

INTERNATIONAL STANDARD

ISO/IEC
9506-3

First edition
1991-08-01

Industrial automation systems — Manufacturing message specification —

Part 3:

Companion standard for robotics

*Systèmes d'automatisation industrielle — Système de messagerie
industrielle —*

Partie 3: Norme d'accompagnement pour la robotique



Reference number
ISO/IEC 9506-3:1991(E)

AVA Documents
Document Sharing Hub

Contents

Page

1	Scope	1
2	Normative References	1
3	Definitions	2
3.1	General definitions	2
3.2	Specific definitions	2
4	Abbreviations	5
5	Robot application description	5
5.1	Manufacturing configurations	5
5.2	Robot specific model	8
5.3	Robot specific functions	14
6	Robot application specific context mapping	20
6.1	Mapping the robot model to the VMD object	20
6.2	Robot specific objects that map to Domains	24
6.3	Robot specific objects that map to Program Invocations	30
6.4	Definition of robot-specific objects which map to other MMS abstract objects	40
6.5	Definition of new MMS abstract objects to support other robot-specific objects	40
7	Robot specific services and protocol	40
7.1	Robot application context definition	40

© ISO/IEC 1991

All rights reserved. No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher.

ISO/IEC Copyright Office • Case Postale 56 • CH-1211 Genève 20 • Switzerland

Printed in Switzerland

7.2	Robot specific abstract syntax definition	40
7.3	Use of MMS services	40
7.4	Definition and use of robot specific services and protocol	62
7.5	The Initiate service and protocol	74
7.6	End of module	78
8	Robot specific standardized objects	78
8.1	General	78
8.2	Standardized Domain objects	79
8.3	Standardized Program Invocation objects	80
8.4	Standardized Named Type objects	81
8.5	Standardized Named Variable objects	82
8.6	Standardized Semaphore objects	90
8.7	Standardized Event Condition objects	91
8.8	Standardized Event Action objects	93
8.9	Standardized Event Enrollment objects	94
8.10	Standardized Operator Station objects	94
8.11	Standardized Journal objects	94
9	Robot conformance classes	94
9.1	General	94
9.2	Robot server conformance	94
9.3	Robot client conformance	98
9.4	PICS	99
A	Example (Informative)	100
A.1	Background	100
A.2	Manufacturing scenario	100
A.3	Operation	101
A.4	The robot VMD description	102
A.5	Operations using MMS services	103
B	Future issues - higher conformance classes (Informative)	110
B.1	Robot server conformance classes	110
B.2	Robot client conformance	112

Figures

Figure 1 Robot system	vii
Figure 2 Robot server/single client	6
Figure 3 Robot server/many clients	7
Figure 4 Robot client	8
Figure 5 Peer to peer	8
Figure 6 Information flow	10
Figure 7 Robot coordinate systems	12
Figure 8 Robot operation states	15
Figure 9 MMS mapping process	21
Figure 10 IDLE state diagram	34
Figure 11 Example scenario	101

Tables

Table 1 Local control	24
Table 2 Robot status detail	42
Table 3 CS-CreateProgramInvocation-Request parameter	44
Table 4 CS-Start-Request parameter	46
Table 5 CS-Resume-Request parameter	50
Table 6 CS-GetProgramInvocationAttributes-Response parameter	56
Table 7 VMDStop service	65
Table 8 VMD attributes / VMDStop	66
Table 9 VMDReset service	67
Table 10 Select service	69
Table 11 AlterProgramInvocationAttributes service	72
Table 12 Init Request Detail	74
Table 13 Init Response detail	76
Table 14 PICS Additional service CBBs	99
Table 15 PICS Local implementation values	100

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

International Standard ISO/IEC 9506-3 was prepared by Joint Technical Committee ISO/TC 184, *Industrial automation systems and integration*.

ISO/IEC 9506 consists of the following parts, under the general title *Industrial automation systems — Manufacturing message specification*:

- *Part 1: Service definition*
- *Part 2: Protocol specification*
- *Part 3: Companion standard for robotics*

Annexes A and B are for information only.

Introduction

This part of ISO/IEC 9506 is intended to be used in an open communication system employing robots and robotic systems connected to a communication network conforming to the OSI model (ISO 7498). This part of ISO/IEC 9506 also recognizes that the robot can act as a controller (client) to devices connected to it such as vision systems and grippers. Client conformance for communication to such devices is not defined by this part of ISO/IEC 9506. Conformance requirements for communication to such devices are defined by the companion Standard appropriate to that device or by ISO/IEC 9506-1 and ISO/IEC 9506-2.

This part of ISO/IEC 9506 does define conformance requirements for the robot when used in a network with multiple clients. The messages are described using the method defined in ISO 8824.

This part of ISO/IEC 9506 provides a description of several conformance classes including a base class. This base class is considered as the minimum conformance for robots connected as a "slave" or server to a host computer or client device on the network. The base class forms the "kernel" of conformance for robots in these types of networks. All other conformance classes will be additions to the base class. This part of ISO/IEC 9506 also provides the robot specific services and protocol including the abstract syntax notation for protocol elements which are undefined in the MMS-General-Module.

This part of ISO/IEC 9506 also recognizes that the robot can act as a controller to devices connected to it such as vision systems and grippers. This part of ISO/IEC 9506 identifies the requirements for communications in such a manner but does not identify MMS service and protocol conformance requirements for the robot when acting in a client role. These requirements are identified by the companion standard covering the device to which the robot intends to communicate.

MMS is intended to be used with yet other standards designed to achieve a systematic and uniform approach to Open Systems Interconnection of Information Processing Systems as defined in ISO 7498. As such, MMS is positioned within the application layer of the OSI model. It defines the Application Service Element and the protocol required to extend information systems networks to the programmable control devices of the automated factory environment. The services defined by MMS are generic and intended to be referenced by the companion standards, each of which is oriented towards a more specific class of application.

This part of ISO/IEC 9506 recognizes that safe operation of robots is required at all times. Safety requirements for robots are specified in ISO DIS 10218. All robot actions delineated in this part of ISO/IEC 9506 are permissible under the safety standard.

Implementation of this part of ISO/IEC 9506 requires a minimum implementation of MMS. This is covered in Clause 9 which references the conformance requirements of ISO/IEC 9506-1 and 2. Implementers of MMS for robots and robotic systems should have a thorough understanding of MMS for proper implementation of this part of ISO/IEC 9506. Implementers should also have a thorough understanding of the modeling, services and protocol defined in this part of ISO/IEC 9506. Users of robots and robotic systems are directed to the clauses on modeling and services found in this document.

