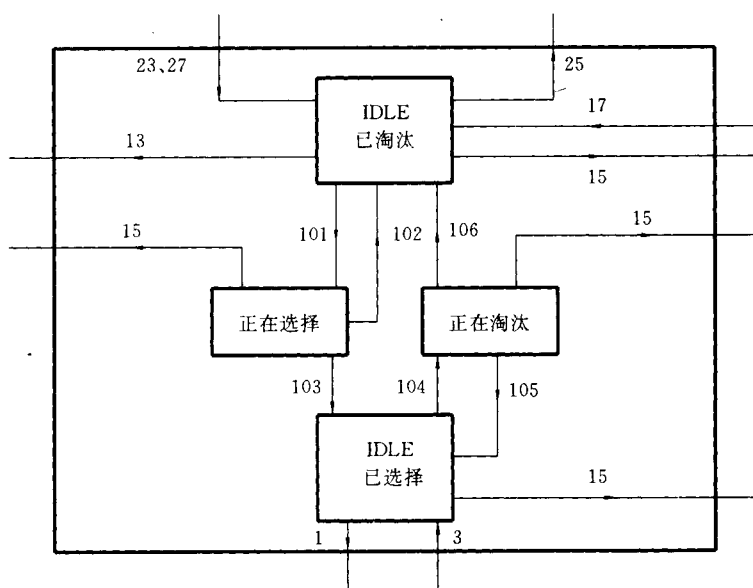


图 10 IDLE 状态图

编号在 100 以下的转移指的是被描述在 GB/T 16720.1 中 11.1.3 中的转移。在本图中增加的转移

是:

- 101 选择指示
- 102 选择应答(—)
- 103 选择应答(+)
- 104 选择指示,参数=淘汰
- 105 选择应答(—)
- 106 选择应答(+)

6.3.4 机器人操作状态和 VMD

6.3.4.1 总则

机器人操作模型(见第5章)的主要状态是与机器人臂的以及能运动的辅助装置的运动参数有关,并和控制机器人臂的被选施控程序调用有关。被反映到 VMD 状态的装置集合不需要包括机器人系统的所有装置,但是必须包括机器人臂的安全联锁。执行者应在 PICS 中提供包括在 VMD 状态中的装置的专用表。在下面的状态说明中,术语“机器人在运动”表示在这套装置中任何一个装置在运动。

机器人操作状态模型与 VMD 元素间的相互关系是取决于这些状态是如何被引入的;此模型亦适用于局部方式影响转移时。第5章中的机器人操作状态模型和机器人臂、辅助装置和控制它们的程序调用等状态间的相互关系如下所述。

6.3.4.2 ROBOT—IDLE(机器人空闲)状态

在 VMD 中机器人没有施控程序调用。‘被选程序调用引用’的 VMD 属性值为 NONE。R_ARM 程序调用具有施控程序调用引用属性等于 UNCONTROLLED。否则 VMD 完全可操作(不空闲)。

6.3.4.3 ROBOT—LOADED(机器人已装载)状态

在 VMD 中机器人有一个或多个施控程序调用。被选程序调用引用的 VMD 属性值为 NONE。R_ARM 程序调用具有施控程序调用引用属性等于 UNCONTROLLED。

6.3.4.4 ROBOT—READY(机器人准备完毕)状态

被选程序调用引用的 VMD 属性具有一施控程序调用的引用值。R_ARM 程序调用对施控程序调用的引用属性,引用了在 IDLE 状态中同一程序调用。

6.3.4.5 ROBOT—EXECUTING(机器人执行)状态

机器人已处于操作中,运动是使能运动,并且臂可能正在动作。由被选程序调用引用的 VMD 属性所引用的程序调用是处于 RUNNING 状态或 STOPPED 状态。

6.3.4.6 ROBOT—MOTION—PAUSED(机器人动作暂停)状态。

机器人已经停止运动。由被选程序调用引用的 VMD 属性所引用的程序调用是处于 RUNNING 状态或 STOPPED 状态。

6.3.4.7 MANUAL—INTERVENTION—REQUIRED(要求手动介入)状态

在本状态中,机器人已停止运动。VMD 的逻辑状态是 NO—STATE—CHANGES—ALLOWED(不得改变状态),LIMITED—SERVICES—PERMITTED(允许有限服务)或 SUPPORT—SERVICES—ALLOWED(允许支持服务)。VMD 的物理状态是 NEEDS—COMMISSIONING(必须试运行)。VMD 的其他属性状态是本地事务。

6.3.5 机器人操作状态转移说明

6.3.5.1 总则

本条款描述与机器人操作状态模型(见图8)转移相对应的 MMS 服务序列。MMS 服务与机器人操作状态模型之间的相互关系取决于被选程序调用(被 VMD 属性引用的程序调用)是通过 MMS 建立程序调用服务动态建立的,还是在机器人系统控制器中预先存在并已定义的。可由 MMS 服务步骤完成如下状态转移:

6.3.5.2 装载—1

若机器人并不包含一个预定的其控制属性值等于 CONTROLLING 的程序调用,则机器人将通过建立一个控制属性等于 CONTROLLING 的程序调用,从 ROBOT—IDLE 状态运行到 ROBOT—LOADED 状态。所使用的 MMS 服务是“建立程序调用”。若机器人含有一个预定的控制属性值等于 CONTROLLING 的程序调用,则机器人在通电时将直通到 ROBOT—LOADED 状态,而不发生此项转移。

在后一种情况中,机器人也可直通到 ROBOT—READY 状态。注意,当此类程序调用建立时,亦可能域已先期建立并装载,在机器人状态图中,并不反映这个过程。

6.3.5.3 清除—2

通过用“删除程序调用”服务删除“施控程序调用”,机器人的运行由 ROBOT—LOADED 状态既可保持状态不变,也可转入 ROBOT—IDLE 状态。如果有多个“施控程序调用”存在,在后一个程序调用被删除前,每次删除,机器人仍留在 ROBOT—LOADED 状态。最后一个“程序调用”的删除,将使机器人返回到 ROBOT—IDLE 状态。“VMD 复位”服务也能引起此项转移。

此项转移只在以前发生过装载转移时才发生。亦应注意对程序调用组成部分的域随后发生的删除对机器人操作状态图没有影响。

6.3.5.4 赋值—3

如果机器人不包含一个预定的施控程序调用的“程序调用”和由 VMD 的被选程序调用引用属性所引用的“程序调用”,则机器人将通过选择一个控制它的“程序调用”的步骤,由 ROBOT—LOADED 状态转入 ROBOT—READY 状态。该被选程序调用的控制属性值应等于 CONTROLLING。所采用的 MMS 服务是“选择”服务。若预定的“程序调用”存在,且被 VMD 的被选程序调用引用属性所引用,则机器人将在通电时直接进入 ROBOT—READY 状态,并且将不发生此项转移。

6.3.5.5 去值(Deassign)—4

机器人是通过选弃,由 VMD 被选程序调用引用属性所引用的施控程序调用而由 ROBOT—READY 状态被转入 ROBOT—LOADED 状态的。所用的 MMS 服务是给定参数为空的“选择”服务(见 7.4.4)。

当先有赋值转移发生时,本转移才能发生。

6.3.5.6 程序开始—5

机器人通过开始施控程序调用,由 ROBOT—READY 状态转入 ROBOT—EXECUTING 状态,该施控程序调用是由 VMD 的被选程序调用引用属性所引用的。所使用的 MMS 服务是“开始”服务。

6.3.5.7 执行结束—6

机器人通过程序调用的正常编程操作,由 ROBOT—EXECUTING 状态转入 ROBOT—LOADED 状态。

6.3.5.8 程序复位—7

机器人可通过复位施控程序调用,从 ROBOT—MOTION—PAUSED 状态转入 ROBOT—READY 状态,该施控程序调用是由 VMD 的被选程序调用引用属性所引用的。所使用的 MMS 服务是“复位”。

6.3.5.9 暂停—8

机器人是通过禁止运动,即设定使能运动为“FAUSE”,由 ROBOT—EXECUTING 状态转入 ROBOT—MOTION—PAUSED 状态的。这通常是通过停止受控程序调用 R_ARM 来完成。所使用的 MMS 服务是“停止”。

6.3.5.10 继续—9

机器人通过恢复施控程序调用,由 ROBOT—MOTION—PAUSED 状态转入 ROBOT—EXECUTING 状态,该施控程序调用是 VMD 的被选程序调用引用属性所引用的。所使用的 MMS 服务是

“恢复”。

6.3.5.11 清除(异常终止)—10

机器人由 ROBOT—MOTION—PAUSED 状态到 ROBOT—IDLE 状态的转移,除施控程序调用没有完成最后一个循环外,其他均与清除相同。所使用的 MMS 服务是机器人专用的“VMD 复位服务”。

6.3.5.12 清除(舍弃)—11

机器人由 ROBOT—READY 状态到 ROBOT—IDLE 状态的转移,除施控程序调用没有完成最后一个循环外,其他均与清除相同。所使用的 MMS 服务是机器人专用的“VMD 复位服务”。

6.3.5.13 应急动作—12

通过某些局部活动,如触发紧急停止电路,机器人可由任何状态转入 MANUAL—INTERVENTION—REQUIRED 状态。在 RUNNING 状态中的任何程序调用都可被从该状态中转出。这些程序调用的结果状态则是本地事务。这种转移也能通过机器人专用的“VMD 停止服务”来实现。

6.3.6 校准步骤

一个物理资源的校准状态是作为模拟物理资源的域的一个属性来模拟的。该属性值有三个可能: NOT—CALIBRATED, CALIBRATING 和 CALIBRATED。为了完成物理资源的校准存在一个包含校准功能的预定域和一个使用该域的预定程序调用。当执行 R_CAL(机器人校准)程序调用时,代表物理资源装置的域校准属性应该被设置为 CALIBRATING。当完成时, R_CAL 程序调用应被设置为 IDLE 状态。如果校准功能已完成,装置校准属性应设置为 CALIBRATED。否则,为 NOT—CALIBRATED。如果所有此类物理资源都已具有属性 CALIBRATED,则 VMD 属性“全部物理资源已校准”值应为 TRUE。

R_CAL 程序调用与施控程序调用的不同是它是一个正常的程序调用,它的存在不应使机器人进入 ROBOT—LOADED 状态。

6.3.7 子系统激活与不激活(activation and Deactivation)

辅助装置的激活与不激活可模拟为在适当的预定义域中对预定义变量的写入操作,或通过一个预定的程序调用的状态转移来模拟。特别是对机器人臂的电源的应用可被模拟为在机器人臂的预定义域内一个布尔变量。第 8 章制定了选择此类变量名称的导则。

6.3.8 状态报告

6.3.8.1 VMD 对象状态

应使用 MMS 状态服务取得 VMD 的属性值。

6.3.8.2 机器人臂对象状态

应使用 MMS 读服务取得能通过 MMS 得到的机器人臂的各种属性状态,对能被访问的每个属性值定义一个标准化的有名变量对象。使用 MMS 读服务读这个有名变量对象,应提供关联属性的状态。

6.3.8.3 施控程序调用对象状态

在本标准中已定义了 MMS 程序调用对象附加属性。这些属性的值或状态应通过使用 MMS“获取程序调用属性”服务取得。

6.3.9 自主状态报告

在机器人 VMD 内,对象的状态亦可自主地报告。这可通过 MMS“非请求状态”服务的使用或通过 MMS“事件通告”服务的使用来产生。

对于各种事件活动对象和事件条件对象定义了标准化名称。这些标准化名称是用于(但不限于)事件通告服务以报告状态。对于不具有标准化名称的条件,也可生成事件通告。当必须对它们使用“非请求状态”和“事件通告”服务时,本标准不作规定。

6.4 映射到其他 MMS 抽象对象的机器人特定对象的定义

本标准没有定义任何映射到其他 MMS 抽象对象的机器人特定的对象。

6.5 支持其他机器人特定对象的 MMS 新抽象对象的定义

本标准没有定义任何附加的 MMS 抽象对象。

7 机器人专用服务和协议

7.1 机器人专用上下文定义

为了能使用一种只包含 ACSE 和 MMS 的应用作为 ASE, 本标准使用了在 GB/T 16720.2—1996 的 17.12 中所定义的对象标识符值和对象描述符值。

7.2 机器人专用抽象语法定义

本标准指定 ASN.1 对象标识符值

{GB/T 16720.3part(3)mms—robot—syntax—version1(1)} 为本章定义的抽象语法。

7.3 MMS 服务的使用

7.3.1 机器人专用 ASN.1 模块的定义

制定的 MMS 服务和协议广泛用于各种制造装置。本章对在 GB/T 16720.2 中标识为要求由伴同标准定义的元素, 定义了机器人的服务和协议。当商定使用本标准所定义的抽象语法时, 需使用这些定义。本章进一步明确了在机器人中是如何使用 MMS 服务的。

所有在本标准中提供的 ASN.1 的定义是 ASN.1 模块“GB/T 16720.3MMS—ROBOT—1”的一部分。为了更易于阅读本文件, 省略了由每个 ASN.1 定义提供的作为模块一部分的开始语句和结束语句。提供的每个 ASN.1 定义, 在定义的开始固定地包含有语句:

GB/T 16720.3—MMS—ROBOT—1

{GB/T 16720.3part(3)mms—robot—module—version1(2)}

DEFINITION ::= BEGIN

在定义的结尾包含有关键字“END”。

注: GB/T 16720.3—MMS—ROBOT—1 表示 MMS(GB/T 16720.3) 机器人伴同标准的 1 号版本。

在本章中所用的许多术语和缩略语使用了 MMS 服务和协议中描述的词汇(见 GB/T 16720.1—1996 第 5 章及 GB/T 16720.2—1996 第 5 章)。尤其是在本标准解释服务表的通用原则时, 要使用 GB/T 16720.1—1996 第 5 章。

IMPORTS MMSpdu,

ParameterSupportOptions,

ServiceSupportOptions,

Identifier,

Integer 16,

StatusResponse

FROM MMS—General—Module—1

{GB/T 16720.2part(2)mms—general—module—version1(2)};

7.3.2 VMD 支持服务和协议

7.3.2.1 “状态”服务和“非请求状态”服务

MMS 状态服务为应答机器人提供了一种表示其 VMD 一般状态的方法。“非请求状态”服务为机器人对主机提供一种无需状态服务请求即可表示其 VMD 一般状态的方法, 何时发送“非请求状态”是一种本地事务。

两种服务考虑到在请求及应答中被包含的伴同标准专用信息。

本标准并未规定对“状态”请求增加任何附加参数。

在状态服务的应答和确认原语中, 以及在“非请求状态”服务的请求和指示原语中应使用机器人状态细目参数。

7.3.2.1.1 机器人状态细目参数

在表 2 中表示了机器人状态细目参数的结构：

表 2 机器人状态细目

参数名	请求应答	指示确认
机器人 VMD 状态	M	M(=)
机器人专用状态	M	M(=)
机器人专用状态屏蔽码	U	U(=)
被选程序调用	M	M(=)

7.3.2.1.1.1 机器人 VMD 状态

此参数是整数型,应传送 VMD 的机器人 VMD 状态属性值,在 6.3.4 和在 5.3.1.3 中分别定义和描述了机器人 VMD 状态属性。

7.3.2.1.1.2 机器人专用状态

此参数是位串型,应传送机器人专用属性集的布尔值。

Safety Interlocks Violated(违反安全联锁)