For the purpose of this part of ISO/IEC 9506, the term "robot" means "manipulating industrial robot" as defined in ISO/TR 8373. As used in this part of ISO/IEC 9506, a robot will generally refer to the manipulator together with its control system and any ancillary equipment, devices, sensors, or communications links, necessary for the robot to perform its task. Figure 1 illustrates the elements of the robot system as described in this part of ISO/IEC 9506. Since the definitions of ISO/TR 8373 only describe robot systems with a single arm and this part of ISO/IEC 9506 anticipates robots with multiple arms operating in a coordinated fashion, these definitions have been generalized. The term "robot system controller" will refer to the (single) task program operating with the (possible multiple) control program of the robot arm(s) of the system.

"MMS services" refers to the abstract services defined in ISO/IEC 9506-1 and "MMS protocol" refers to the protocol defined in ISO/IEC 9506-2.

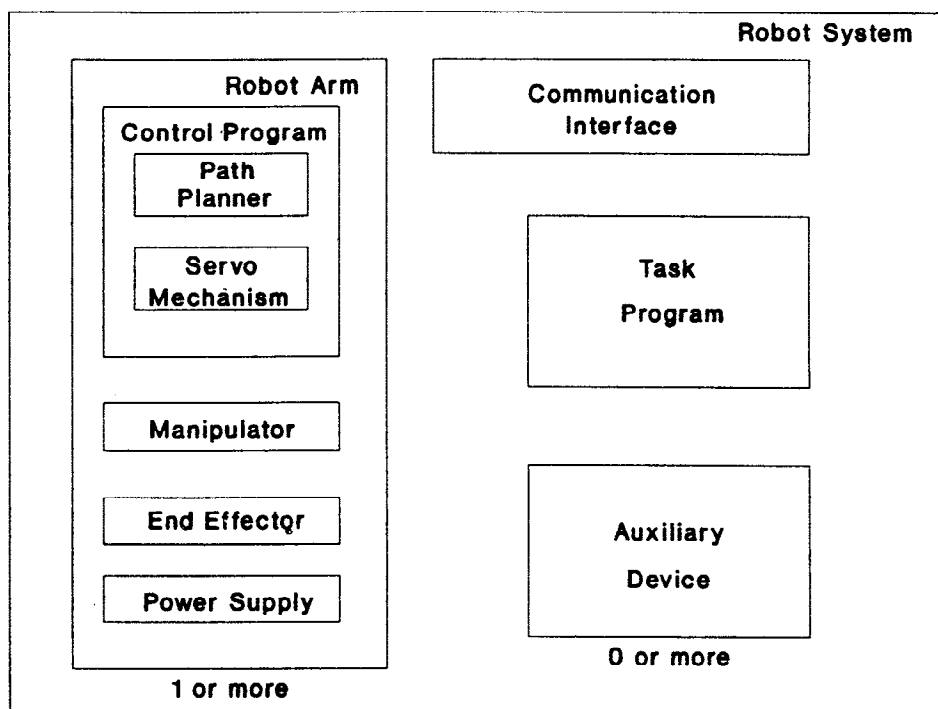


Figure 1 Robot system

This page intentionally left blank

Industrial automation systems — Manufacturing message specification —

Part 3: Companion standard for robotics

1 Scope

This part of ISO/IEC 9506:

- a) describes the model of a robot and how the attributes of the robot are mapped onto the attributes of a Virtual Manufacturing Device (VMD),
- b) defines the robot specific services and protocol including the abstract syntax notation for protocol elements requiring companion standard specification by MMS,
- c) defines robot specific standardized objects,
- d) provides a description of conformance classes including a base class and several enhanced classes.

Definitions are provided of the services and protocol of robots operating as a server in the abstract syntax defined in this part of ISO/IEC 9506. The semantics of MMS services performed by robots while communicating under other abstract syntaxes are not defined by this part of ISO/IEC 9506. This part of ISO/IEC 9506 does not identify MMS service and protocol conformance requirements for a robot acting in a client role. These requirements are intended to be identified by the companion standard covering the device to which the robot intends to communicate.

2 Normative References

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 9506. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this part of ISO/IEC 9506 are encouraged to investigate the possibility of applying the most recent editions of the standards listed below. Members of ISO and IEC maintain registers of currently valid international standards.

ISO/IEC 9506-1:1990	<i>Industrial automation systems - Manufacturing Message Specification - Part 1 - Service definition</i>
ISO/IEC 9506-2:1990	<i>Industrial automation systems - Manufacturing Message Specification - Part 2 - Protocol specification</i>
ISO 7498:1984	<i>Information processing systems - Open Systems Interconnection - Basic Reference Model</i>

ISO/IEC 9506-3: 1991(E)

ISO/TR 8509:1985	<i>Information processing systems - Open Systems Interconnection - Service Conventions</i>
ISO 8824:1987	<i>Information processing systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1)</i>
ISO 8571:1988	<i>Information processing systems - Open Systems Interconnection - File Transfer, Access and Management</i>
ISO 8649:1988	<i>Information processing systems - Open Systems Interconnection - Service definition for the Association Control Service Element</i>
ISO 8650:1988	<i>Information processing systems - Open Systems Interconnection - Protocol specification for the Association Control Service Element</i>
ISO DIS 10218 ¹	<i>Manipulating Industrial Robots - Standard for Safety</i>
ISO/TR 8373:1988	<i>Manipulating Industrial Robots - Vocabulary</i>
ISO 9787:1990	<i>Manipulating Industrial Robots - Coordinate Systems and Motions</i>

¹ To be published

3 Definitions

3.1 General definitions

Clause 3 of ISO/IEC 9506-1:1990 and clause 3 of ISO/IEC 9506-2:1990 list a number of terms defined in ISO 7498, in ISO/TR 8509, and in ISO 8824, as well as its own definitions. ISO/TR 8373 and ISO 9787 also define a number of terms used in this part of ISO/IEC 9506. These definitions are included in this part of ISO/IEC 9506 by reference.

3.2 Specific definitions

3.2.1 control program:

The inherent set of control instructions which defines the capabilities, actions, and responses of a robot system. This type of program is fixed and usually not modifiable by the user.

3.2.2 cycle:

A single execution of a task program.

3.2.3 local control:

A boolean value which indicates whether or not it is possible for remote operations to effect changes in the state of the MMS server. If local control is true, remote operations cannot change the state of the server.

Note: This definition is appropriate to MMS Companion Standards and is different from that contained in ISO DIS 10218.

3.2.4 manipulating industrial robot [robot]:

An automatically controlled, reprogrammable, multi-purpose, manipulative machine with several degrees of freedom, which can be either fixed in place or mobile for use in industrial automation applications.

Note: For the purposes of the remaining clauses of this part of ISO/IEC 9506, the term "robot" will mean "manipulating industrial robot".

3.2.5 manipulator:

A machine, the mechanism or which usually consists of a series of segments jointed or sliding relative to one another, for the purpose of grasping and/or moving objects (pieces or tools) usually in several degrees of freedom. It may be controlled by an operator, a programmable electronic controller, or any logic system (for example cam device, wired, etc.).