Any Physical Resource Power On(物理资源供电)

All Physical Resources Calibrated(全部物理资源已校准)

Local Control(本地控制)

在 6.1.3.2 中定义了这些属性,在 5.3.1.4 中描述了本地控制。

7.3.2.1.1.3 机器人专用状态屏蔽码(Mask)

此任选参数是位串型,应传送在机器人专用状态参数中对应位的有效性。若位串中此位是 1,在机器人专用状态参数中对应位是有效的。若位串中此位是 0,在机器人专用状态参数中对应位是无用的。此参数可缺省是当各值全部为 1 时。

7.3.2.1.1.4 被选程序调用

此参数表示机器人系统的施控程序调用已被选作程序调用。若无被选程序调用,则此参数值为 NONE。

7.3.2.1.2 机器人专用状态协议

CS 状态请求、CS 状态应答及 CS 未确认状态等的抽象语法规则如下,并在随后段落中加以描述。GB/T 16720.2 描述了在本条中未提供明确表明的所有其他参数的导出。

CS—Status—Request::=NULL

CS—Status—Response::=RobotStatusDetail

CS—Unsolicited Status::=RobotStatusDetail

RobotStatusDetail::=SEQUENCE{

```

    robotVMDState          [0]IMPLICIT RobotVMDState,
    robotSpecificStatus     [1]IMPLICIT RobotSpecificStatus
    robotSpecificStatusMask [2]IMPLICIT ROBOTSpecificStatus DEFAULT '1111'B,
    selectedProgramInvocation CHOICE{
        ProgramInvocation    [3]IMPLICIT Identifier,
        noneSelected         [4]IMPLICIT NULL
    }

```

}

RobotVMDState::=INTEGER{

```

    robot—idle             (0),
    robot—loaded           (1),
    robot—ready            (2),

```

```

robot—executing          (3),
robot—motion—paused      (4),
manualInterventionRequired (5)
}
RobotSpecificStatus ::= BITSTRING {
    SafetyInterlocksViolated      (0),
    anyPhysicalResourcePowerOn    (1),
    allPhysicalResourcesCalibrated (2),
    localControl                  (3)
}

```

7.3.3 程序调用管理服务和服务协议

7.3.3.1 通用结构

本条款描述了由程序调用管理操作的机器人专用的服务和协议。为了包容两个相关程序调用相互配合的概念,扩大了程序调用模型。在一对相互配合的程序调用中,一个程序调用被称为施控程序调用,另一个被称为受控程序调用。两个程序调用通常被模拟为具有一个可用的交互处理通信通道,通过通信通道,它们传递提供数据及过程同步的报文。然而,除非引起程序调用状态变更,通常,这种报文对MMS 远程用户是不可见的。

扩展程序调用特性模型是为了说明施控程序调用接收一个开始或恢复指令时,作为其服务步骤的一部分,它应在相关的受控程序调用内引起转移。施控程序调用应在其对服务请求的应答中插入其适合于受控程序调用的应答信息。那些在受控程序调用指挥下服务的服务步骤,通常不对其他程序调用传播。

程序调用间的关系是“多”对“一”的关系;一个施控程序调用能控制多个受控程序调用;一个受控程序调用在任何特定时间只能由一个施控程序调用所控制。见 7.4.4 的选择服务。

7.3.3.2 建立程序调用服务

需要扩展建立程序调用服务请求,以包容正被建立程序调用的控制属性规范。这应通过给建立程序调用请求附加一个参数来完成。

7.3.3.2.1 CS 建立程序调用请求

在表 3 中表示了 CS 建立程序调用请求参数的结构。

表 3 CS 建立程序调用请求参数

参 数 名 称	请 求	指 示
控 制	M	M(=)

7.3.3.2.1.1 控制

此参数应指出程序调用的控制属性值。本参数可以具有值 CONTROLLING、CONTROLLED 或 NORMAL。

7.3.3.2.2 CS 建立程序调用应答

对于建立程序调用应答,本标准规定不附加各种参数。

7.3.3.2.3 扩充的服务步骤

应该实施规定在 GB/T 16720.1 中的服务步骤。若该步骤成功地完成,则控制属性应按照在服务请求中控制参数的值来设置,并且错误代码属性值应被设置为零。

a) 若在服务请求中控制参数是 CONTROLLING,受控程序调用引用表的属性值应被设置为 NONE。程序位置属性应被设置为一个空串,而运行方式属性应被置为 FREE—RUN。

b) 若在服务请求中的控制参数是 CONTROLLED,施控程序调用引用的属性应被设置为 UN-

CONTROLLED。

7.3.3.2.4 建立程序调用协议

CS 建立程序调用请求的抽象语法规则如下,并在随后的段落中加以描述。GB/T 16720.2 中描述了在本条款中未提供明确表明的其他参数的导出。

```
CS—CreateProgramInvocation—Request ::= INTEGER {
    normal          (0),
    controlling     (1),
    controlled      (2)
}
```

```
CS—CreateProgramInvocation—Response ::= NULL
```

7.3.3.3 开始服务

开始服务允许客户请求应答机器人初始化程序调用的执行。通过使用 CS 开始请求参数,扩充了开始服务的服务步骤。服务步骤以程序调用的控制属性是 CONTROLLING, CONTROLLED 或是 NORMAL 和客户是否具有 R_CTRL 信志的控制为条件。

7.3.3.3.1 CS 开始请求

在表 4 中表示了 CS 开始请求参数的结构。仅当程序调用的控制属性值为 CONTROLLING 时,CS 开始请求参数应是非空。

表 4 CS 开始请求参数

参数名称	请求	指示
开始位置	U	U(=)
运行方式	C	C(=)
无限定	S	S(=)
循环计数	S	S(=)
步进计数	S	S(=)

7.3.3.3.1.1 开始位置

该任选参数是字符串型,仅当程序调用具有其控制属性值等于 CONTROLLING 时存在。若控制属性值等于 CONTROLLING,该参数的用法是用户任选。若使用该参数,则其格式被描述在 PICS 中(见 9.4)。该参数的值表示何处是程序调用开始执行的“开始位置”。若从服务请求中省略了开始位置参数,则应在程序的第一步开始执行。第一程序步或缺省第一步的含义是本地事务。对于任何在 PICS(见 9.4)中的执行程序均须定义这个含义。

7.3.3.3.1.2 运行方式

只有当程序调用的控制属性值是 CONTROLLING 时,此参数存在。它表示程序调用的运行方式属性值。按照被选程序调用的运行方式应存在下列参数之一。

7.3.3.3.1.3 无限定

此参数是空型,若运行方式属性值被设置为 FREE—RUN,应选择此参数。

7.3.3.3.1.4 循环计数

此参数是整数型,若运行方式属性值被设置为 CYCLE—LIMITED,应选择此参数。并且将“保持循环计数”属性值设置为此参数值。该参数值大于零。

7.3.3.3.1.5 步进计数

此参数是整数型,若运行方式属性值被设置为 STEP—LIMITED,应选择此参数。并将“保持步进计数”属性值设置为此参数值。此参数值应大于零。在 PICS 中(见 9.4)应定义 STEP—LIMITED 运行方式的执行程序。若不被支持,则不得选用此选择。

7.3.3.3.2 CS 开始应答

本标准对“开始应答”不定义附加参数。

7.3.3.3.3 扩充的服务步骤

应将下列服务步骤应用于开始服务。

a) 应实施在 GB/T 16720.1—1996 的 11.4.2 中的服务步骤错误校验。
 b) 若请求 MMS 用户并不持有 R_CTRL 信志,应返回一个结果(一)。
 c) 若开始服务请求的程序调用命名参数所标识的程序调用控制属性不具有值 CONTROLLING,则应检查开始位置和运行方式参数是否存在。若存在其中任何一个,则返回一个结果(一)。否则应实施 GB/T 16720.1—1996 的 11.4.2 的开始服务步骤的余下部分,并且跳过本步骤的其余部分。

d) 若由开始服务请求的程序调用命名参数所标识的程序调用的控制属性等于 CONTROLLING,则检验程序调用是否被 VMD 的被选程序调用引用属性所引用。如果不是,则返回一个结果(一)并跳过本步骤的其余部分。否则对此程序调用的受控程序调用引用属性表上的每个元素,应实施下列各步:

1) 检验被引用的程序调用的控制属性值是否等于 CONTROLLED 以及其施控程序调用引用属性是否引用了该程序调用。

2) 若被引用的受控程序调用是在 IDLE 状态,对这个被引用的受控程序调用实施开始步骤,并且将它运行到 RUNNING 状态。

3) 若被引用的受控程序调用是在 STOPPED 状态,则对于这个引用的受控程序调用实施恢复步骤,并将其运行到 RUNNING 状态。

4) 若受控程序调用不能进入 RUNNING 状态,则对表上每个先前的受控程序调用,实施停止步骤。若任何停止步骤失效,如何处理此状态则是一个本地事务。可能采用 VMD 停止步骤是一个适当的方法。最后,对此服务请求返回一个结果(一)并跳过本步骤的其余部分。

e) 若由开始服务请求的程序调用命名参数所标识的程序调用的控制属性等于 CONTROLLING,则施控程序调用的属性应被设置如下:

1) 若在服务请求中选择了无限定参数,则程序调用的运行方式属性应被设置为 FREE—RUN。

2) 若在服务请求中选择了循环计数参数,则程序调用的运行方式属性应被设置为 CYCLE—LIMITED,并将程序调用的保持循环计数属性设置为循环计数参数值。

3) 若在服务请求中选择了步进计数参数,则程序调用的运行方式属性应被设置为 STEP—LIMITED,并将程序调用的保持步进计数属性设置为步进计数参数值。

4) 若在服务请求中存在开始位置参数,则此参数值将限制程序调用的程序控制信息,以便为程序调用的执行提供一个起始点。用来传递起始位置的表示法是本地事务,并且通常将取决于所使用的编程语言,在 PICS(见 9.4)中描述了开始位置参数所使用的格式。

5) 若在服务请求中不存在开始位置参数,则在程序调用的执行中,应使用开始位置缺省值。

f) 若由开始服务请求的程序调用命名参数所标识的程序调用的控制属性值等于 CONTROLLED,并且程序调用被约束于代表机器人臂的一个域中,则可设置域的使能运动属性为 TRUE。

g) 实施 GB/T 16720.1—1996 的 11.4.2 中的开始服务步骤的其余部分。

7.3.3.3.4 开始协议

下面规定了 CS 开始请求的抽象语法,并在随后的段落中描述。GB/T 16720.2 中描述了在本条款中未提供明确表明其他参数的导出。

```
CS—Start—Request ::= CHOICE {
    normal          NULL,
    controlling     SEQUENCE {
        StartLocation    [0]IMPLICIT VisibleString OPTIONAL,
        StartCount       [1]StartCount DEFAULT cycleCount 1
    }
}
```

```

}}
StartCount ::= CHOICE {
    noLimit      [0] IMPLICIT NULL,
    cycleCount   [1] IMPLICIT INTEGER,
    stepCount    [2] IMPLICIT INTEGER
}

```

CS—Start—Response ::= UNLL

注：由于《协议规范》标准 7.1 中确认请求 PDU 产生式的注解字段的约束，若选用 CS 开始请求的正常选择，则不必发送 CS 开始请求语句。

7.3.3.3.4.1 施控程序调用

若程序调用的控制属性具有值 CONTROLLING，则施控选择应在 CS 开始请求中作出；若控制属性是 CONTROLLED 或 NORMAL，则应进行正常选择。

7.3.3.3.4.2 运行方式

在开始计数类型内所作的选择应推断出运行方式参数值。无限定选择表示运行方式应被设置为 FREE—RUN。循环计数选择表示运行方式应被设置为 CYCLE—LIMITED 并保持循环计数属性被设置为所指示的值。步进计数选择表示运行方式应被设置为 STEP—LIMITED 并保持步进计数属性被设置为所指示的值。

对于不支持 STEP—LIMITED 运行方式的系统，开始计数选择值应被限于“无限定”([0])或“循环计数”([1])。

7.3.3.4 停止服务

对停止应答的停止请求本标准定义不附加参数。

7.3.3.4.1 扩充的服务步骤

对停止服务应采用下列的服务步骤：

- 实施 GB/T 16720.1—1996 中 11.5.2 的停止服务步骤的错误校验。
- 若请求 MMS 用户并不持有 R_CTRL 信志，应返回一个结果(—)。
- 若程序调用的控制属性是 CONTROLLED，并且程序调用约束于代表机器人臂的一个域中，则设置使能运动属性为 FALSE。
- 实施 GB/T 16720.1—1996 中 11.5.2 的停止服务步骤的其余部分。

停止机器人的运行，可用两种停止服务的方式之一来进行。若施控程序调用被停止，控制机器人操作的任务程序将被停止。在其后某个时间内，机器人完成了施控程序调用发出的最后指令，纵使表示着机器人臂控制程序的受控程序调用保持在 RUNNING 状态，亦将中止机器人运动。若表示机器人臂控制程序的受控程序调用被停止，则机器人臂的活动立即中止，即在完成最后程序步之前，以这种方式维持最后编程路径。“VMD 停止”服务则是提供了停止机器人运行的第三种方法。

7.3.3.4.2 停止协议

对于规定在 GB/T 16720.1—1996 中的停止协议，本标准没有定义任何语法扩充。

CS—Stop—Request ::= NULL

CS—Stop—Response ::= NULL

7.3.3.5 恢复服务

恢复服务允许应答机器人从一个先前已停止的执行点上恢复执行一个程序调用。

7.3.3.5.1 CS 恢复请求

在表 5 中表示了 CS 恢复请求参数结构。

表 5 CS 恢复请求参数

参数名称	请求	指示
恢复类型	C	C(=)
连续方式	S	S(=)
变更方式	S	S(=)
无限定	S	S(=)
循环计数	S	S(=)
步进计数	S	S(=)

7.3.3.5.1.1 恢复类型

仅当程序调用的控制属性是 CONTROLLING 时,此参数存在。如果存在,它表示程序调用是否应按程序调用变或不变的运行方式属性执行恢复。

7.3.3.5.1.2 连续方式

若程序调用恢复,并按不变的运行方式属性执行,应选择此参数。

7.3.3.5.1.3 变更方式

若程序调用恢复,并对一个新值按变更运行方式属性执行,应选择此参数。若选择了此参数则应存在下列参数之一。

7.3.3.5.1.4 无限定

此参数是空类型。若运行方式的属性值是被变更为 FREE—RUN,应选择此参数。

7.3.3.5.1.5 循环计数

此参数是整数型,若运行方式属性值被变更为 CYCLE—LIMITED,应选择此参数。并应将保持循环计数属性值设置为此参数值。该参数值应大于零。

7.3.3.5.1.6 步进计数

此参数是整数型,若运行方式属性值被变更为 STEP—LIMITED,应选择此参数,并应将保持步进计数属性值设置为该参数值。该参数值应大于零。在 PICS 中(见 9.4)定义了 STEP—LIMITED 运行方式的执行程序。如果不支持此参数,则不得选用此选择。

7.3.3.5.2 CS 恢复应答

对于恢复应答本标准定义不附加参数。

7.3.3.5.3 扩充的服务步骤:

恢复服务应采用下列服务步骤。

- 实施 GB/T 16720.1—1996 中 11.6.2 的服务步骤的错误校验。
- 若请求 MMS 用户并不持有 R_CTRL 信志,应返回一个结果(—)。
- 若程序调用的控制属性没有值 CONTROLLING,则检查恢复类型参数是否不存在。如果存在,则返回一个结果(—)。如不存在,则应实施 GB/T 16720.1—1996 中 11.6.2 恢复服务步骤的其余部分,并跳过本步骤的其余部分。

d) 若恢复服务请求的程序调用命名参数所标识的程序调用的控制属性等于 CONTROLLING,则对该程序调用的受控程序调用引用属性表上的每个元素实施下列各步:

1) 检查被引用的程序调用的控制属性值是否等于 CONTROLLED,并且它的施控程序调用的引用属性是否引用了该程序调用。

2) 若被引用的受控程序调用是在 IDLE 状态,则对被引用的受控程序调用实施一个开始步骤,使它进入 RUNNING 状态。

3) 若被引用的受控程序调用是在 STOPPED 状态,则对被引用的受控程序调用实施一个恢复步骤,使它进入 RUNNING 状态。

4) 若受控程序调用不能进入 RUNNING 状态,则对表中每个以前的受控程序调用实施停止

步骤。若其任何停止步骤失效,如何处理此状况则是一个本地事务。采用 VMD 停止步骤可能是适当的方法。最后,对此服务请求返回一个结果(一)并跳过本步骤的其余部分。

e) 若选择了连续方式,则实施下列校验:若程序调用的运行方式属性值是 CYCLE—LIMITED,且保持循环计数属性值是零,或若运行方式属性值是 STEP—LIMITED,而保持步进计数属性值是零,则应返回一个结果(一)。除上述情况外,则应实施描述在 GB/T 16720.1—1996 的 11.6.2 中的恢复服务步骤。

f) 若在恢复类型参数中选择了变更方式参数,则程序调用属性应被设置如下:

1) 若在服务请求中选择了“无限定”参数,则程序调用的运行方式属性应置为 FREE—RUN。

2) 若在服务请求中选择了“循环计数”参数,则程序调用的运行方式属性应置为 CYCLE—LIMITED,并将程序调用的保持循环计数属性置为循环计数参数值。

3) 若在服务请求中选择了“步进计数”参数,则程序调用的运行方式属性应置为 STEP—LIMITED,并将程序调用的保持步进计数属性置为步进计数参数值。

g) 实施 GB/T 16720.1—1996 的 11.6.2 中的恢复服务步骤的其余部分。

7.3.3.5.4 恢复协议

CS 恢复请求的抽象语法规则如下,并在随后各段中描述。GB/T 16720.2 中描述了所有在本条中没有提供明确表明的参数的导出。

```
CS—Resume—Request ::= [0]CHOICE {
    normal          NULL,
    controlling      SEQUENCE { CHOICE {
        Continue Mode    [0]IMPLICIT NULL,
        Change Mode      [1]Start Count
    } } }
```

```
CS—Resume—Response ::= NULL
```

注: 由于在 GB/T 16720.2—1996 的 7.1 中的被确认请求 PDU 产生式中注释了字段的约定,若选择了 CS 恢复请求的正常选择,则无需发送 CS 恢复请求。

7.3.3.5.4.1 施控程序调用

若程序调用的控制属性有值 CONTROLLING,则应在 CS 恢复请求中作施控选择;若控制属性是 CONTROLLED 或 NORMAL,则作正常选择。

7.3.3.5.4.2 恢复类型

恢复类型参数值应由 CS 恢复请求内所作的选择中推断。若所作的选择是连续方式,则恢复类型参数值应是连续方式。若所作的选择是变更方式,则恢复类型参数值是变更方式。

7.3.3.5.4.3 变更方式

变更方式参数值应由开始计数类型内所作的选择推断。选择“无限定”表示运行方式应设置为 FREE—RUN。选择“循环计数”表示运行方式应设置为 CYCLE—LIMITED,并将保持循环计数属性设置为该参数的值。选择“步进计数”表示运行方式应设置为 STEP—LIMITED,并将保持步进计数属性设置为该参数的值。

7.3.3.6 复位服务

对复位请求或复位应答本标准未定义附加参数。

7.3.3.6.1 扩充的服务步骤

复位服务应采用下列服务步骤。

a) 实施 GB/T 16720.1—1996 中 11.7.2 的服务步骤的错误校验。

b) 若请求的 MMS 用户并不持有 R_CTRL 信志,应返回一个结果(一)。

c) 实施 GB/T 16720.1—1996 中 11.7.2 的服务步骤的其余部分。

7.3.3.6.2 复位协议

对 GB/T 16720.2 中规定的复位协议,本标准没有定义任何语法扩充。

CS—Reset—Request::=NULL

CS—Reset—Response::=NULL

7.3.3.7 截杀服务

对于截杀请求或截杀应答,本标准未定义附加参数。

7.3.3.7.1 扩充的服务步骤

截杀服务应采用下列服务步骤。

- a) 实施 GB/T 16720.1—1996 中 11.8.2 的服务步骤的错误校验。
- b) 若请求的 MMS 用户并不持有 R_CTRL 信志,应返回一个结果(—)。
- c) 实施 GB/T 16720.1—1996 中 11.8.2 的服务步骤的其余部分。

7.3.3.7.2 截杀协议

对 GB/T 16720.2 中规定的截杀协议,本标准未定义任何语法扩充。

CS—Kill—Request::=NULL

CS—Kill—Response::=NULL

7.3.3.8 删除程序调用服务

对删除程序调用请求或删除程序调用应答,本标准未定义附加参数。

7.3.3.8.1 扩充的服务步骤

删除程序调用服务应采用下列服务步骤。

- a) 实施 GB/T 16720.1—1996 中 11.3.2 的服务步骤的错误校验。
- b) 若请求的 MMS 用户并不持有 R_CTRL 信志,应返回一个结果(—)。
- c) 若被删除的程序调用的控制属性有值 CONTROLLED,并且施控程序调用的引用属性值不是 UNCONTROLLED,则检验被施控程序调用的引用属性引用的程序调用是否处于 IDLE 状态。如果不是,则返回一个结果(—),并跳过本步骤的其余部分。如果是 IDLE 状态,则去掉被删除的程序调用的引用方式,以改变施控程序调用的引用属性所引用的程序调用的受控程序调用的引用属性表。
- d) 若被删除的程序调用的控制属性有值 CONTROLLING,则对出现在受控程序调用引用表上的每个程序调用,用 UNCONTROLLED 来代替施控程序调用引用属性的值。
- e) 实施 GB/T 16720.1—1996 中 11.3.2 的服务步骤的其余部分。

7.3.3.8.2 删除程序调用协议

对 GB/T 16720.2 规定的删除程序调用协议,本标准未定义任何语法扩充。

CS—DeleteProgramInvocation—Request::=NULL

CS—DeleteProgramInvocation—Response::=NULL

7.3.3.9 获取程序调用属性服务

获取程序调用属性服务用于请求应答机器人返回与规定程序调用相关连的全部属性,并应包括所有机器人专用的程序调用属性。

7.3.3.9.1 CS 获取程序调用属性请求

对获取程序调用属性请求,本标准未定义附加参数。

7.3.3.9.2 CS 获取程序调用属性应答

在表 6 中表示了 CS 获取程序调用属性应答参数的结构。

表 6 CS 获取程序调用属性应答参数

参数名称	应答	确认
错误码	M	M(=)
控制	M	M(=)
施控	S	S(=)
程序调用表	M	M(=)
程序位置	C	C(=)
运行方式	M	M(=)
自由运行	S	S(=)
保持循环计数	S	S(=)
保持步进计数	S	S(=)
受控	S	S(=)
施控程序调用	C	C(=)
正常	S	S(=)

7.3.3.9.2.1 错误码

它是整数型参数,应包含程序调用的错误码的属性。

7.3.3.9.2.2 控制

此参数表示程序调用的控制属性。此参数值可以是 CONTROLLING, CONTROLLED 或 NORMAL。按照此属性值,应存在下列参数之一。

7.3.3.9.2.3 施控

若程序调用控制属性值是 CONTROLLING,则应选择此参数值。若选择了该参数,则出现下列附加参数。

7.3.3.9.2.4 程序调用表

此参数应表示与此程序调用有关的所有程序调用的名称,通过给出的施控程序调用引用属性值引用此程序调用。此表可以有多个元素也可以没有元素。

7.3.3.9.2.5 程序位置

它是一个字符串型的参数。如果存在,则表示当程序调用是与顺序编程语言有关时的程序执行位置。对与顺序编程语言无关的程序调用,此参数不存在。在 PICS(见 9.4)中,将描述此参数的用法。

7.3.3.9.2.6 运行方式

此参数表示程序调用的运行方式属性值。按照运行方式值应存在下列参数之一。

7.3.3.9.2.7 自由运行

选择此空参数,应将运行方式标识为 FREE—RUN。

7.3.3.9.2.8 保持循环计数

选择此参数,应将运行方式标识为 CYCLE—LIMITED,而且,参数值应表示保持执行的循环数,并且是程序调用的保持循环计数属性值。

7.3.3.9.2.9 保持步进计数

选择此参数,应将运行方式标识为 STEP—LIMITED。而且,参数值应表示保持执行的步数,并且是程序调用的保持步进计数属性值。在 PICS 中(见 9.4)应定义 STEP—LIMITED 运行方式的执行程序。如果不被支持,则不可作此选择。

7.3.3.9.2.10 受控

如果程序调用的控制属性值是 CONTROLLED,应选取此参数值。如果选取了此参数,将出现下列附加参数。

7.3.3.9.2.11 施控程序调用

仅当程序调用的控制属性值是 CONTROLLED 时,存在此参数。如果存在,它表示一个程序调用的

名称,该程序调用是被其施控程序调用的引用属性值所引用的。如果没有此类程序调用被引用,则此参数值应为 UNCONTROLLED。

7.3.3.9.2.12 正常

如果程序调用的控制属性值是 NORMAL,应选取此参数值。若选取了此参数,则没有附加参数。

7.3.3.9.3 扩充的服务步骤

在 GB/T 16720.1—1996 的 11.9.2 中规定了服务步骤。在应答中应包括机器人专用的属性细目的信息。

7.3.3.9.4 获取程序调用属性协议

下面规定了 CS 获取程序调用属性请求及 CS 获取程序调用属性应答的抽象语法,并在后随的段落中加以描述。在 GB/T 16720.2 中描述了在本条款中没有明确表明的其余参数的导出。

```

CS—GetProgramInvocationAttributes—Request ::= NULL
CS—GetProgramInvocationAttributes—Response ::= SEQUENCE{
    errorCode          [0]IMPLICIT INTEGER,
    control             [1]CHOICE{
        controlling    [0]IMPLICIT SEQUENCE{
            controlledPI [0]IMPLICIT SEQUENCE OF Identifier,
            ProgramLocation [1]IMPLICIT VisibleString OPTIONAL,
            runningMode   [2]CHOICE{
                freeRunning [0]IMPLICIT NULL,
                cycleLimited [1]IMPLICIT INTEGER,
                stepLimited  [2]IMPLICIT INTEGER
            }
        },
        controlled      [1]CHOICE{
            controllingPI [0]IMPLICIT Identifier, — Reference to Controlling
            none           [1]IMPLICIT NULL — uncontrolled
        },
        normal           [2]IMPLICIT NULL
    }
}

```

7.3.3.9.4.1 控制

控制参数值应由在 CS 获取程序调用属性请求的控制元素内所作的选择中推断。如果作出的选择是施控选择,则控制参数值应是 CONTROLLING。如果作出的选择是受控选择,则控制参数值应是 CONTROLLED。如果作出的选择是正常选择,则控制参数值应是 NORMAL。

7.3.3.9.4.2 运行方式

运行方式参数值应从运行方式下所作的选择中推断。如果在服务应答中选用了自由运行方式,则运行方式参数应取值 FREE—RUN。如果在服务应答中选用了有限循环选择,则运行方式参数应取 CYCLE LIMITED 值。如果在服务应答中选用了有限步进选择,则运行方式应取值 STEP LIMITED。对于不支持 STEP—LIMITED 运行方式的系统,则运行方式选择应限于值自由运行([0]),或循环计数([1])。

7.3.4 其他产生式

在本标准中不采用 MMS 通用模块(见 GB/T 16720.2—1996 附录 A.1)所要求的产生式,并被设置为 NULL。在 GB/T 16720.1 和 GB/T 16720.2 中已完全定义了与这些产生式相对应的服务和协议。不再需要对机器人专门扩展。它们可以在本标准定义的抽象语法中直接使用。

CS—Input—Request ::=NULL
CS—Output—Request ::=NULL
CS—InitiateDownloadSequence—Request ::=NULL
CS—DownloadSegment—Request ::=NULL
CS—TerminateDownloadSequence—Request ::=NULL
CS—InitiateUploadSequence—Request ::=NULL
CS—UploadSegment—Request ::=NULL
CS—TerminateUploadSequence—Request ::=NULL
CS—RequestDomainDownload—Request ::=NULL
CS—RequestDomainUpload—Request ::=NULL
CS—LoadDomainContent—Request ::=NULL
CS—StoreDomainContent—Request ::=NULL
CS—DeleteDomain—Request ::=NULL
CS—GetDomainAttributes—Request ::=NULL
CS—DefineEventCondition—Request ::=NULL
CS—DeleteEventCondition—Request ::=NULL
CS—GetEventConditionAttributes—Request ::=NULL
CS—ReportEventConditionStatus—Request ::=NULL
CS—AlterEventConditionMonitoring—Request ::=NULL
CS—TriggerEvent—Request ::=NULL
CS—DefineEventAction—Request ::=NULL
CS—DeleteEventAction—Request ::=NULL
CS—GetEventActionAttributes—Request ::=NULL
CS—ReportEventActionStatus—Request ::=NULL
CS—DefineEventEnrollment—Request ::=NULL
CS—DeleteEventEnrollment—Request ::=NULL
CS—AlterEventEnrollment—Request ::=NULL
CS—ReportEventEnrollmentStatus—Request ::=NULL
CS—GetEventEnrollmentAttributes—Request ::=NULL
CS—AcknowledgeEventNotification—Request ::=NULL
CS—GetAlarmSummary—Request ::=NULL
CS—GetAlarmEnrollmentSummary—Request ::=NULL
CS—ReadJournal—Request ::=NULL
CS—WriteJournal—Request ::=NULL
CS—InitializeJournal—Request ::=NULL
CS—ReportJournalStatus—Request ::=NULL
CS—CreateJournal—Request ::=NULL
CS—DeleteJournal—Request ::=NULL
CS—GetCapabilityList—Request ::=NULL
CS—Input—Response ::=NULL
CS—Output—Response ::=NULL
CS—InitiateDownloadSequence—Response ::=NULL
CS—DownloadSegment—Response ::=NULL