3.2.6 motion enabled:

A boolean value which if TRUE indicates that a valid command presented to the control program of a robot arm will result in motion of the arm.

3.2.7 pose:

Combination of position and orientation of a part of a robot (for example its mechanical interface) or of a workpiece in a coordinate system.

ISO/IEC 9506-3: 1991(E)

3.2.8 remote operation:

An operation involving data acquisition or control operating over an OSI communication network using MMS services.

3.2.9 robot arm:

As used in this part of ISO/IEC 9506, a manipulator, an end effector, its power supply, and the control program which controls the manipulator.

3.2.10 robot system:

A robot system includes:

- the robot (hardware and software) consisting of the manipulator whether mobile or not; power supply and control system;
- the end-effector(s);
- any equipment, devices, or sensors required for the robot to perform its task;
- any communication interface that is operating and monitoring the robot, equipment, or sensors, as far as these peripheral devices are supervised by the robot control system.

3.2.11 robot system controller:

The entire control system of the robot, consisting of the (single) task program and the control program(s) for the robot arm(s) and the auxiliary device(s).

3.2.12 step:

An atomic element of task program execution. It may or may not involve robot motion.

Note: The concept of a step is dependent on the robotic programming language.

3.2.13 task program:

The set of motion and auxiliary function instructions which define the specific intended task of the robot system; this type of program is normally generated by the user.

Note: an application is a general area of work, a task is specific within the application.

4 Abbreviations

The following abbreviations are used in the text of this part of ISO/IEC 9506.

ACSE:	Association Control Service Element
ASE:	Application Service Element
ASN.1:	Abstract Syntax Notation One
C:	conditional parameter
CBB:	conformance building block
Cnf:	confirm
CS:	companion standard
DIS:	Draft International Standard
FTAM:	File Transfer, Access, and Management
Ind:	indication
I/O:	input/output
IS:	International Standard
M:	mandatory
MICS:	Mechanical Interface Coordinate System
MMS:	Manufacturing Message Specification
OSI:	Open System Interconnection
PDU:	protocol data unit
PICS:	Protocol implementation conformance statement
Req:	request
Rsp:	response
S:	Selection
TR:	Technical Report
U:	user option parameter
VMD:	Virtual Manufacturing Device

5 Robot application description

5.1 Manufacturing configurations

5.1.1 General considerations

The conformance classes which are defined in clause 9 of this part of ISO/IEC 9506 are described independently of the configurations in which they are used. The configurations described in this subclause are tutorial in nature, and are included to give insight into the rationale which underlies this part of ISO/IEC 9506. Actual configurations can exhibit characteristics of several of these configurations simultaneously.

In a remote communication environment, one node is referred to as the client, the other node is referred to as the server. A host connected to a robot is considered to be a client of the robot. The host generally gives commands to, and monitors functions of, the robot. The robot is considered the server to the host. In the case of the robot connected through an OSI communication channel to intelligent peripherals, e.g., such devices as grippers, vision

ISO/IEC 9506-3: 1991(E)

systems, or other robots, the robot is viewed as a client with respect to such devices. The connections between devices described in this part of this Standard are considered to be logical connections.

Although operator panels and teach pendants can be used to direct operation of a robot, they are not considered clients in the present sense since they are not connected to the robot through OSI communication channels. Rather, they are considered part of the robot server.

This part of ISO/IEC 9506 does not impose any requirements on the client configurations when the client communicates with a robot. It requires only that the client has the capability to send the appropriate requests to the robot and to receive responses from the robot.

5.1.2 Configuration one: Robot server, single client

This configuration consists of one client (e.g., host computer) controlling, or in communication with, one robot (see Figure 2). The client, or host, sends requests to the robot, or server, to which the robot should respond. The robot can include its own subsystems such as vision and gripper controls. These subsystems will not be directly controlled via MMS and are outside the scope of Configuration one.

In simple implementations of Configuration one, there need be only one MMS association. In more sophisticated implementations, there can be multiple concurrent MMS associations.

An example is the case where two MMS associations are used between the host and the robot for increased throughput.

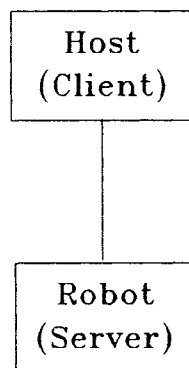


Figure 2 Robot server/single client

5.1.3 Configuration two: Robot server, many clients

In configuration two, the robot is a server, but there are many clients (host computers). See Figure 3. Support for multiple concurrent associations is now required on the part of the robot server, since any of the client hosts can initiate an association.

A multiple client configuration requires a mechanism for taking and relinquishing control of the robot. Without this capability, there is no means for preventing two or more clients from attempting to control the robot simultaneously.

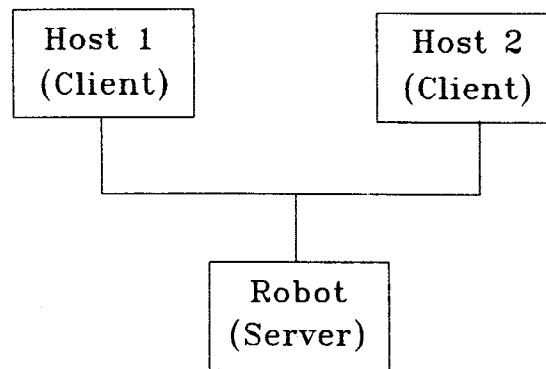


Figure 3 Robot server/many clients

An example of this configuration is a host performing robot control and a second host monitoring the robot as part of a larger system.

5.1.4 Configuration three: Robot client

In configuration three, the robot is a client to one or more devices (see Figure 4). It is also possible that the robot is simultaneously a server to one or more host clients as in configurations one and two respectively. The robot in a client capacity requires the ability to act as an initiator of requests, rather than just as a responder.

This part of ISO/IEC 9506 addresses only the interactions between a system acting as a client and robots acting as servers, and does not define the interactions with other devices such as vision systems, grippers, etc.. These interactions can only be defined based on the requirements of those devices.

An example of this configuration is a robot acting as a client to another system acting as a file server.

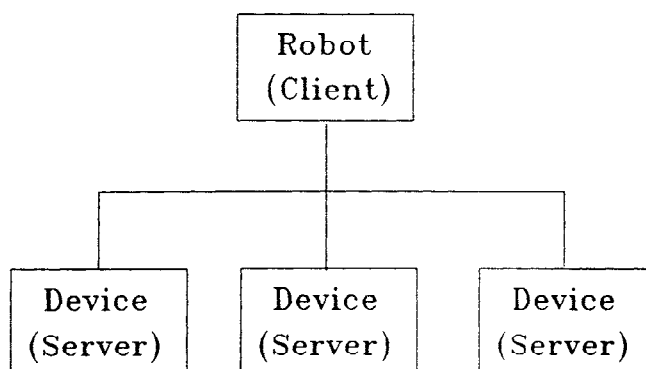


Figure 4 Robot client

5.1.5 Configuration four: Peer to peer

In configuration four, several robots are considered peers, and can act as both clients and servers (see Figure 5). In this configuration each robot includes both client and server capabilities.