CS—TerminateDownloadSequence—Response ::=NULL
CS—InitiateUpLoadSequence—Response ::=NULL
CS—UpLoadSegment—Response ::=NULL
CS—TerminateUpLoadsequence—Response ::=NULL
CS—RequestDomainDownLoad—Response ::=NULL
CS—RequestDomainUpLoad—Response ::=NULL
CS—LoadDomainContent—Response ::=NULL
CS—StoreDomainContent—Response ::=NULL
CS—DeleteDomain—Response ::=NULL
CS—Get DomainAttributes—Response ::=NULL
CS—DefineEventCondition—Response ::=NULL
CS—DeleteEventCondition—Response ::=NULL
CS—GetEventConditionAttributes—Response ::=NULL
CS—ReportEventConditionStatus—Response ::=NULL
CS—AlterEventConditionMonitoring—Response ::=NULL
CS—TriggerEvent—Response ::=NULL
CS—DefineEventAction—Response ::=NULL
CS—DeleteEventAction—Response ::=NULL
CS—GetEventActionAttributes—Response ::=NULL
CS—ReportEventActionStatus—Response ::=NULL
CS—DefineEventEnrollment—Response ::=NULL
CS—DeleteEventEnrollment—Response ::=NULL
CS—AlterEventEnrollment—Response ::=NULL
CS—ReportEventEnrollmentStatus—Response ::=NULL
CS—GetEventEnrollmentAttributes—Response ::=NULL
CS—AcknowledgeEventNotification—Response ::=NULL
CS—GetAlarmSummary—Response ::=NULL
CS—GetAlarmEnrollmentSummary—Response ::=NULL
CS—ReadJournal—Response ::=NULL
CS—WriteJournal—Response ::=NULL
CS—InitializeJournal—Response ::=NULL
CS—ReportJournalStatus—Response ::=NULL
CS—CreateJournal—Response ::=NULL
CS—DeleteJournal—Response ::=NULL
CS—GetCapabilityList—Response ::=NULL
CS—EventNotification ::=NULL
CsAdditionalObjectClasses ::=NULL
EN—Additional—Detail ::=NULL
EE—Additional—Detail ::=NULL
JOU—Additional—Detail ::=NULL
CS—GetNameList—Request ::=NULL
CS—GetNameList—Response ::=NULL
CS—Service—Error ::=NULL

7.4 机器人专用服务和协议的定义和使用

7.4.1 附加服务请求和应答

7.4.1.1 通用结构

GB/T 16720.2 允许伴同标准对只在伴同标准中使用的附加服务进行定义。它是通过考虑在已确认服务请求产生式中包容附加服务请求规范,在已确认服务应答产生式中包容附加服务应答规范,在未确认服务的产生式中包容附加的未确认服务规范,以及在服务错误产生式中包容附加服务错误规范来定义附加服务的。

本标准定义机器人的四种附加服务。即 VMD 停止、VMD 复位、选择和变更程序调用属性。在 7.4.2 到 7.4.5 中描述了这些服务。这些服务的抽象语法是由附加服务请求和附加服务应答规定的。没有对附加未确认服务或附加服务错误定义机器人专用的抽象语法。

7.4.1.2 附加服务请求

下面定义了附加服务请求的抽象语法。

```
AdditionalService—Request ::= CHOICE {
    vMDStop      [0] IMPLICIT VMDStop—Request,
    vMDReset     [1] IMPLICIT VMDReset—Request,
    select       [2] IMPLICIT Select—Request,
    alterPI      [3] IMPLICIT AlterProgramInvocationAttributes—Request
}
```

附加服务请求参数应标识服务类型 and 该服务的自变量,所提供的上下文标记标识了服务类型。对每个单个服务定义,通过一种类型的定义,规定了服务的自变量形式,该类型是被附加服务请求产生式所引用的。在附加服务请求选择中,每种服务都是一个已确认的服务。

7.4.1.3 附加服务应答

下面定义了附加服务应答的抽象语法。

```
AdditionalService—Response ::= CHOICE {
    vMDStop      [0] IMPLICIT VMDStop—Response,
    vMDReset     [1] IMPLICIT VMDReset—Response,
    select       [2] IMPLICIT Select—Response,
    alterPI      [3] IMPLICIT AlterProgramInvocationAttributes—Response
}
```

附加服务应答参数应标识服务类型 and 对该服务的应答。所提供的上下文的标记标识了服务类型。对每个单个的服务定义,通过一种类型的定义,规定了对服务的应答形式,该类型是被附加服务应答产生式所引用的。

7.4.1.4 附加服务错误

本标准未定义附加服务错误。

```
AdditionalService—Error ::= NULL
```

7.4.1.5 附加的未确认服务

本标准未定义附加的未确认服务。

```
AdditionalUnconfirmedService ::= NULL
```

7.4.2 VMD 停止服务

VMD 停止服务提供了一种停止应答机器人全部控制作用的方式和一种将机器人置于要求手动介入状态的方式。

VMD 停止服务不相当于在 ISO 10218 中定义的紧急停止。本服务并不提供一个与硬件紧急停止相同的服务。本服务的使用能导致机器人制动器动作,电机断电等类似活动。然而,这些活动并不必需

作为本服务的一部分。

7.4.2.1 结构

在表 7 中表示服务原语分量的结构。

表 7 VMD 停止服务

参数名称	请求	指示	应答	确认
自变量	M	M(=)		
结果(+)			S	S(=)
结果(-)			S	S(=)
错误类型			M	M(=)

7.4.2.1.1 自变量

本服务的自变量中没有参数。

7.4.2.1.2 结果(+)

此参数指示所请求的服务已经成功。成功的结果并不给出服务特定参数。

7.4.2.1.3 结果(-)

结果(-)参数指示服务请求失败,在 GB/T 16720.1 第 17 章中详细定义了错误类型参数,给出了失败的原因。

7.4.2.2 服务步骤

如果请求此服务的 MMS 用户未持有 R_CTRL 信志,则返回结果(-)。反之,如持有,则应答机器人应立即停止机器人的运动和相关联的控制作用,并把机器人置于 MANUAL—INTERVENTION—REQUIRED 机器人 VMD 状态。在 RUNNING 状态中的全部程序调用应从 RUNNING 状态中移出。这些程序调用的状态是本地事务。VMD 逻辑状态应被置为 NO—STATE—CHANGES—ALLOWED。在表 8 中概括了这些活动和对 VMD 属性的影响。

图 8 VMD 属性/VMD 停止

VMD 属性	实施服务后的状态
机器人 VMD 状态	MANUAL—INTERVENTION—REQUIRED
本地控制	未定义
VMD 逻辑状态	NO—STATE—CHANGES—ALLOWED
VMD 物理状态	NEEDS—COMMISSIONING

7.4.2.3 VMD 停止协议

对于附加服务请求和附加服务应答的 VMD 停止选择的抽象语法应由 VMD 停止请求和 VMD 停止应答分别规定。下面规定了这两种类型。GB/T 16720.2 中描述了所有在本条款中没有为其提供明确表明的其他参数的导出。

VMD Stop—Request ::= NULL

VMD Stop—Response ::= NULL

7.4.3 VMD 复位服务

VMD 复位服务提供了一个使应答机器人进入初始化状态的方法。并为客户提供了使请求应答机器人在 VMD 初始化中实施自诊断的能力。在此意义上,所有与物理状态有关的信息将被认作此服务的结果。

在实施一个初始化的例行程序中,或在实施自诊断中,应答机器人也许不能保持应用关联。然而,若有可能,关联应尽量保持。因为这种服务结果而使连接切断,并不是本标准的目的。

7.4.3.1 结构

在表 9 中表示了服务原语分量的结构。

表 9 VMD 复位服务

参数名称	请求	指示	应答	确认
自变量	M	M(=)		
已扩充的推导	M	M(=)		
结果(+)			S	S(=)
状态应答			M	M(=)
结果(-)			S	S(=)
错误类型			M	M(=)

7.4.3.1.1 自变量

此参数包含 VMD 复位服务请求的参数。

7.4.3.1.1.1 已扩充的推导

此参数是布尔型,它指示是(真)否(假)作为 VMD 复位服务的一部分实施自诊断。

7.4.3.1.2 结果(+)

此参数指示所请求的服务成功。该服务的结果(+)参数包含用于 MMS 状态服务中的状态应答参数。在此应答中包含 VMD 机器人专用属性。

7.4.3.1.3 结果(-)

结果(-)参数指示服务请求失败。在 GB/T 16720.1—1996 的第 17 章中详细地定义了该错误类型参数,给出了失败的原因。

7.4.3.2 服务步骤

一旦收到 VMD 复位服务指示,应答机器人将实施下列步骤:

- 若请求该服务的 MMS 用户未持有 R_CTRL 信志,则返回结果(-)。
- 若机器人处于 ROBOT—EXECUTING 状态,则返回结果(-)。
- 若局部条件妨碍此服务的完成,则返回结果(-)。
- 删除所有 MMS 可删除的程序调用和域。
- 将机器人置于 ROBOT—IDLE 状态。
- 使用已扩充的导出参数以完成状态服务步骤(见 GB/T 16720.2—1996 的 9.2.2)。

若任一步骤失败,则服务失败,并应返回结果(-)的应答。其他状况时,则应返回结果(+),所含有的状态应答是作为在 7.3.2.1.1 中所描述的机器人状态细目来增加的。

7.4.3.3 VMD 复位协议

附加服务请求和附加服务应答的 VMD 复位选择的抽象语法是分别由 VMD 复位请求和 VMD 复位应答所规定。下面规定了这两种类型。GB/T 16720.2 中描述了在本条款中没有提供明确表明的其他参数的导出。

VMD Reset—Request ::= BOOLEAN—Extended Derivation

VMD Reset—Response ::= Status—Response

7.4.4 选择服务

选择服务提供了一种方法以标识一个程序调用作为控制机器人 VMD 的被选程序调用。它亦将施控程序调用与一组受控程序调用相结合。选择服务亦允许选弃施控程序调用以及从受控程序调用引用表中去除所有受控程序调用。

7.4.4.1 结构

在表 10 中表示了服务原语分量的结构。

表 10 选择服务

参数名称	请求	指示	应答	确认
自变量	M	M(=)		
施控程序调用	U	U(=)		
程序调用表	C	C(=)		
结果(+)			S	S(=)
结果(-)			S	S(=)
错误类型			M	M(=)

7.4.4.1.1 自变量

此参数包含了选择服务请求的各项参数。

7.4.4.1.1.1 施控程序调用

这是标识符型参数,它指示要选用的施控程序调用。此程序调用的控制属性值应等于 CONTROL-
LING。若此参数不存在,它表示现存施控程序调用已被选弃,而且无程序调用可被选用。

7.4.4.1.1.2 程序调用表

此参数是标识符表类型,它指示将被置于所引用的施控程序调用控制下的程序调用。此参数只当施
控程序调用参数存在时才存在。

7.4.4.1.2 结果(+)

此参数指示服务请求成功,成功的结果不提供服务特定参数。

7.4.4.1.3 结果(-)

此参数指示服务请求失败。在 GB/T 16720.1—1996 第 17 章中详细定义的错误类型参数给出了失
败的原因。

7.4.4.2 服务步骤

一旦收到选择服务请求,MMS 应答户应实施下列各步:

- a) 若请求此服务的 MMS 用户并不持有 R_CTRL 信志,则返回一个结果(-)。
- b) 若施控程序调用参数存在,且由施控程序调用参数指示的程序调用的控制属性不等于 CON-
TROLLING,则返回结果(-)并跳过本步骤的其余部分。
- c) 若施控程序调用参数存在,且由施控程序调用参数指示的程序调用不处于 IDLE 状态,则返回
结果(-),并跳过本步骤的其余部分。
- d) 对受控程序调用引用表的每个元素,检验程序调用控制属性是否等于 CONTROLLED,以及施
控程序调用的引用属性值是否是等于 UNCONTROLLED 或等于 VMD 被选程序调用属性的引用。若
表中的任何元素与此测试不符,则返回结果(-)并跳过本步骤的其余部分。
- e) 若 VMD 被选程序调用的引用属性不是 NONE,则检验程序调用是否处于 IDLE 状态。若不是,
则返回结果(-)并跳过本步骤的其余部分。
- f) 若 VMD 被选程序调用的引用属性不是 NONE,对被此属性引用的程序调用,实施下列各步:
 - 1) 对在施控程序调用中的受控程序调用引用表上的各个受控程序调用,设施控程序调用的引
用为 UNCONTROLLED。
 - 2) 设受控程序调用的引用表为空表。
- g) 设 VMD 的被选程序调用的引用属性为 NONE。
- h) 若施控程序调用参数不存在,则返回结果(+)并跳过本步骤的其余部分。
- i) 若由施控程序调用参数指示的程序调用的受控程序调用的引用属性表不是空表,则返回结果
(-),并跳过本步骤的其余部分。
- j) 设 VMD 的被选程序调用的引用属性等于由施控程序调用参数指示的程序调用的引用。
- k) 若程序调用参数表为空表,则设定受控程序调用的引用属性表为空表。

l) 若程序调用参数表不是空表,则对表中每个元素,设定施控程序调用的引用属性去引用被施控程序调用参数指示的程序调用。并将此程序调用增添到施控程序调用的受控程序调用的引用属性表上。

m) 返回结果(+).

7.4.4.3 选择协议

选取附加服务请求和附加服务应答的选择的抽象语法分别地由选择请求和选择应答规定。下面规定了这两种类型。GB/T 16720.2 中描述了所有在本条款中没有提供明确表明的其他参数的导出。

```
Select—Request ::= SEQUENCE{
    controlling      [0]IMPLICIT Identifier OPTIONAL,
    controlled        [1]IMPLICIT SEQUENCE OF Identifier OPTIONAL
    —this field shall appear if and only if the controlling field is included
}
```

```
Select—Response ::= NULL
```

7.4.5 变更程序调用属性服务

变更程序调用属性服务提供了一种变更一个程序调用的某个属性值的方法。

7.4.5.1 结构

在表 11 中表示了服务原语分量的结构。

表 11 变更程序调用属性服务

参数名称	请求	指示	应答	确认
自变量	M	M(=)		
施控程序调用	M	M(=)		
运行方式	C	C(=)		
无限定	S	S(=)		
循环计数	S	S(=)		
步进计数	S	S(=)		
结果(+)			S	S(=)
结果(-)			S	S(=)
错误类型			M	M(=)

7.4.5.1.1 自变量

此参数包含了变更程序调用属性服务请求的各项参数。

7.4.5.1.1.1 施控程序调用

这是标识符型参数,它标识其属性被变更的程序调用。该程序调用的控制属性应等于 CONTROLLING。

7.4.5.1.1.2 运行方式

此参数值应指示程序调用的运行方式属性值。按照被选取的程序调用运行方式。应存在下列参数之一。

7.4.5.1.1.3 无限定

这是空型参数,若运行方式属性值被设置为 FREE—RUN,则应选择此参数。

7.4.5.1.1.4 循环计数

这是整数型参数,若运行方式属性值被设置为 CYCLE—LIMITED,则应选择此参数,并将保持循环计数属性值设置为此参数值。

7.4.5.1.1.5 步进计数

这是整数型参数,若运行方式属性值被设置为 STEP—LIMITED,则应选择此参数,并将保持步进计数属性值设置为此参数值。在 PICS 中(见 9.4)应定义 STEP—LIMITED 运行方式的执行程序。若此参数没有得到支持,则不可选取该选择。

7.4.5.2 服务步骤

下列服务步骤应用于变更程序调用属性服务。

- a) 若请求的 MMS 用户并不持有 R_CTRL 信志,则返回一个结果(—)。
- b) 若程序调用的控制属性不等于 CONTROLLING,则返回结果(—)并跳过本步骤的其余部分。
- c) 施控程序调用属性应设置如下:
 - 1) 若在服务请求中选取了“无限定”参数,则程序调用的运行方式属性应被设置为 FREE—RUN。

2) 若在服务请求中选取了“循环计数”参数,则程序调用的运行方式属性应被设置为 CYCLE—LIMITED。并将程序调用的保持循环计数属性设置为循环计数参数的值。

3) 若在服务请求中选取了“步进计数”参数,则程序调用的运行方式属性应被设置为 STEP—LIMITED,并将程序调用的保持步进计数属性设置为步进计数参数的值。

7.4.5.3 变更程序调用属性协议

下面规定了变更程序调用属性请求和变更程序调用属性应答的抽象语法。GB/T 16720.2 中描述了所有在本条款中没有提供其明确表明其他参数的导出。

```
AlterProgramInvocationAttributes—Request ::= SEQUENCE {
    programInvocation [0] IMPLICIT Identifier,
    startCount [1] IMPLICIT StartCount DEFAULT cycleCount 1 }
AlterProgramInvocationAttributes—Response ::= NULL
```

7.4.5.3.1 运行方式

运行方式参数值应由在开始计数类型中所作的选择推断。“无限定”选择表示运行方式被设置为 FREE—RUN,并选取“无限定”参数。循环计数选择表示运行方式被设置为 CYCLE—LIMITED,并选取“循环计数”参数。步进计数选择表示运行方式设置为 STEP—LIMITED,并选取步进计数参数。

开始计数的抽象语法应从在运行方式参数值中所作选择来推断,如

- a) 若选取无限定,则开始计数值应是“无限定”选择;
- b) 若选取循环计数,则开始计数值应是“循环计数”选择;
- c) 若选取步进计数,则开始计数值应是“步进计数”选择。

7.5 初始化服务和协议

7.5.1 初始请求细目参数

GB/T 16720.2 给出了应由伴同标准来定义的初始请求细目参数。在表 12 中表示了初始请求细目参数的结构。

表 12 初始请求细目

参数名称	请求	指示
推荐版本号	M	M(=)
推荐参数 CBB	M	M(=)
支持服务呼叫	M	M(=)
支持附加服务呼叫	M	M(=)

对通信实例若建议采用本标准中定义的抽象语法,则本参数存在。否则,应不存在。它应包含从本标准中所定义的抽象语法导出的与表示上下文中通信有关的参数。参数分量规定如下:

7.5.1.1 推荐版本号

此参数是整数型,它应包含一个表示本标准版本小号的号。该号是用在表示上下文中所推荐的版本小号,该表示上下文是为了此通信的实例而由在本标准定义的抽象语法中导出的。一个大于“一”的编号表明支持由“一”到所推荐的编号间的所有版本小号。

本标准的大号的修订是通过性质不同的抽象语法(见 GB/T 16720.2—1996 第 17 章)的登记和定

义来反映的。小号的修订被反映在版本小号的参数中。在同一个大号的修订级别上,本标准的版本小号是与本标准的比它号小的小号版本相兼容的。

若此参数值不能支持所请求的值,则可以通过 MMS 提供者来减小该参数值。在指示原语中的值应小于或等于在请求原语中的值,但不得小于“一”。

7.5.1.2 推荐参数 CBB

在 GB/T 16720.1—1996 的 8.2.3.2 中规定了此参数。

7.5.1.3 支持服务呼叫

在 GB/T 16720.1—1996 的 8.2.3.3 中规定了此参数。

7.5.1.4 支持附加服务呼叫

这是位串型参数,它规定了一组被 MMS 呼叫户所支持的附加服务,该附加服务是本标准为在表示上下文中使用而定义的,而表示上下文是从本标准在此应用关联上定义的抽象语法所导出的。

在指示原语中的参数值规定了由 MMS 呼叫户支持的附加服务集与由 MMS 提供者支持的服务集的交集。

在 7.5.3 中规定了对位串型的单个位的服务赋值。在赋值位中“一”值表示支持对应的服务。“零”值表示不支持。任何附加位是无用的。

对确认服务的支持应被定义为接收一个请求指示的能力和准确地执行对应答器作用所规定的服务步骤。

若一个确认服务得到支持,则除协议错误外,应不对该服务的接收发送拒绝 PDU。若确认服务不被支持,则用一个“UNRECOGNIZED SERVICE”拒绝码对服务的接收发送一个拒绝 PDU。

7.5.2 初始应答细目参数

GB/T 16720.2 提供了一个应由伴同标准定义的初始应答细目参数。在表 13 中表示了初始应答细目参数的结构。

表 13 初始应答细目

参数名称	请求	指示
协商版本号	M	M(=)
协商参数 CBB	M	M(=)
支持服务受叫	M	M(=)
支持附加服务受叫	M	M(=)

若在本标准中所定义的抽象语法对此通信实例已经商定,则此参数应存在。否则不应存在。它应包含在表示上下文中有通信的参数,该表示上下文是由本标准所定义的抽象语法导出的。分量参数被说明如下。

7.5.2.1 协商版本号

这是整数型的参数,应包含一个表示本标准版本小号的号,此号是将被用于表示上下文中的本标准的版本小号。该表示上下文是为通信实例而由本标准所定义的抽象语法中取得。此号应小于或等于在请求原语中所推荐的版本号参数,但不得小于“一”。

本标准大号的修订是通过性质不同的抽象语法的登记和定义来反映的(见 GB/T 16720.2—1996 中 17 章)。在同一个大号的修订级别上,本标准的版本小号与本标准的比它号小的小号版本相兼容。

7.5.2.2 协商参数 CBB

在 GB/T 16720.1—1996 的 8.2.4.2 中说明了此参数。

7.5.2.3 支持服务受叫

在 GB/T 16720.1—1996 的 8.2.4.3 中说明了此参数。

7.5.2.4 支持附加服务受叫

这是位串型的参数,它规定了一组被 MMS 受叫户所支持的附加服务。此附加服务是本标准为在表

示上下文中使用而定义的。而该表示上下文则是从本标准在此应用关联上所定义的抽象语法导出的。

在指示原语中的参数值规定了被 MMS 受叫户支持的附加服务集与被 MMS 提供者支持的附加服务集的交集。

在 7.5.3 中规定了对位串型的单个位的服务赋值。在赋值位中“一”值表示支持对应的服务。“零”值表示不支持。任何附加位是无用的。

对确认服务的支持被定义为接收一个请求指示的能力和准确地执行对应答器作用所规定的服务步骤。

若一个确认服务得到支持,则除协议错误外,应不对服务的接收发送拒绝 PDU。若确认服务不被支持,则用一个“UNRECOGNIZED SERVICE”拒绝码对服务接收发送一个拒绝 PDU。

7.5.3 初始化协议

初始请求细目抽象语法和初始应答细目参数应由初始请求细目和初始应答细目类型分别地说明。下面规定了这些类型并在随后的段落中加以描述。GB/T 16720.2 中描述了所有在本条款中没有提供明确表明其他参数的导出。

```
InitRequestDetail ::= SEQUENCE{
    ProposedVersionNumber      [0]IMPLICIT Integer 16,
    ProposedParameterCBB       [1]IMPLICIT ParameterSupportOptions,
    ServicesSupportedCalling    [2]IMPLICIT ServiceSupportOptions,
    additionalSupportedCalling  [3]IMPLICIT AdditionalSupportOptions
}

InitResponseDetail ::= SEQUENCE{
    negotiatedVersionNumber     [0]IMPLICIT Integer 16,
    negotiatedParameterCBB      [1]IMPLICIT ParameterSupportOptions,
    ServicesSupportedcalled     [2]IMPLICIT ServiceSupportOptions,
    additionalSupportedcalled    [3]IMPLICIT AdditionalSupportOptions
}

AdditionalSupportOptions ::= BITSTRING{
    vMDStop                    (0),
    vMDReset                   (1),
    select                      (2),
    alterProgramInvocationAttributes (3)
}
```

7.6 模块结束

采用下列 END 语句结束此模块。

END

8 机器人专用标准化对象

8.1 总则

本章定义了机器人专用并可对各类机器人通用的标准化 MMS 可见实体名称及其他属性,标准化对象涉及几种不同的实体,包括类型、域、程序调用、有名变量、信志、事件条件和事件活动等。

本章所定义每个标准化对象,均有与其相关的标准化含义。并非被考虑的所有列出的对象都可被所有机器人装置所支持。然而,只要本章所列的对象能有一种机器人支持,则它也能由本标准定义的语义所支持。反之,如果一种功能标识为与一个执行程序中出现的标准化对象相关的语义时,则标准化名称便可以用来标识该功能。非标准化名称则希望由实施者或用户在开发应用程序过程中来定义。这些

非标准化名称的使用是本地事务。

本章中定义的所有标准化名称均带有前缀“R_”(即 R 加下划线),R 表示是本标准的一员,而下划线则表示是标准化名称。

注:本章包括了有名类型对象的定义。在执行程序中,该对象通常不可见时,也应将其包括在内,目的是:1)对使用一个以上的标准化有名变量对象的复合类型提供方便的符号;2)对本地有名变量对象的首选结构给执行程序者(实施者)提供指导。

8.2 标准化域对象

8.2.1 R_ARM

该名称用于标识与机器人臂有关的域,该域模拟了在 6.2 中叙述过的机器人手臂对象。对多臂机器人系统,每个臂分别由单独的域表示,如 R_ARM—1、R_ARM—2 等。

对象:域

关键属性:域名=“R_ARM”

属性:能力表

属性:状态

属性:MMS 可删除=FALSE

属性:可共享=TRUE(真)

属性:域内容

属性:从属对象表

属性:程序调用引用表={R_ARM}

8.2.2 R_CAL

该名称用于标识与校准程序有关的域。校准操作已在 6.3.6 中叙述过。

对象:域

关键属性:域名=“R_CAL”

属性:能力表

属性:状态

属性:MMS 可删除=FALSE

属性:可共享=FALSE

属性:域内容

属性:从属对象表

属性:程序调用引用表={R_CAL}

8.2.3 R_SAFE

该名称用于标识与设备安全有关的域。它包括但不限于紧急制动开关、光防护门、压敏垫等。

对象:域

关键属性:域名=“R_SAFE”

属性:能力表

属性:状态

属性:MMS 可删除=FALSE

属性:可共享=FALSE

属性:域内容

属性:从属对象表

属性:程序调用引用表

8.3 标准化程序调用对象

8.3.1 R_ARM

该名称用于标识操纵机器人手臂的程序调用。程序调用永远同 R_ARM 域相关联。

对象：程序调用

关键属性：程序调用名="R_ARM"

属性：状态

属性：域引用表={R_ARM}

属性：MMS 可删除=FALSE

属性：可重用=TRUE

属性：监控器 属性：执行自变量

属性：错误代码=0

属性：控制=CONTROLLED(受控)

属性：施控程序调用引用

8.3.2 R_CAL

该名称用于标识执行校准步骤的程序调用。此程序调用永远同 R_CAL 域相关连,校准程序已在 6.3.6 中叙述。

对象：程序调用

关键属性：程序调用名="R_CAL"

属性：状态

属性：域引用表={R_CAL}

属性：MMS 可删除=FALSE

属性：可重用=TRUE

属性：监控器

属性：执行自变量

属性：错误代码=0

属性：控制=NORMAL(正常)

8.4 标准化有名类型对象

8.4.1 R_PIS 空间位置

一种表达空间位置的数据值,它用三维数组浮点值表示。如 5.2.2 所述,一个位置可按 X、Y 和 Z 三个坐标来表达。三维数组数代表一个点的 X、Y 和 Z 坐标值。X、Y 和 Z 坐标轴相互垂直,且其正方向遵守右手定则。

对象：有名类型

关键属性：类型名=vmd-specific "R_PIS"

属性：MMS 可删除=FALSE

属性：类型说明=数组

{维数 3,

浮点{格式字长 32,指数字长 8}}

8.4.2 R_OIS 空间姿态

一种表示空间姿态的数据值称为 R_OIS,它用四维数组浮点值表示。四维数组的含义在 PICS(见 9.4)中叙述。

对象：有名类型

关键属性：类型名=vmd specific "R_OIS"

属性：MMS 可删除=FALSE

属性：类型说明=数组

{维数 4,

浮点{格式字长 32,指数字长 8}}

注:其目的是适应欧拉角,方向余弦和四元法。当表示空间姿态的国标制定后,此类型将变为该标准的定义。

8.4.3 R_PSE 位姿

一种表达空间位置和姿态的数据,这种类型是由 R_PIS 和 R_OIS 结合而成。

对象:有名类型

关键属性:类型名=vmd—specific“R_PSE”

属性:MMS 可删除=FALSE

属性:类型说明=结构

{分量类型 类型名 vmd—specific“R_PIS”,

分量类型 类型名 vmd—specific“R_OIS”}

一般而言,关节坐标同机器人位姿间的关系不是一一对应的。为了使这种关系不产生歧义,需要另外的信息。当前,还没有表达这种信息的标准,尽管关于这一课题的工作正在进行中。当这种标准制定后,将会提供适当的标准类型来描述这种信息。

8.4.4 R_EEF 末端执行器

一种表达末端执行器属性的数据值。此类型可用于描述作为机器人手臂部件的末端执行器,或作为系统部件的另一末端执行器的部件。

对象:有名类型

关键属性:类型名=vmd—specific“R_EEF”

属性:MMS 可删除=FALSE

属性:类型描述=结构{分量{

{分量名“标识符号数”,

分量类型 objld},

{分量名“工具描述符”,

分量类型 visible—String80},

{分量名“TOOL—MICS”,

分量类型 类型名 vmd—specific“R_PSE”},

{分量名“User—Detail”,

分量类型 visible—string512}}}

8.5 标准化有名变量对象

8.5.1 一般定义

第 5 章中制定的机器人系统模型,在下述各项间加以区别。

- 在任务程序同机器人手臂控制程序间传递的数据值称为编程值。
- 在机器人手臂中的路径规划器的输出数据值称为命令值。
- 对应于机器人关节实际位置的数据值称为实际值。

在本标准中,采用下述命名惯例:若对象代表的是编程值,变量名的形式为“R_P×××”;若对象代表的是命令值,变量名的形式为“R_C×××”;若对象代表的是实际值,变量名的形式为“R_A×××”。

若对象表示的是某个对象相对于某坐标的位姿(位置和姿态),变量名的形式为“R_T×××”,它表明该变量从一种坐标系变换成另一种坐标系。这种命名约定表明任何种类坐标系的变量可以互相变换。字母 T、U、B、M 及 W 分别代表工具坐标系、用户坐标系、机座坐标系、机械接口坐标系和绝对坐标系。因此,用 R_TMB 表示机械接口相对于机座坐标系的位置,或者等于将机座坐标系变换成机械接口坐标系。

8.5.2 通用有名变量对象

在 GB/T 16720.1—1996 的第 18 章中所列的所有通用有名变量对象,如果相关的功能被支持,则在这里也能支持。

8.5.3 VMD 专用标准化有名变量对象

8.5.3.1 R_VPWR 电源

该有名变量对象表示 VMD 的任何物理资源供电属性值。当电源加于 VMD 中任何一个物理资源时,这个布尔变量值为 TRUE(真),反之则为 FALSE(假)

对象: 有名变量

关键属性: 变量名=vmd—specific“R_VPWR”