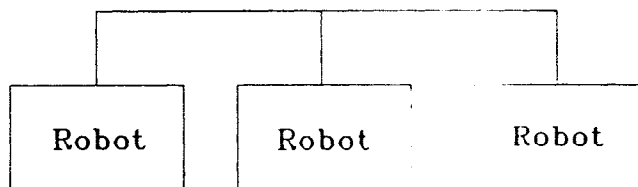


Figure 5 Peer to peer

5.2 Robot specific model

5.2.1 Robot physical model

5.2.1.1 General

This part of ISO/IEC 9506 describes an abstract model of a robot system. The attributes of this model are required to describe the activity of the robot as viewed from a communications channel. Any real robot system can have many more characteristics than are described here. Nor is it required that a robot system have all of the attributes described in this part of ISO/IEC 9506.

A robot system is composed of one or more robot arms together with a robot system controller. There can also be auxiliary devices whose activities are coordinated with the robot arm but are physically and logically separate from the robot arm. In particular, the safety interlocks are one such auxiliary device of which the Emergency Stop Button can be considered to be a component.

5.2.1.2 Robot arm

5.2.1.2.1 Robot arm subsystems

The central element of a robot system is the robot arm which is composed of the manipulator, its power supply, the robot arm control program, and the end effector. The focus of this part of ISO/IEC 9506 is the remote operation of this robot arm, together with coordinated control of the auxiliary devices.

The manipulator is composed of a set of mechanical linkages and joints. Each link along with its associated joint comprises an axis of the robot. The joint is driven by an actuator which is controlled by the robot arm control program. This robot arm control program can be considered to have two main components, a servomechanism and a path planner.

There are characteristics of the robot arm modeled in this part of ISO/IEC 9506 which equally affect both the servomechanism and path planner subsystems. It is essential to know the number of joints of the manipulator and the characteristics of each joint in order to perform the robot task. Knowledge of the nature of the calibration state (calibrated, not calibrated, or calibrating) and power state (arm power on/off) is necessary for proper operation of the robot system.

The flow of information in a robot arm can be described as follows (See Figure 6):

A task program generates the programmed pose of the manipulator.

The programmed pose is then transferred to the path planner subsystem which generates the commanded state expressed in joint values.

The commanded state is then input to the servomechanism subsystem which drives the manipulator.

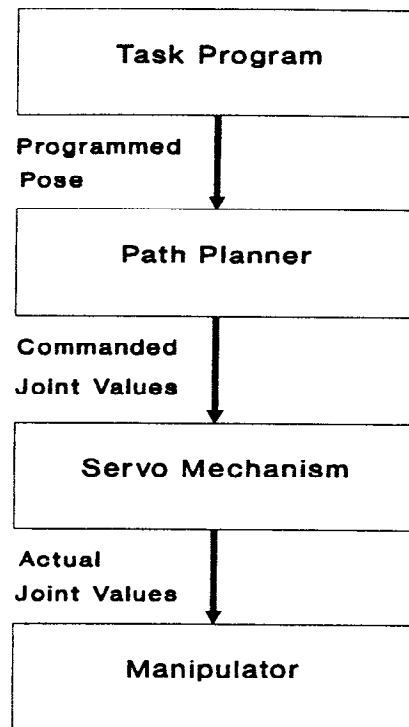


Figure 6 Information flow

5.2.1.2.2 Path planner

The path planner subsystem accepts commands from the task program (or from other local sources) for motion of the manipulator and converts these requests into a time series of servomechanism commands. In effect, the path planner is responsible for converting a desired manipulator trajectory expressed in terms of the position or speed of the tool tip into appropriate servomechanism commands. In order to express these motions in a consistent fashion, a set of coordinate systems are introduced with which to describe the robot arm. These are described in detail in 5.2.2.

The path planner is responsible for monitoring certain characteristics of the manipulator motion and applying desired modifications. Most notable of these is the control of the speed and acceleration of the tool tip.

The programmed speed is the speed normally resulting from the command input to the path planner from the task program of the robot system controller. It is the normal tool tip speed established in the task program.

The speed factor is an override multiplication factor that is applied to alter the speed of the manipulator. The speed factor will alter the programmed speed of the manipulator so that the entire manipulator motion, as directed by the task program, can be executed at a speed which is uniformly adjusted upward or downward from the programmed speed.

The programmed acceleration is the acceleration normally resulting from the commands input to the path planner from the task program of the robot system controller. It is the normal tool tip acceleration established in the task program.

The acceleration factor is an override multiplication factor that is applied to alter the acceleration of the manipulator. The acceleration factor will alter the programmed acceleration of the manipulator so that the entire manipulator motion, as controlled by the task program, can be executed with an acceleration which is uniformly adjusted upward or downward from the programmed acceleration.

Additional attributes of the path planner include a set of input values and output values. The input values to the path planner describe the programmed state of the manipulator expressed in Euclidean space including position, velocity, and acceleration. The output values from a path planner describe the commanded state of the manipulator expressed in joint space.

5.2.1.2.3 Robot arm servomechanism

The robot arm servomechanism subsystem consists of a coupled set of servomechanisms, one for each joint of the manipulator. The joints of a manipulator can be of revolute or prismatic type. Each joint can be individually calibrated, can have upper and lower bounds on travel, can have a brake mechanism, and has an associated set of link parameters relating it to the other joints in the kinematic chain which comprise the robot arm. Each joint is controlled by a servomechanism which sends control signals to the joint actuator and may obtain feedback information from joint sensors. The servomechanism issues commands to drive the joint to a desired value and monitors the actual value as it moves.

5.2.1.3 Auxiliary devices

The robot auxiliary devices are robot task related subsystems that are directly controlled by the robot system controller. The control of auxiliary devices is integrated into the control of the robot task. The information used in auxiliary device control, for example specific parameters of the auxiliary devices, are an integral part of the overall parameters of the robot task. Control of auxiliary devices can be implemented in one or more robot programs.

Note: Examples of auxiliary devices are safety interlocks, weld controllers, paint systems, water jet cutter systems, vision systems, sensors, and grippers.

5.2.2 Robot coordinate system

5.2.2.1 Conventions

The robot coordinate systems used by this part of ISO/IEC 9506 are the world, base and mechanical interface coordinate systems. These standardized coordinate systems are described in ISO 9787. Additional robot coordinate systems are described in the 5.2.2.2 and 5.2.2.3 as an aid to the understanding this part of ISO/IEC 9506.

ISO/IEC 9506-3: 1991(E)

The World Coordinate System establishes a fixed frame of reference for the entire manufacturing operation and is commonly used for work cell layout. The definition of this coordinate system will generally be made by the user institution. The Base Coordinate System establishes the location of the mounting plate of the robot and is generally supplied by the manufacturer of the robot. The Mechanical Interface Coordinate system is a moving coordinate system attached to the last link of the manipulator which establishes the location of the end-effector relative to the robot base position.