属性: MMS 可删除=FALSE

属性: 类型说明=布尔量

属性: 存取方法

8.5.3.2 R_VCAL 校准

该有名变量对象表示 VMD 所有物理资源已校准的属性值。

对象: 有名变量

关键属性: 变量名=vmd—specific“R_VCAL”

属性: MMS 可删除=FALSE

属性: 类型说明=布尔量

属性: 存取方法

8.5.3.3 R_VLOCAL VMD 本地控制

该有名变量对象表示 VMD 的本地控制属性值。

对象: 有名变量

关键属性: 变量名=vmd—specific“R_VLOCAL”

属性: MMS 可删除=FALSE

属性: 类型说明=布尔量

属性: 存取方法

8.5.3.4 R_VSAFEV 违反安全联锁

该有名变量对象表示 VMD 的违反安全联锁属性的状态。

对象: 有名变量

关键属性: 变量名=vmd—specific“R_VSAFEV”

属性: MMS 可删除=FALSE

属性: 类型说明=布尔量

属性: 存取方法

8.5.4 特定域标准化有名变量对象

8.5.4.1 P_DPWR 装置供电

该有名变量对象表示机器人臂或其他辅助装置对象的装置供电属性值。若装置已供电,则该对象有值 TRUE。

对象: 有名变量

关键属性: 变量名=特定域{
域标识符
项标识符 “R_DPWR”}

属性: MMS 可删除=FALSE

属性: 类型说明=布尔量

属性: 存取方法

8.5.4.2 R_DCAL 装置校准

该有名变量对象表示机器人臂或其他辅助装置对象的装置校准属性值。当装置资源已经校准,对象值为0;当装置资源未被校准,对象值为1,若装置资源正在校准,对象值为2。

对象: 有名变量

关键属性: 变量名=特定域{
域标识符
项标识符 “R_DCAL”}

属性: MMS 可删除=FALSE

属性: 类型说明=整数 8

属性: 存取方法

8.5.4.3 R_DLOCAL 装置本地控制

该有名变量对象表示机器人臂或其他辅助装置对象的本地控制属性值。当装置资源处在本地控制中时,对象值为TRUE(真),反之则为FALSE(假)。

对象: 有名变量

关键属性: 变量名=特定域{
域标识符
项标识符 “R_DLOCAL”}

属性: MMS 可删除=FALSE

属性: 类型说明=布尔量

属性: 存取方法

8.5.5 R_ARM 特定域标准化有名变量对象

8.5.5.1 R_NJ 关节数

该有名变量对象表示机器人臂对象的关节属性数的值。该值为整数,它标明与特定的机器人臂相关的关节数。

对象: 有名变量

关键属性: 变量名=特定域{
域标识符 “R_ARM”,
项标识符 “R_NJ”}

属性: MMS 可删除=FALSE

属性: 类型说明=整数 8

属性: 存取方法

8.5.5.2 R_JT 关节类型

该有名变量对象表示机器人臂对象的关节类型属性值。该对象为一整数数组,标识机器人臂关节的类型。数组元素为0时,表示为回转关节,1表示棱柱形关节,数组的大小(元素数)为机器人的关节数R_NJ。与机器人上的每个关节对应的关节信息的排序是一个本地事务。

对象: 有名变量

关键属性: 变量名=特定域{
域标识符 “R_ARM”,
项标识符 “R_JT”}

属性: MMS 可删除=FALSE

属性: 类型说明=数组

{元素数 R_NJ 值,元素类型 整数 8}

属性: 存取方法

8.5.5.3 R_JCAL 关节校准

该有名变量对象是表示在机器人臂对象的关节说明有序表中每一个关节的校准属性值。该数组包含每个关节的校准属性,与机器人上每个关节对应的关节信息排序是一个本地事务。

对象: 有名变量

关键属性: 变量名=特定域{

域标识符 “R_ARM”,

项标识符 “R_JCAL”}

属性: MMS 可删除=FALSE

属性: 类型说明=数组

{元素数 r—NJ 值,元素类型 布尔量}

属性: 存取方法

8.5.5.4 R_JBK 制动器合

该有名变量对象表示机器人臂对象的关节制动属性值。该数组包括每个关节的制动状态值。数组元素为 0 表示相应的关节无制动(关节制动属性为 FALSE(假)),元素为 1 表示制动器合属性值为 FALSE(假),元素为 2 表示制动器合属性值为 TRUE(真)。与机器人上每个关节对应的关节信息的排序是一个本地事务。

对象: 有名变量

关键属性: 变量名=特定域{

域标识符 “R_ARM”,

项标识符 “R_JBK”}

属性: MMS 可删除=FALSE

属性: 类型说明=数组

{元素数 r—NJ 值,元素类型 整数 8}

属性: 存取方法

8.5.5.5 R_JBD 关节界限

该有名变量对象表示机器人臂对象的上、下限边界属性值,数组含有在关节坐标中的每个上限和下限值。与机器人上每个关节对应的关节信息排序是一个本地事务。

对象: 有名变量

关键属性: 变量名=特定域{

域标识符 “R_ARM”,

项标识符 “R_JBD”}

属性: MMS 可删除=FALSE

属性: 类型说明=数组

{元素数 r—NJ 值,元素类型 结构

{分量{

{分量名“上限”,

分量类型 浮点

{格式字长 32,指数字长 8}

}

{分量名“下限”,

分量类型 浮点

{格式字长 32 指数字长 8}

}}}

属性：存取方法

8.5.5.6 R_AJV 实际关节值

该有名变量对象表示机器人臂对象的实际关节值的属性值。该数组含有关节坐标的实际值。关节信息排序与该机器人中每个关节的排序一致，是一个本地事务。

对象：有名变量

关键属性：变量名=特定域{

域标识符 “R_ARM”

项标识符 “R_AJV”}

属性：MMS 可删除=FALSE

属性：类型说明=数组

{元素数 r_NJ 值，

元素类型 浮点

{格式字长 32,指数字长 8}}

属性：存取方法

8.5.5.7 R_TUB 用户坐标系对机座坐标系的变换

该有名变量对象表示机器人臂对象的 USER—BASE 属性值。该值是表达在机座坐标系中的用户坐标系位姿值，或等于将机座坐标系变换成用户坐标系。它是位姿类型。

对象：有名变量

关键属性：变量名=特定域{

域标识符 “R_ARM”，

项标识符 “R_TUB”}

属性：MMS 可删除=FALSE

属性：类型说明=类型名 vmd—specific“R_PSE”

属性：存取方法

8.5.5.8 R_TTM 工具坐标系对机械接口坐标系的变换

该有名变量对象表示机器人臂对象的 TOOL—MICS 属性值。该值是表达在机械接口坐标系中的工具坐标系的位姿值，或等于将机械接口坐标系变换成工具坐标系，它是位姿型。

对象：有名变量

关键属性：变量名=特定域{

域标识符 “R_ARM”，

项标识符 “R_TTM”}

属性：MMS 可删除=FALSE

属性：类型说明=类型名 vmd—specific“R_PSE”

属性：存取方法

8.5.5.9 R_TBW 机座坐标系对绝对坐标系的变换

该有名变量对象表示机器人臂对象的 BASE—WORLD 属性值，该值是表达在绝对坐标系中的机座坐标系的位姿，或等于将绝对坐标系变换成机座坐标系。它是位姿型。

对象：有名变量

关键属性：变量名=特定域{

域标识符 “R_ARM”，

项标识符 “R_TBW”}

属性：MMS 可删除=FALSE

属性：类型说明=类型名 vmd—specific“R_PSE”

属性：存取方法

8.5.5.10 R_ATUP 实际工具用户点

该有名变量对象表示在用户坐标系中工具的实际位姿值。它是由机器人臂属性 TOOL—MICS、MICS—BASE、USER—BASE、实际关节值和执行程序专用参数得到的。

对象：有名变量

关键属性：变量名=特定域{
域标识符 “R_ARM”，
项标识符 “R_ATUP”}

属性：MMS 可删除=FALSE

属性：类型说明=类型名 vmd—specific“R_PSE”

属性：存取方法

8.5.5.11 R_CTUP 指令工具用户点

该有名变量对象表示机器人臂对象指令 TOOL—USER 属性值。该对象包含了在用户坐标系中工具的目标位姿值。若没有运动正在进行，则该值是实际位姿，或是最后的已接受指令的机器人臂的位姿。

对象：有名变量

关键属性：变量名=特定域{
域标识符 “R_ARM”，
项标识符 “R_CTUP”}

属性：MMS 可删除=FALSE

属性：类型说明=类型名 vmd—specific“R_PSE”

属性：存取方法

8.5.5.12 R_AMBP 对于机座的实际机械接口位置

该有名变量对象表示了机器人臂对象的 MICS—BASE 属性值。其值表示了对于机器人机座的 MICS 的实际位置。它是由机器人臂属性实际关节值和执行程序的专用参数得到的。

对象：有名变量

关键属性：变量名=特定域{
域标识符 “R_ARM”，
项标识符 “R_AMBP”}

属性：MMS 可删除=FALSE

属性：类型说明=类型名 vmd—specific“R_PSE”

属性：存取方法

8.5.5.13 R_SF 速度系数

该有名变量对象表示了机器人臂对象的编程速度属性值。该对象包含了所编程的工具中心点速度对指令速度的比值。它是一个无量纲数，通常为 1。

对象：有名变量

关键属性：变量名=特定域{
域标识符 “R_ARM”，
项标识符 “R_SF”}

属性：MMS 可删除=FALSE

属性：类型说明=浮点

{格式字长 32，指数字长 8}

属性：存取方法

8.5.5.14 R_PSTU“工具—用户”坐标系编程速度

该有名变量对象表示了机器人臂对象的编程速度属性值,该对象包含工具速度编程值。

对象: 有名变量

关键属性: 变量名=特定域{
域标识符 “R_ARM”,
项标识符 “R_PSTU”}

属性: MMS 可删除=FALSE

属性: 类型说明=浮点
{格式字长 32,指数字长 8}

属性: 存取方法

8.5.5.15 R_EEFU 末端执行器在使用中

该有名变量对象表示了当前在使用中的末端执行器的属性值,它是机器人臂对象本身的一个属性。

对象: 有名变量

关键属性: 变量名=特定域{
域标识符 “R_ARM”,
项标识符 “R_EEFU”}

属性: MMS 可删除=FALSE

属性: 类型说明=类型名 vmd-specific“R_EEF”

属性: 存取方法

8.6 标准化信志对象

8.6.1 R_CTRL

此 VMD 特定信志对象可用于表示一个对机器人臂资源具有控制的信志。它适用于在多主机系统中要求由一个主机对所有的 VMD 的物理资源进行排它控制时。

对象: 信志

关键属性: 信志名=vmd-specific“R_CTRL”

属性: MMS 可删除=FALSE

属性: 类别=TOKEN

属性: 令牌数=1

属性: 被占令牌数

属性: 持有者清单

属性: 请求者清单

8.7 标准化事件条件对象

8.7.1 R_RVS 机器人 VMD 状态变化

此事件条件对象标识 VMD 属性机器人 VMD 状态由 EXECUTING 变化到任何其他状态的事件条件。

对象: 事件条件

关键属性: 事件条件名=vmd-specific“R_RVS”

属性: MMS 可删除=FALSE

属性: 事件条件类别=MONITORED

属性: 状态=(初始 ACTIVE)

属性: 优先权=正常优先权

属性: 严重度=正常严重度

属性: 附加细目=NULL

属性: 事件登录引用表

属性：使能=TRUE
属性：报警摘录报告=FALSE
属性：监控变量引用=UNSPECIFIED
属性：评价间隔=(本地事务)
属性：最后转移到激活的时间
属性：最后转移到空闲的时间

8.7.2 R_SIV 违反机器人安全连锁

此事件条件对象标识了当违反 VMD 属性安全连锁时由 FALSE 转移到 TRUE 时的事件条件。

对象：事件条件

关键属性：事件条件名=vmd-specific“R_SIV”

属性：MMS 可删除=FALSE

属性：事件条件类别=MONITORED

属性：状态=ACTIVE

属性：优先权=正常优先权

属性：严重度=正常严重度

属性：附加细目=NULL

属性：事件登录引用表

属性：使能=TRUE

属性：报警摘录报告=FALSE

属性：监控变量引用=UNSPECIFIED

属性：评价间隔=(本地事务)

属性：最后转移到激活的时间

属性：最后转移到空闲的时间

8.7.3 R_RIC 机器人本地控制改变

此事件条件对象标识了当 VMD 属性本地控制值改变时的事件条件。

对象：事件条件

关键属性：事件条件名=vmd-specific“R_RLC”

属性：MMS 可删除=FALSE

属性：事件条件类别=MONITORED

属性：状态=ACTIVE

属性：优先权=正常优先权

属性：严重度=正常严重度

属性：附加细目=NULL

属性：事件登录引用表

属性：使能=TRUE

属性：报警摘录报告=FALSE

属性：监控变量引用=UNSPECIFIED

属性：评价间隔=(本地事务)

属性：最后转移到激活的时间

属性：最后转移到空闲的时间

8.7.4 R_ARM 机器人臂操作

此事件条件对象标识了当 R_ARM 程序调用脱离 RUNNING 状态时的事件条件。

对象：事件条件

关键属性：事件条件名=vmd—specific“R_ARM”

属性：MMS 可删除=FALSE

属性：事件条件类别=MONITORED

属性：状态=ACTIVE

属性：优先权=正常优先权

属性：严重度=正常严重度

属性：附加细目=NULL

属性：事件登录引用表

属性：使能=TRUE

属性：报警摘录报告=FALSE

属性：监控变量引用=UNSPECIFIED

属性：评价间隔=(本地事务)

属性：最后转移到激活的时间

属性：最后转移到空闲的时间

8.8 标准化事件活动对象

8.8.1 R_STC 机器人状态变化

此事件活动对象标识了当出现一个有名事件时所采取的活动。该活动将提供 MMS 状态服务结果的事件通告。

对象：事件活动

关键属性：事件活动名=vmd—specific“R_STC”

属性：MMS 可删除=FALSE

属性：确认服务请求=

状态{扩充导出=FALSE}

属性：修饰符表=NULL

属性：事件登录引用表

8.8.2 R_ARM 机器人臂操作

此事件活动对象标识了当 R_ARM 程序调用脱离 RUNNING 状态时所采取的活动。该活动将给出 MMS 获取程序调用属性服务结果的事件通告。

对象：事件活动

关键属性：事件活动名=vmd—specific“R_ARM”

属性：MMS 可删除=FALSE

属性：确认服务请求=

获取程序调用属性{“R_ARM”}

属性：修饰符表=NULL

属性：事件登录引用表

8.9 标准化事件登录对象

本标准未定义任何机器人专用的标准化事件登录对象。

8.10 标准化操作站对象

本标准未定义任何机器人专用的标准化操作站对象。

8.11 标准化日志对象

本标准未定义任何机器人专用的标准化日志对象。

9 机器人一致性类别

9.1 总则

本章列出机器人应用的 MMS 一致性类别集。一致性类别是以机器人的应用模型为基础。应用表是以类别名称与程序加载方法类型相结合来表示的。上装/下装是针对能直接支持其装置的远程上装和下装系统的。二次存储是针对仅能从其本地文件库加载的系统。所有各种类别均包含了基本一致性类别。GB/T 16720.1 描述了以服务器或 VMD 的行为为基础的一致性。