Two commonly used coordinate systems are introduced in addition to the standardized coordinate systems; the Tool Coordinate System, and the User Coordinate System. The Tool Coordinate System is used to define the position of a robotic tool (end effector). The User Coordinate System is used to provide an alternative frame of reference for the robot arm other than the Base Coordinate System. Figure 7 illustrates the robot coordinate system conventions.

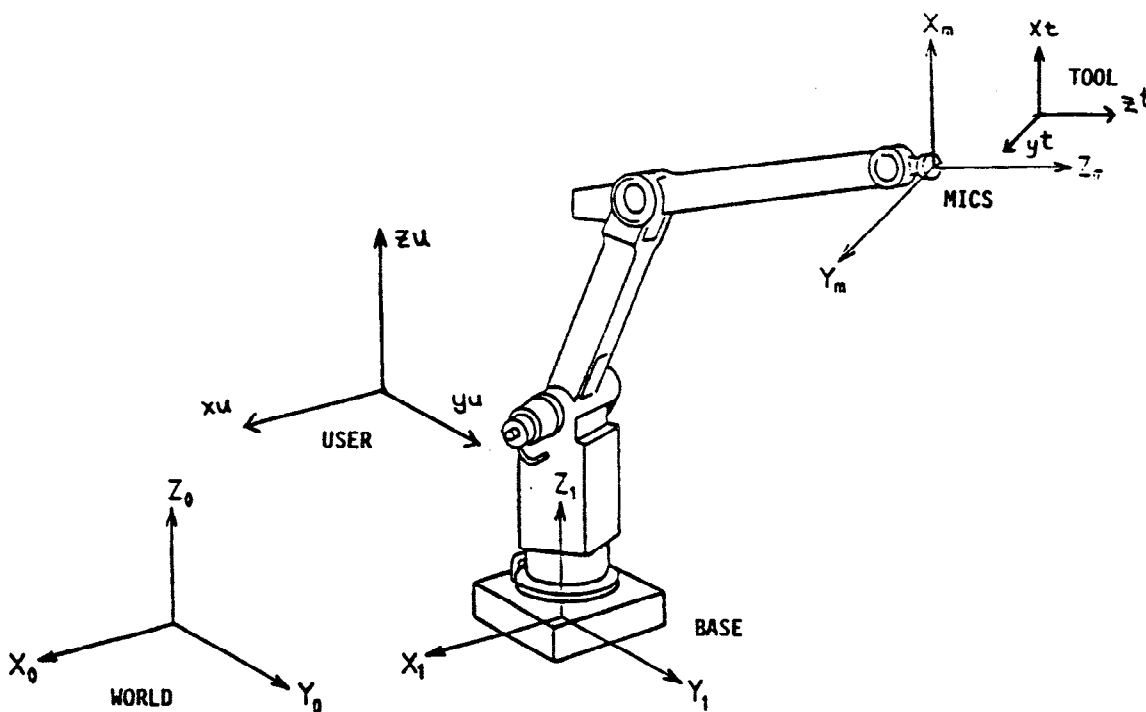


Figure 7 Robot coordinate systems

5.2.2.2 Tool Coordinate System (X_t , Y_t , Z_t)

The origin of the Tool Coordinate System is positioned at a implementor defined point on the robotic tool. The + Z_t and + X_t axes are implementor defined and perpendicular to each other. The + Y_t axis is determined by the right hand rule.

An object within the Tool Coordinate System is represented by a transformation of the form (x_t , y_t , z_t , a_t , b_t , c_t) where:

x_t represents a translation along the X_t coordinate axis.

y_t represents a translation along the Y_t coordinate axis.

z_t represents a translation along the Z_t coordinate axis.

a_t represents a counter clockwise rotation about the positive X_t coordinate axis.

b_t represents a counter clockwise rotation about the positive Y_t coordinate axis.

c_t represents a counter clockwise rotation about the positive Z_t coordinate axis.

The transformation follows a right-handed rule sign direction convention.

5.2.2.3 The User Coordinate System (X_u , Y_u , Z_u)

The origin of the User Coordinate System is positioned at a implementor defined point. The + Z_u and + X_u axes are implementor defined and perpendicular to each other. The + Y_u axis is determined by the right hand rule.

An object within the User Coordinate System is represented by a transformation of the form (x_u , y_u , z_u , a_u , b_u , c_u) where:

x_u represents a translation along the X_u coordinate axis.

y_u represents a translation along the Y_u coordinate axis.

z_u represents a translation along the Z_u coordinate axis.

a_u represents a counter clockwise rotation about the positive X_u coordinate axis.

b_u represents a counter clockwise rotation about the positive Y_u coordinate axis.

c_u represents a counter clockwise rotation about the positive Z_u coordinate axis.

The transformation follows a right-handed rule sign direction convention.

5.2.2.4 Transformations of coordinate systems

In the course of describing robot motion, it is necessary to describe the arbitrary position and orientation of objects in space. This description can be made with respect to any of the coordinate systems described above. Mathematically, the description of the position and orientation of an object with respect to some coordinate system is equivalent to the description of a transformation from the referenced coordinate system to a coordinate system embedded in the object. Thus, transformations assume a large role in the mathematics of robot motion.

Of the five coordinate systems described above, four are often used as reference coordinate systems in which to describe other objects. The Tool Coordinate System is the coordinate system embedded in the object whose position and orientation is needed. This part of this Standard identifies five transformations which are useful to describe the kinematic relationship of the components of a robot system. These are:

- o The Base Coordinate System with respect to World Coordinate System transformation.
- o The Mechanical Interface Coordinate System with respect to Base Coordinate System transformation.
- o The User Coordinate System with respect to Base Coordinate System transformation.
- o The Tool Coordinate System with respect to the User Coordinate System transformation.
- o The Tool Coordinate System with respect to the Mechanical Interface Coordinate System transformation.

5.3 Robot specific functions

5.3.1 Control systems

5.3.1.1 Robot system controller

An essential element of the robot system is the robot system controller which represents the intelligence governing the operation of the robot. The robot system controller is usually implemented through the adaptation of a multi-purpose computer running one or more computer programs, but it can also be implemented by other means. In computer based implementations, it may be the case that the underlying computer is used for more functions than the control of the robot but this part of ISO/IEC 9506 is solely concerned with those aspects of the computer operation which relate to robot operation. Robot operation, in this sense, includes the operation of all auxiliary devices which are necessary for the robot to perform its function.

An example of such a computer system which lies beyond the scope of this part of ISO/IEC 9506 is an integrated cell controller which directs a robot's activity as one part of its operation.

5.3.1.2 Coordination of robot control

In many cases control over the robot operation can be exercised from multiple locations. The introduction of communication facilities adds another potential point of control. In order to preclude conflicts in control, the robot is assumed to possess a method of assigning control uniquely to one control point. This control assignment functions as a mutually exclusive semaphore.

5.3.1.3 Robot operation states

5.3.1.3.1 Robot operation state diagram

The robot system is modeled by a finite state automaton as described in Figure 8.

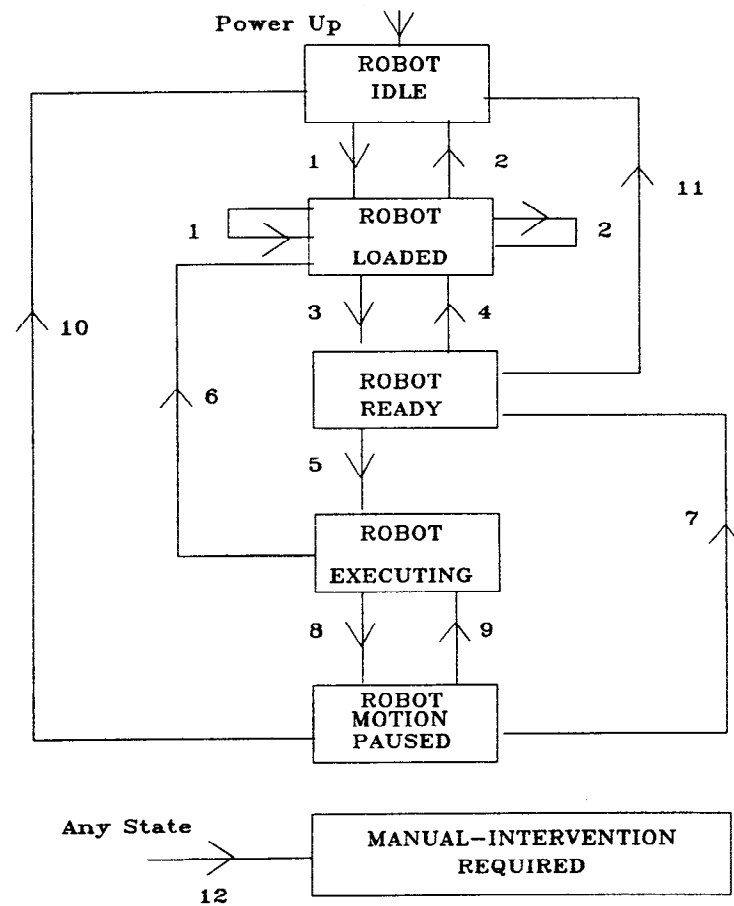


Figure 8 Robot operation states

ISO/IEC 9506-3: 1991(E)

Transition Descriptions for Figure 8.

- | | |
|----------------------|---|
| 1. Load | Some task program is placed into the robot system controller. |
| 2. Clear | The task program is removed from the robot system controller. |
| 3. Assign | One loaded task program is made ready for execution. |
| 4. Deassign | The inverse of Assign. |
| 5. Program Start | The robot is placed into the ROBOT-EXECUTING state. |
| 6. End of Execution | The robot motion is halted and the robot is returned to the ROBOT-LOADED state. |
| 7. Program Reset | A paused robot is returned to the ROBOT-READY state. |
| 8. Pause | The robot has temporarily ceased motion. |
| 9. Continue | A paused robot has been returned to the ROBOT-EXECUTING state. |
| 10. Clear(abort) | Same as clear, except the task program has not completed. |
| 11. Clear(abandon) | Same as clear, except the task program has not started. |
| 12. Emergency Action | Places the robot into a state in which manual intervention is required. |

5.3.1.3.2 Robot operation state description

5.3.1.3.2.1 ROBOT-IDLE

The ROBOT-IDLE state corresponds to the state of the robot when the robot has no task program available within the system. This state will occur when the robot is initially powered up or when it has completed a task program execution and the task program is removed from the system.

5.3.1.3.2.2 ROBOT-LOADED

The ROBOT-LOADED state corresponds to the state of the robot when one or more task programs are available within the system.

5.3.1.3.2.3 ROBOT-READY

The ROBOT-READY state indicates that one of the task programs available for execution by the robot has been assigned.

5.3.1.3.2.4 ROBOT-EXECUTING

In the ROBOT-EXECUTING state the robot is in operation and its motion is enabled.

5.3.1.3.2.5 ROBOT-MOTION-PAUSED

In the ROBOT-MOTION-PAUSED state motion is not enabled. The task program of the robot can be running or it can be stopped.

5.3.1.3.2.6 MANUAL-INTERVENTION-REQUIRED

In this state, all motion has ceased. In order for any other action to be performed, some local action is required by a machine operator. Following this local action, the robot may be in any of the other states.

5.3.1.3.3 Robot operation state transitions**5.3.1.3.3.1 General**

In the description of the transitions which follow, it is assumed that all these state transitions may occur through local action. Many of these transitions can also be accomplished through remote actions. The local actions which cause these transitions will not be further described. Remote actions using MMS services which cause these transitions are described in clause 6.

5.3.1.3.3.2 Load

The Load transition describes the act of loading a task program into the robot system controller. The robot system controller may have several task programs within its memory; for the purposes of this transition, however, only task programs which can operate and control the motion of the robot arm are considered. If there are no such programs within the robot system controller, this transition will take the robot from ROBOT-IDLE to ROBOT-LOADED state. If there is at least one such task program in memory, the robot remains in ROBOT-LOADED state as a result of this transition.

5.3.1.3.3.3 Clear

This transition occurs when a task program is removed from the robot system controller. If this task program is the last or the only such program, the robot is placed in ROBOT-IDLE state. Otherwise, it continues in ROBOT-LOADED with the remaining task programs.

5.3.1.3.3.4 Assign

The process of assigning one of the possible task programs for actual execution will place the robot in ROBOT-READY state. For robots which can only contain one task program the Load and Assign transitions may be combined into a single operation.

ISO/IEC 9506-3: 1991(E)

5.3.1.3.3.5 Deassign

To move the robot from the ROBOT-READY state to the ROBOT-LOADED state requires a Deassign transition. This is the inverse of the Assign transition. Some robots may appear to allow an Assign transition from the ROBOT-READY state. This can always be modeled as a Deassign followed by an Assign transition.

5.3.1.3.3.6 Program Start

The Program Start is the fundamental operation of placing the robot in motion. As a result of this transition, the robot moves into the ROBOT-EXECUTING state.

5.3.1.3.3.7 End of Execution

As a result of local actions within the robot system controller, the robot may move out of the ROBOT-EXECUTING state into the ROBOT-LOADED state. Before the robot can be placed in the ROBOT-EXECUTING state again, an Assign transition is required.

5.3.1.3.3.8 Program Reset

A robot which is in the ROBOT-MOTION-PAUSED state can be moved to the ROBOT-READY state by the Program Reset transition.