机器人的柔性应用要求机器人与网络相连,且要求其活动与其他设备协调。机器人有两种基本操作方式,即作为客户和作为服务器。机器人既可起其中任何一种装置的作用,也可同时起两种装置的作用。起客户作用时和作服务器应答时机器人的服务和一致性要求应分别考虑。

本章描述了当机器人起服务器作用时,最低限度的服务和一致性要求。同时,也提出了机器人作客户时和作服务器时的增强性服务和一致性要求。

9.2 机器人服务一致性

9.2.1 一般要求

所有 MMS 应答服务器的最低限度要求,已在 GB/T 16720.1—1996 的 17.9 中描述过。支持应答器作用的最低限度要求是:

- 起动;
- 终止;
- 异常终止;
- 拒绝;
- 标识。

这些要求与对本标准中定义的抽象语法的任何执行程序的一致性要求具有同等效力。

在下述一致性类别中,机器人是作为 MMS 服务器,同作为 MMS 客户起作用的远程主计算机相连。

9.2.2 基本一致性类别

9.2.2.1 静态一致性要求

基本一致性类别规定了机器人应保证与本标准一致的最小服务集。并成为包含在所有机器人的其他一致性类别中的服务核心。

此一致性类别适用于每次仅支持一个关联的机器人服务器。支持一个以上关联的机器人服务器,在高级一致性类别(即多主机类别)中加以考虑。这些高级一致性类别具有外加的要求。

为满足第 5 章中机器人模型、第 6 章中 MMS 模型映射以及第 7 章中服务步骤元素的要求,提出或接受建立一种规定为本标准中定义的抽象语法的应用关联的远程用户,一旦关联建立成功,将立即获得对 R_CTRL 信志的控制。信志不需要明确的 MMS 服务步骤的支持以满足此一致性类别。R_CTRL 信志对远程 MMS 用户的自动定位,不适用于可支持多关联中某种高级一致性类别。

9.2.2.2 服务一致性要求

9.2.2.2.1 固定要求

基本一致性类别支持下述各种服务:

- 取消;
- 状态;
- 非请求状态;
- 取域属性;
- 删除域;
- 建立程序调用;

删除程序调用；
开始；
停止；
恢复；
复位；
获取程序调用属性；
读；
写；
信息报告；
选择。

除以上服务外，机器人应支持在 9.2.2.2.2.1 中或在 9.2.2.2.2.2 中规定的服务集。

9.2.2.2.2 域装载

9.2.2.2.2.1 上装/下装

这组服务是被从主机系统向机器人系统控制器进行远程装载的机器人所支持的服务集。服务集提供了向/从机器人系统控制器上装及下装程序和数据的方法。

初始下装序列；
下装段；
终止下装序列；
初始上装序列；
上装段；
终止上装序列；
请求域下装；
请求域上装。

9.2.2.2.2.2 二次存储

这组服务是由不能用机器人系统控制器的执行存储器直接下装和上装的机器人所支持的服务集。

通常为执行此 MMS 服务的服务步骤，需要某些文件传送技术的支持。可以采用 FTAM (ISO 8571) 和 GB/T 16720.1—1996 的附录 C 中的文件服务或其他合适的技术，来实现此类文件传送。本标准不规定要支持哪一种技术。

为使此服务集允许从或向发出请求的主机装载和存储域，TPY 参数应被支持。若该参数值为首发请求者名称，则仅在此一致性类别中才能支持此参数。

装载域内容；
存储内容。

9.2.2.3 参数一致性构造块

基本一致性类别支持下述参数一致性构造块及其值：

STR1
STR2
VNAME
NEST=2

9.2.2.4 动态一致性要求

机器人不应由于局部应急制动，或由除异常终止和终止以外的任何 MMS 请求服务而异常终止或终止一个关联。通信关联的建立或解除，不应造成机器人执行状态的改变。特别是当通信关联被终止或异常终止后，机器人不应中断运行。

9.2.3 高级一致性类别

9.2.3.1 通用要求

本条款定义了机器人可遵循的高级一致性类别。所有高级一致性类别包含基本一致性类别所提供的服务。每种高级一致性类别均可彼此无关地得到支持。

9.2.3.2 多主机类别

机器人应提供的符合此类别的服务和一致性要求有待进一步研究。附录 B 提供了考虑这种一致性类别的初步指导。

9.2.3.3 增强类别

机器人应提供的符合此类别的服务和一致性要求有待进一步研究。附录 B 提供了考虑这种一致性类别的初步指导。

9.2.3.4 事件类别

机器人应提供的符合此类别的服务和一致性要求有待进一步研究。附录 B 提供了考虑这种一致性类别的初步指导。

9.2.3.5 动态定义类别

机器人应提供的符合该类别的服务和一致性要求有待进一步研究。附录 B 提供了考虑这种一致性类别的初步指导。

9.3 机器人客户一致性

9.3.1 通用要求

在此一致性类别中,机器人系统控制器对附属装置提供“主机”服务。但在本标准中,不可能把所有可能的装置都考虑进来。下述几种是比较主要的:

- 视觉及其他传感器;
- 夹持器及其他操作机;
- 轴延伸器,如 X—Y 定位器;
- AGV 及其他机器人。

除了一台机器人对另一台起服务器作用的机器人作为客户身份的情况外,本标准并未就附属装置的一致性加以说明。但描述了关于机器人作为“主机”身份的一致性要求。对这种机器人系统控制器的 MMS 要求,是对那些由附加服务器装置支持的 MMS 服务的补充。

非 MMS 依从装置在本标准中不作考虑。

9.3.2 监控类别

机器人应提供的符合此类别的服务和一致性要求有待进一步研究。附录 B 提供了考虑这种一致性类别的初步指导。

9.3.3 控制类别

机器人应提供的符合此类别的服务和一致性要求有待进一步研究。附录 B 提供了考虑这种一致性类别的初步指导。

9.4 PICS

9.4.1 总则

本条款描述了协议执行程序一致性语句形式(PICS)。每个编程者应执行全部 PICS。PICS 共有六部分,在 GB/T 16720.1—1996 第 18 章中描述了第 1、2、3 和 4 部分。第 5 部分描述补充了附加服务 CBB。第 6 部分描述机器人专用功能的本地执行程序值。

9.4.2 PICS 第 5 部分:附加服务 CBB

编程者应填写表 14 的各项,该表表示当运行在由本标准定义的抽象句法中时 MMS 的执行程序是否满足作为服务器时的要求,作为客户时的要求,或同时满足这两者的要求。上述三种要求分别以“服务器”、“客户”或“兼有”来表示。若 MMS 执行程序未能满足对服务器或 CBB 的服务器或客户的要求,则该栏便为空白。

表 14 PICS 附加服务 CBB

PICS 序号

发布日期

附加服务一致性构造块	服务器、客户或兼有
VMD 复位 VMD 停止 选择 变更程序调用属性	

9.4.3 PICS 第 6 部分:本地执行程序值

编程者应填写表 15 的执行程序值。对需要解释的项目,编程者应在表中填入由编程者提供的参考附页或段落。

表 15 PICS 本地实现值

PICS 序号

发布日期

PICS 本地执行程序值	
支持 STEP _ LIMITED(6.3.2.6) 支持装置表(6.3.4.1) 第一步含义(7.3.3.3.3) R _ OIS 的解释(8.4.2) 程序位置解释(6.3.2.5)	

附录 A

(提示的附录)

示 例

A1 背景

考虑一个生产操作示例如汽车仪表盘的组装。该操作的主要任务包括固定仪表盘的骨架,将分层电子电路板紧固于骨架上,在盘上安装一组灯泡,并将一个罩固定在组件上。为了以后在汽车仪表面板上的安装,完成这些操作后,把部件包装起来。

A2 生产方案

仪表盘的组装操作是由一个机器人系统来执行的。机器人系统是由一台机器人,一个简单的固定装置和一个安全连锁装置组成。仪表组装操作是由一个单元控制器协调的一个更大操作步骤的一部分。

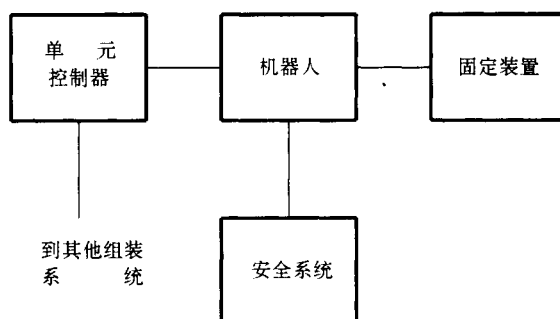


图 A1 示例的方案

A3 操作

A3.1 单元控制器的任务

单元控制器决定着在仪表盘组装单元中的协调监控和机器人系统的控制。在此例中,单元控制器的作用是一个 MMS 客户,而机器人是作为 MMS 的服务器。

单元控制器有下列主要的响应特性:

- a) 仪表盘组装操作的初始化;
- b) 启动组装操作顺序;
- c) 从机器人系统中抽取关键质量控制参数;
- d) 应答由机器人系统生成的报警条件;
- e) 完成批量后平稳地关闭操作。

A3.2 机器人任务

机器人系统实施下列任务:

- a) 取一个仪表盘骨架并把它放入固定装置中;
- b) 经由分离的 I/O,指令固定装置卡紧;
- c) 把分层电路板放在骨架上;
- d) 把一组灯泡配置在表盘骨架上;
- e) 把灯泡罩子配置在灯泡上;
- f) 指令固定装置松开,移开已完成的组件,通知完成组装的单元控制器,继续进行另一件组装。

A4 机器人 VMD 描述

A4.1 域和有名变量对象

机器人的 VMD 包括下列域和有名变量对象：

- a) R_ARM——此有名域是与机器人臂有关，并包含了几个表示机器人臂物理资源的有名变量对象；
- b) R_SAFE——此有名域是与机器人相关设备的安全性有关，并且包含了几个表示此机器人设备安全性的有名变量对象；
- c) R_CAL——此有名域是与机器人校准步骤有关，它包含了执行校准步骤所需的执行软件；
- d) Assemble_Cluster——此有名域包含了完成仪表盘组装操作的执行软件。

A4.2 程序调用对象

机器人的 VMD 包含了下列程序调用对象：

- a) R_ARM——此标准化的程序调用是用于控制机器人臂运动；
- b) R_CAL——此标准化程序调用是用于实施校准步骤；
- c) Assemble_Cluster——此程序调用是用于实施仪表盘装配操作。

A4.3 事件条件对象

机器人的 VMD 包含了下列事件条件对象：

- a) R_ARM——此标准化事件条件对象是用于当 R_ARM 程序调用脱离 RUNNING 状态时，命名被触发事件。

A4.4 事件活动对象

机器人的 VMD 包含了下列事件活动对象：

- a) R_ARM——此标准化的事件活动对象用于命名一个获取程序调用属性服务的活动。

A5 使用 MMS 服务的操作

A5.1 组装单元初始化

单元控制器，通过首先建立一个与机器人控制器相关的通信，初始化仪表盘的组装过程。所用的 MMS 服务是初始化。单元控制器发送：

```
initiate—RequestPDU{
    ProposedMaxServOutstanding Calling 1,
    ProposedMaxServOutstandingCalled 1,
    init Request Detail{
        ProposedVersionNumber 1,
        ProposedParameterCBB{str1,vnam},
        ServicesSupportedCalling{identify,eventNotification},
        additionalSupportedCalling{    }
    }}
}
```

机器人应答：

```
initiate—ResponsePDU{
    negotiatedMaxServOutstandingCalling 1,
    negotiatedMaxServOutstandingCalled 1,
    initResponseDetail{
        negotiatedVersionNumber 1,
        negotiatedParameterCBB{str1,vnam},
    }}
```

```

servicesSupportedCalled{
    identify,
    status,
    read,
    write,
    conclude,
    defineEventEnrollment,
    start,
    stop,
    resume,
    reset},
additionalSupportedCalled{
    vMDStop,
    vMDReset,
    select}
}

```

单元控制器建立一个事件登录,因此在任何程序调用完成时,它将接到通告。单元控制器发送:

```

confirmed—RequestPDU{
    invokeID1,
    defineEventEnrollment{
        eventEnrollmentName vmd—specific“Assemble _ Cluster”,
        eventConditionName vmd—specific“R _ ARM”,
        eventConditionTransitions{active—to—idle},
        alarmAcknowledgementRule none,
        eventActionName vmd—specific“R _ ARM”
    }
}

```

机器人应答:

```

confirmed—Response PDU{
    invoke ID 1,
    defineEventEnrollment NULL
}

```

单元控制器校准机器人以保证其坐标系统的参照是正确的。用于校准机器人的 MMS 服务是“开始”。对标准化程序调用 R _ CAL 发出“开始”指令。单元控制器发送:

```

confirmed—RequestPDU{
    invokeID 2,
    start{
        programInvocationName “R _ CAL”
    }
}

```

实施校准功能时,R _ CAL 程序转到执行。机器人应答:

```

confirmed—Response PDU{
    invokeID 2,
    start NULL
}

```

一旦完成校准,单元控制器下装在机器人系统中实施仪表盘组装操作所需的适配域。所使用的

MMS 服务是对 Assemble _ Cluster 域的初始下装序列、下装段和终止下装序列。单元控制器发送：

```
confirmed—Request PDU{
    invokeID 3,
    initiateDownloadSequence{
        domainName "Assemble _ Cluster",
        listOfCapabilities{ },
        sharable FALSE
    }}
}
```

机器人应答：

```
confirmed—Response PDU{
    invokeID 3,
    initiateDownloadSequence NULL
}
```

此刻，下装处理已被初始化了，机器人继续请求下装段，直到下装处理完成为止。对于本例，假定首次记录包含了正文“BEGIN”和最后记录包含了正文“END”。当然，一个真实下装将包含机器人控制信息的二进制记录。机器人发送：

```
confirmed—Request PDU{
    invokeID 1,
    downloadsegment "Assemble _ Cluster"
}
```

单元控制器应答：

```
confirmed—Response PDU{
    invokeID 1,
    downloadsegment{
        loadData non—coded '424547494E'H,
        more Follows TRUE
    }}
}
```

处理完几个这种记录，最终达到最后记录。机器人发送：

```
confirmed—Request PDU{
    invokeID 2,
    downloadsegment "Assemble _ Cluster"
}
```

单元控制器应答：

```
confirmed—Response PDU{
    invokeID 2,
    downloadsegment{
        loadData non—coded '454E44'H,
        moreFollows FALSE
    }}
}
```

至此已经处理全部下装记录，机器人终止下装处理。机器人发送：

```
confirmed—Request PDU{
    invokeID 3,
    terminateDownloadsequence{

```

domainName “Assemble _ cluster”

单元控制器应答:

```
confirmed—Response PDU{
    invokeID 3,
    terminateDownloadSequence NULL
}
```

接着,单元控制器用刚下装的域建立机器人控制程序。单元控制器发送:

```
confirmed—Request PDU{
    invokeID 4,
    CreateProgramInvocation{
        ProgramInvocationName“Assemble _ Cluster”,
        ListOfDomainNames{“Assemble _ Cluster”},
        monitorType TRUE
    },
    CreateProgramInvocation controlling
}
```

机器人应答:

```
confirmed—Response PDU{
    invokeID 4,
    createProgramInvocation NULL
}
```