5.3.1.3.3.9 Pause

The Pause transition causes the robot to cease its movement. The task program may continue to execute or it may stop.

5.3.1.3.3.10 Continue

The Continue transition will move a robot from the ROBOT-MOTION-PAUSED state back into the ROBOT-EXECUTING state.

5.3.1.3.3.11 Clear (Abort)

The Clear transition can also be made from the ROBOT-MOTION-PAUSED state. In this case, the current execution of the task program is aborted (i.e. abnormally terminated) and the assigned task program is deassigned. All potential task programs are removed from the robot system controller.

5.3.1.3.3.12 Clear (Abandon)

The Clear transition can also be made from the ROBOT-READY state. In this case, the current execution of the task program is aborted (i.e. abnormally terminated) and the assigned task program is deassigned. All potential task programs are removed from the robot system controller.

5.3.1.3.3.13 Emergency Action

The Emergency Action transition may occur from any state into the MANUAL-INTERVENTION-REQUIRED state as a result of local or remote action. Recovery from this state can never be automatic and shall require the explicit action of a human operator to place the robot into some other state.

5.3.1.4 Local control

In addition to the Robot State, the robot can be described as either being in local control or not being in local control. Being in local control means that some local agent, e.g. a human operator at a control console or at a teach pendant or equivalent, is exercising control over what motions the robot may make or over what task programs the robot may run. If the robot is not in local control, the robot can still be running a task program, but the ability to stop or otherwise modify that running task program does not reside in any local agent. Robots which are capable of having remote controllers may then have such remote controllers take control of the robot.

5.3.2 Task program execution - cycles and steps

The operation of a robot may be decomposed into cycles and steps. A cycle is a sequence of operations which is repeated regularly. Each cycle of robot operation completes the task for which the robot is programmed. The cycle is the natural unit of robot operation.

Within a cycle, more atomic actions of the robot can be distinguished. These atomic actions are referred to as steps, and correspond to some elementary motion of the robot or to some elementary action of the task program or of one of the auxiliary devices. Fundamental to this notion of steps is the idea of sequential operation; a robot is composed of systems elements each of which performs elementary operations in a time ordered sequence. Not all robot systems allow control based on individual steps; for those that do, this part of ISO/IEC 9506 provides control procedures for handling individual steps.

When the robot begins operation, it can be set to operate for a fixed number of cycles, or (if in step mode) for a fixed number of steps within a cycle. This latter mode is normally used only for debugging. Alternatively, the robot can be placed in operation to perform the same cycle repetitively, without count, until some external event terminates the process.

5.3.3 Calibrate

In addition to normal operation, most robots have an auxiliary function available for calibration of the axes of the robot. This function is considered to operate in a manner similar to a normal task program, subject to the same state diagram (Figure 8) except that the presence of the calibration function does not place the robot in the LOADED state.

6 Robot application specific context mapping

6.1 Mapping the robot model to the VMD object

6.1.1 Description of mapping procedure

This clause relates the general model of the robot, developed in clause 5, to the abstract model of the Virtual Manufacturing Device (VMD) described in ISO/IEC 9506-1. The uniform application of these abstract concepts to real systems is essential if the final goal of inter-operability is to be achieved. While this part of ISO/IEC 9506 cannot anticipate every variant of robot systems which are or could be manufactured, it provides general guidelines for associating elements of real systems with the abstract model that should be applicable for most robots.

The mapping process is a multi-step procedure including the following steps:

- a) An abstract model of a set of real physical systems is first constructed. In this case, this is the model developed in clause 5.
- b) From the abstract model a set of abstract objects and attributes is constructed.
- c) The set of attributes is related to a set of attributes of an MMS abstract object. If necessary, extensions to the MMS abstract object are defined to permit this relation.
- d) Finally, the device specific abstract object is related to the MMS abstract object.

This process is illustrated in Figure 9.

In order to accommodate all the features of the robot model, extensions to the model of Program Invocations are developed in 6.3. These extensions permit the description of interworking between two classes of Program Invocations referred to as 'Controlling' and 'Controlled'. The Controlled Program Invocation is used to model the operation of the physical device controllers including the robot arm control program. The Controlling Program Invocation is used to model that specific Program Invocation which directs the operation of the robot system. While a VMD can have several Controlling Program Invocations in existence, only one is identified as being selected for operation at any specific time. In addition to the association of physical devices with Controlled Program Invocations, each physical device is also mapped to a predefined Domain. This mapping is described in 6.2.

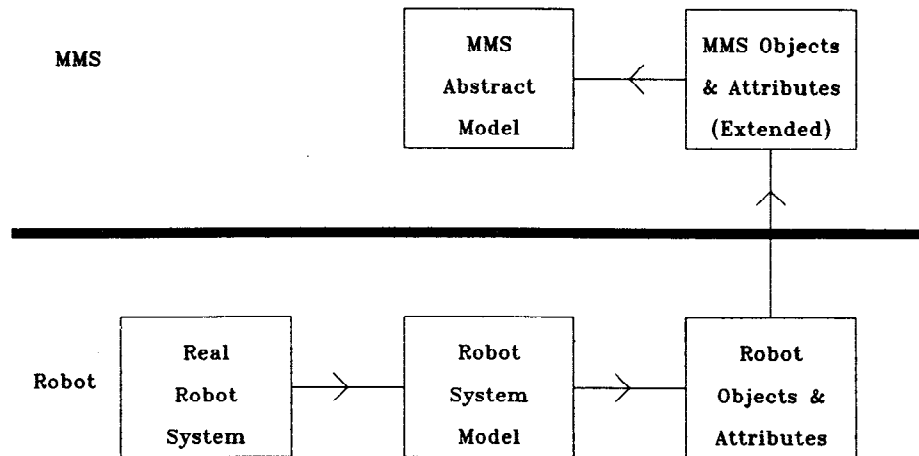


Figure 9 MMS mapping process

6.1.2 Robot system characterization

6.1.2.1 Single arm robots

From the modeling point of view, the simplest robot systems are those with only one arm. The VMD is associated with the entire robot system, robot arm, robot system controller, and any auxiliary devices. The robot arm control program is modeled as a Controlled Program Invocation associated with the motion hardware.

6.1.2.2 Multiple coordinated arm robots

There are robot systems which employ more than one arm. If a system has a single controlling intelligence such that the motion of the several arms is always conducted in a coordinated fashion, then such a system may be modeled as a single VMD. In this case, there will be a single Controlling Program Invocation which controls the motions of both arms.

This modeling is independent of the visual appearance of the robot system. Two separate pieces of hardware can be mounted at two separate locations. If the action of the two is always in close coordination, a single VMD is the proper modeling concept.