选择命令被用来把该程序调用与机器人臂的操作结合起来。单元控制器发送:

```
confirmed—Request PDU{
    invokeID 5,
    additionalService select{
        Controlling“Assemble _ Cluster”,
        Controlled{“R _ ARM”}
    }
}
```

机器人发送:

```
confirmed—Response PDU{
    InvokeID 5,
    additionalService select NULL
}
```

A5.2 操作

单元控制器指令机器人开始装配操作。所用的 MMS 服务是“开始”。“开始”指令以运行方式属性设置为“无限”循环操作方式被发送到 Assemble _ Cluster 程序调用。机器人接收 Assemble _ Cluster 程序调用的开始指令,并且,如果条件许可,将开始 Assemble _ Cluster 和 R _ ARM 这两种程序调用。

单元控制器发送:

```
Confirmed—Request PDU{
    invokeID 6
    start{
        programInvocationName“Assemble _ Cluster”
    },
}
```

```

Start{
    controlling startCount noLimit NULL
}

```

机器人应答:

```

Confirmed—Response PDU{
    invokeID 6,
    start NULL
}

```

机器人进而实施仪表盘各步的装配。单元控制器通过 MMS 状态服务监控机器人的状态。单元控制器发送:

```

Confirmed—Request PDU{
    invokeID 7,
    status FALSE
}

```

机器人应答:

```

Confirmed—Response PDU{
    invokeID 7,
    status{
        vmdLogicalStatus state—changes—allowed,
        vmdPhysicalStatus Operational
    },
    status{
        robotVMDState robot—executing,
        robotSpecificStatus{
            anyPhysicalResourcesPowerOn,
            allPhysicalResourcesCalibrated
        },
        selectedProgramInvocation programInvocation“Assemble _ Cluster”
    }
}

```

在操作过程中,单元控制器可询问有关处理的状态。例如:它可对已预定义的 MMS 变量“Part _ Count”,请求获得已生产的零件数的信息。

```

Confirmed—Request PDU{
    invokeID 8,
    read{
        variableAccessSpecification listOfVariable
        {{variableSpecification name vmd—specific“Part _ Count”}}}}

```

机器人应答:

```

Confirmed—Response PDU{
    invokeID 8,
    read{
        listOfAccessResult{success integer 25}
    }
}

```

A5.3 异常

若机器人停止操作,即机器人程序调用脱离运行状态。将生成一个事件通告并发送到单元控制器。
机器人发送:

```
unconfirmed—PDU {eventNotification {
    eventEnrollmentName vmd—specific “Assemble _ Cluster”,
    eventConditionName vmd—specific “R _ ARM”,
    Severity 64,
    transitionTime undefined NULL,
    actionResult {
    eventActionName vmd—specific “R _ ARM”,
    eventActionResult success {
        getProgramInvocationAttributes {
            state stopped,
            listOfDomainNames { “R _ ARM” },
            mmsDeletable TRUE,
            reusable TRUE,
            monitor TRUE,
            executionArgument simpleString “ ” },
        getProgramInvocationAttributes {
            errorCode 0,
            control Controlling {
                controlled { “R _ ARM” },
                runningMode freeRunning NULL
            } } } } }
```

此刻,单元控制器能够进一步采取诊断活动。首先它发送一个状态请求。单元控制器发送:

```
Confirmed—Request PDU {
    invokeID 9,
    Status FALSE
}
```

机器人应答:

```
Confirmed—Response PDU {
    invokeID 9,
    Status {
        vmdLogicalStatus no—state—changes—allowed,
        vmdPhysicalStatus needs—commissioning
    },
    status {
        robotVMDState robot—motion—paused,
        robotSpecificStatus {
            safetyInterlocksViolated,
            anyPhysicalResourcePowerOn,
            allPhysicalResourcesCalibrated,
            localControl
        },
    }
```

```
selectedProgramInvocation noneSelected NULL
}}
```

现在,因知安全连锁已被扰乱,须选择停止整个机器人。它以发出“VMD 停止”服务请求来实现。单元控制器发送:

```
confirmed—Request PDU{
  invokeID 10,
  additionalService vMD Stop NULL
}
```

机器人应答:

```
Confirmed—Response PDU{
  invokeID 10,
  additionalService vMDStop NULL
}
```

A5.4 关闭

当单元控制器准备终止仪表盘组装操作和关闭单元时,它发送终止请求。单元控制器发送:

```
Conclude—RequestPDU NULL
```

机器人应答:

```
conclude—ResponsePDU NULL
```

附 录 B

(提示的附录)

进一步问题——高级一致性类别

B1 机器人服务器一致性类别

B1.1 总则

提供本附录是为了展示为定义更高级的一致性类别正研究的方向。进一步工作是要求完成这些类别的定义和标识一致性要求。一旦确定,这些类别的描述将被列入标准的正文。

B1.2 多主机类别

B1.2.1 通用要求

在 5.1.2 中叙述的(配置一)要求是最低限的考虑。这对诸如在配置二中所表示的联接将是不够的,因为最低限系统一次仅支持一个联接。为了支持与多主机的第三方文件的传送和其他功能,要求增强一致性。

B1.2.2 多关联

当第二主机获有控制权时,要求有多个关联,以便于第三方文件的传送和允许由第一主机监控机器人。

B1.2.3 申请引用(TPY)

当多主机时,要求有一个机制,以规定一个源系统名,以便第三方文件传送,以及为了信志协调和发出变更系统事件信号。网络需要了解哪个系统是数据的来源或目的地。

B1.2.4 获取文件

当多主机时,机器人需要有从网络中的任何主机获取文件的能力。

B1.2.5 取得和放弃控制

当网络中为多主机时,需要有一种方式保证一次只有一个主机能够承担机器人控制。

B1.3 增强类型**B1.3.1 操作器 I/O**

此种类别将支持操作器 I/O 作为一个服务器。

B1.3.2 信志修饰符附加

在更复杂的系统中,要求要比单一信志更多,在单一操作中,可能需要取得及放弃对一个信志的控制权。

B1.3.3 校准

当主机发出请求时,此类别将提供一个服务去校准机器人。支持此服务的要求将通过支持 R_CAL 预定名称的要求指明。

B1.4 事件类别

此类别将支持能使机器人发出事件信号的服务。此类别将限定支持在机器人中预定义事件的要求。

B1.5 动态定义类别

此服务类别使变量名、变量类型、事件和信志能通过网络进行说明和修改。

B2 机器人客户一致性**B2.1 通用要求**

对于在使用 MMS 的网络中充当客户的机器人已经定义下述要求的类别。提供这些要求类别是为编程者和其他伴同标准制定者进行指导。机器人客户一致性要求应由将与机器人进行通信的装置的伴同标准来确定。条款 B2.2 和 B2.3 为伴同标准制定者标识了机器人客户最常使用的服务。

在澄清这些客户一致性类别的定义中,它需要进行补充工作。但一旦完成,这些说明并不移入标准正文。因为,伴同标准不定义机器人起客户作用时的一致性要求。

B2.2 监控器类别**B2.2.1 总则**

为了像模型 3 中所示的作为主机装置起作用,机器人需要监控下一级装置。在 B2.2.2 至 B2.2.5 中描述了客户机器人为监控下一级装置的操作所提供的附加服务。

B2.2.2 状态

在监控装置中的一个关键功能是能请求其状态。

B2.2.3 非请求状态及信息报告

下一级的装置可能启动其非请求状态及信息报告。客户机器人应能接收这些报文并采取适当的活动。

B2.2.4 读

为了判断未包含在状态或信息报告中的信息,客户应该能够从下一级装置中读取变量。要求机器人必须对下一级装置中出现的数据类型有某些了解。

B2.2.5 标识

客户应该能够发出一个标识请求以取得与下一级装置有关的信息。

B2.3 控制类别**B2.3.1 总则**

控制类别的增强提供了起客户作用的机器人所需的服务,通过网络去控制下一级系统。按照正被控制的装置的唯一性决定是否增加补充要求。

B2.3.2 程序调用服务

该服务允许机器人对下一级装置启动和停止程序调用。作为一个客户这些服务的使用要求机器人清楚程序调用对实际装置的映射。对每种试图去控制的装置类型可以有补充要求。

B2.3.3 取得和放弃控制权

该服务允许机器人客户对下一级装置的信号取得和放弃控制权。当控制的下一级装置是机器人时，可以要求此服务。

B2.3.4 报告信志状态

该服务允许机器人客户请求信志状态，如果多于一个客户能够取得信志，就需要有此服务。它使机器人能知道其他哪个客户目前控制该信志。

B2.3.5 写

此服务使客户能改变被控装置的变量。

附 录 C

(提示的附录)

中 英 对 照

Acceleration Factor	加速度系数
Activation	激活
Additional Service	附加服务
Error	错误
Requests	请求
Response	应答
Additional Services	各种附加服务
AlterProgramInvocationAttributes	变更程序调用属性
Any Physical Resource Power On	任一物理资源通电
Application Reference	申请引用
ASN. 1	抽象语法表示法 1
Assign	赋值
Auxiliary Device	辅助装置
Base Class	基本类别
Base Coordinate System	机座坐标系
BASE—WORLD	机座坐标系——绝对坐标系
Brakes	制动器
Calibrate	校准
CALIBRATED	受校准
CALIBRATING	正在校准
Calibration	校准
CBB	一致性构造块
Change Mode	变更方式
Clear	清除
Clear Transition	清除转移
Clear(abandon)	清除(舍弃)
Clear(abort)	清除(异常终止)
Client	客户
Conformance	一致性
Continue Mode	连续方式
Control Class	控制类别

Control Program	控制程序
Control Systems	(各种)控制系统
Controlled	受控
Controlled Program Invocation	受控程序调用
Controlling	施控
Controlling Program Invocation	施控程序调用
Coordinate System	坐标系
Coordinate System Conventions	坐标系约定
Coordinate System Transformation	坐标系变换
Creat Program Invocation Protocol	建立程序调用协议
Cycle	循环
Cycle Count	循环计数
CYCLE—LIMITED	有限循环
Deactivation	不激活
Deassign	去值
Delete Program Invocation	删除程序调用
Deselect Transition	选弃转移
Device Calibrated	装置已校准
Device Power on	装置通电
Domain—Specific	特定域
Domains	(各个)域
Dynamic Conformance	动态一致性
Dynamic Definition Class	动态定义类别
Emergency Action	应急动作
Emergency Stop	急停
End Effector	末端执行器
End of Execution	执行结束
Enhanced Class	增强类别
Error	错误
Error Code	错误代码
Event Class	事件类别
Free Running	自由运行
FREE—RUN	“自由运行”
GetProgramInvocationAttributes	获取程序调用属性
ID Number	标识符编号
Initiate	初始化
InitRequestDetail	初始请求细目
InitResponseDetail	初始应答细目
Joint	关节
Joint Brakes	关节制动器
joint Servo	关节伺服
Joint Type	关节类型
Joint Value	关节值

Joints	(各个)关节
Load	装载
Load Transition	装载转移
Local Action	本地活动
Local Control	本地控制
Lower Bound	下限
Manipulator	操作机
MANUAL—INTERVENTION—REQUIRED	“要求手动介入”
Manufacturing Configurations	(各种)制造配置
Mapping	映射
Mechanical Interface Coordinate System	机械接口坐标系
MICS—BASE	机械接口坐标系—机座坐标系
MMS—ROBOT	制造报文规范—机器人
Monitor Class	监控类别
Motion Enabled	使能运动
Multiple Associations	多关联
NEEDS—COMMISSIONING	“需要试运行”
No Limit	无限定(数)
NO—STATE—CHANGES—ALLOWED	“不允许状态改变”
NONE	“无”
NORMAL	“正常”
NOT—CALIBRATED	“不受校准的”
NOTCALIBRATED	“无须校准的”
Number of Joints	关节编号
Object Model	对象模型
Other Productions	其他产生式
Parameter CBB	参数一致性构造块
Path Planner	路径规划器
Pause	暂停
Pause Transition	暂停转移
PICS	协议执行程序一致性语句
Power	电源
Power On	通电
PRISMATIC	棱柱形
Program Invocation management	程序调用管理
Program Location	程序位置
Program Reset	程序复位
Program Reset Transition	程序复位转移
Program Start	程序开始
Programmed Acceleration	编程加速度
Programmed Speed	编程速度
Remote Operation	远程操作

Reset	复位
Resume	恢复
Resume Service Procedure	恢复服务步骤
Resume Type	恢复类型
REVOLUTE	旋转的
Robot Arm	机器人臂
Robot Control	机器人控制
Robot Operation	机器人操作
Robot Operation States	机器人操作状态
Robot Physical Model	机器人物理模型
Robot Program	机器人程序
Robot Program Invocation	机器人程序调用
Robot Status	机器人状态
Robot System	机器人系统
Robot System Controller	机器人系统控制器
Robot VMD State	机器人虚拟制造装置状态
ROBOT—EXECUTING	“机器人执行”
ROBOT—IDLE	“机器人空闲”
ROBOT—LOADED	“机器人装载”
ROBOT—MOTION—PAUSED	“机器人运动暂停”
ROBOT—READY	“机器人准备完毕”
RUNNING	“运行”
Running Mode	运行方式
Safety Interlocks Violated	违反安全连锁
Secondary Storage	二次存储
Select	选择
Selected Controlling Program Invocation	被选施控程序调用
Server	服务器
Service Procedure	服务步骤
Service CBB2	服务一致性构造块 2
Servo mechanism	伺服机构
Speed	速度
Speed Factor	速度系数
Start	开始
Start Location	开始位置
Start Service Procedure	开始服务步骤
STATE—CHANGES—ALLOWED	“允许变更状态”
States	状态
Status	状态
Status Changs	状态变更
Step	步进,步
Step Count	步进计数
STEP—LIMITED	“限数步进”

Stop	停止
Stop Service Procedure	停止服务步骤
Subsystem	子系统
Take and Relinquish Control	取得和放弃控制权
Task Program	任务程序
Tool Coordinate System	工具坐标系
Tool Descriptor	工具描述符
TOOL—MICS	“工具坐标系——机械接口坐标系”
TOOL—USER	“工具坐标系——用户坐标系”
Transformation	变换
UNCONTROLLED	“不受控制的”
Undefined	未定义的
UnsolicitedStatus	非请求状态
UNSPECIFIED	“非专用的”
Upload/Download	上装/下装
Upper Bound	上限
User Coordinate System	用户坐标系
USER—BASE	“用户坐标系——机座坐标系”
Version Number	版本号
VMD	“虚拟制造装置”
VMD State	虚拟制造装置状态
VMD Support Services	虚拟制造装置支持服务
VMD Reset	虚拟制造装置复位
VMD Stop	虚拟制造装置停止
World Coordinate System	绝对坐标系

中 华 人 民 共 和 国
国 家 标 准
工 业 自 动 化 系 统
制 造 报 文 规 范
第 3 部 分：机 器 人 伴 同 标 准
GB/T 16720.3—1996

*

中国标准出版社出版
北京复兴门外三里河北街16号
邮政编码：100045
电 话：68522112

中国标准出版社秦皇岛印刷厂印刷
新华书店北京发行所发行 各地新华书店经售
版权专有 不得翻印

*

开本 880×1230 1/16 印张 5 字数 147 千字
1997年10月第一版 1997年10月第一次印刷
印数 1—500

*

书号：155066·1-14080 定价 31.00 元

*

标 目 318—52



GB/T 16720.3—1996