6.1.2.3 Multiple independent arm robots

There are systems in which multiple arms are employed within the same location, but for which the motion of the arms is independent, that is, there is a separate controlling intelligence for each arm. In such cases, the system should be modeled as having multiple VMDs; one for each controlling intelligence. In this case, there will be a

ISO/IEC 9506-3: 1991(E)

single Controlling Program Invocation which controls motion in each VMD. Note that the distinction between the two cases depends greatly on the configuration of the robot system controller. It is possible that the same piece of hardware can be configured either as a multiple independent arm robot or as a multiple coordinated arm robot. In such cases, the appropriate model will be different for each configuration.

6.1.2.4 Auxiliary devices

The tasks which operate the auxiliary devices do so in a coordinated manner with the robot arm control program and can be thought of as part of a single controlling task. They are modeled as additional Controlled Program Invocations.

6.1.3 The Robot VMD

6.1.3.1 Robot VMD attribute model

Given the association of the VMD with the robot system, additional attributes of the VMD (beyond those given by ISO/IEC 9506-1) have been defined to describe the robot. Since the essential nature of the VMD is the coordinated control of the robot arm and the auxiliary devices, the Controlling Program Invocation which provides this control is identified in the VMD. The Controlling Program Invocation will, during the course of its execution, produce outputs which will be used as input to the Controlled Program Invocations:

The Robot Operation State is derived from the condition of the hardware (whether motion is enabled or not) as represented by a set of Controlled Program Invocations, and from the state of the selected Controlling Program Invocation (See 6.3).

Object: VMD

All MMS defined Attributes

Attribute:	Safety Interlocks Violated (TRUE, FALSE)
Attribute:	Robot Operation State (ROBOT-IDLE, ROBOT-LOADED, ROBOT-READY, ROBOT-EXECUTING, ROBOT-PAUSED, MANUAL-INTERVENTION-REQUIRED)
Attribute:	Any Physical Resource Power On (TRUE, FALSE)
Attribute:	All Physical Resources Calibrated (TRUE, FALSE)
Attribute:	Local Control (TRUE, FALSE)
Attribute:	Reference to Selected Controlling Program Invocation

6.1.3.2 Robot VMD attribute description

6.1.3.2.1 Safety Interlocks Violated

This attribute, of type boolean, indicates the state of the safety interlocks. If the value of this attribute is TRUE, the safety interlocks have been violated since the robot system was last reset. The method of resetting the safety interlocks is a local matter.

6.1.3.2.2 Robot Operation State

This attribute is the state of the robot system as described in 5.3.1.3. The relationship of this attribute to the physical devices and to the state of the Controlling Program Invocations is defined in 6.3.4.

6.1.3.2.3 Any Physical Resource Power On

This attribute, of type boolean, indicates whether (TRUE) or not (FALSE) any physical resource has power applied to it. This attribute is the logical 'OR' of a similar primitive attribute of each physical resource in the system.

6.1.3.2.4 All Physical Resources Calibrated

This attribute, of type boolean, indicates whether (TRUE) or not (FALSE) all the physical resources in the system which have calibration attributes have the value equal to CALIBRATED (See 6.2.2.4 and 6.2.3.4).

6.1.3.2.5 Local Control

This attribute, of type boolean, indicates whether (TRUE) or not (FALSE) some local agent has control of any of the physical resources of the system. Having control indicates the ability to cause actions which change the physical resources and hence the attributes which represent those resources. Local Control can reflect either a human operator or some automatic procedure, either of which inhibit the assertion of control remotely.

If Local Control is FALSE, control may reside in some remote agent. This condition can also reflect a condition in which the robot is running under no direct control other than its task program.

The value of the Local Control attribute, the Robot VMD State, and the VMD Logical Status are interrelated. Table 1 illustrates the relationship between these attributes.

Table 1 Local control

Local Control	Robot Operation State	VMD Logical Status
TRUE	Any	NO-STATE-CHANGES-ALLOWED or LIMITED-SERVICES-PERMITTED or SUPPORT-SERVICES-ALLOWED
FALSE	MANUAL- INTERVENTION- REQUIRED	NO-STATE-CHANGES-ALLOWED or LIMITED-SERVICES-PERMITTED or SUPPORT-SERVICES-ALLOWED
	Any other	Any

Note: While in Local Control, the VMD State can continue to change. However, it is a local matter how the state changes occur.

6.1.3.2.6 Selected Controlling Program Invocation

This attribute identifies the Program Invocation which has been selected to control the operation of the robot. This Program Invocation represents the assigned task program (See 5.3.1.3.3.4). This Program Invocation has its Control attribute equal to CONTROLLING (See 6.3.2.3) and has been selected through the use of the Select operation (See 7.4.4). If there is no such Program Invocation identified, this attribute shall have the value NONE.

6.2 Robot specific objects that map to Domains

6.2.1 General

The robot, as modeled in this part of ISO/IEC 9506, consists of a set of physical resources representing the robot arm (or arms) and auxiliary devices. Each physical resource or group of resources shall be mapped to a Domain which represents that resource. Each Domain shall have attributes associated with the control of the resource, state of the resource's power, and may have an attribute which indicates the state of calibration of the resource.

The Robot Arm standardized Domain shall be associated with the robot arm resource, or in the case of multiple arms, one Domain shall be associated with each such arm. This Domain shall include within its definition all peripheral elements which are normally part of the robot arm.

Auxiliary devices which are controlled by separate programs shall be modeled by separate predefined Domains. In particular, the safety interlocks shall be modeled by one such Domain.

These Domains shall be bound to the underlying resources for all data reporting services (that is, for read services). For services which effect control or change of state of the device, the binding shall be moderated by the Local Control attribute of the VMD.

There are no extensions defined for the MMS Domain; the attributes of the robot specific objects shall be realized by Domain specific Named Variable objects defined within the Domain. Naming standards for such Named Variable objects are prescribed in clause 8.

6.2.2 The Robot arm resource

6.2.2.1 Robot arm attribute model

A robot arm includes all elements of the arm that are normally controlled by the robot arm control program. The attributes of the Robot Arm object describe the robot arm from a kinematic and control perspective.

This abstract model contains only those attributes required for remote operation of the robot. Any real robot has many more attributes that are not included in this part of ISO/IEC 9506.

ISO/IEC 9506-3: 1991(E)

Object: Robot Arm

Attribute : Local Control (TRUE, FALSE)
Attribute : Device Power On (TRUE, FALSE)
Attribute : Device Calibrated (CALIBRATED, NOT-CALIBRATED, CALIBRATING)
Attribute : Motion Enabled (TRUE, FALSE)
Attribute : Number Of Joints - Integer
Attribute : BASE-WORLD - Pose
Attribute : Servomechanism
 Attribute : MICS-BASE - pose
Attribute : Ordered List Of Joint Description
 Attribute : Joint Type (REVOLUTE, PRISMATIC)
Attribute : Calibrated (CALIBRATED, NOTCALIBRATED, CALIBRATING)
Attribute : Joint Brakes (TRUE, FALSE)
Constraint : Joint Brakes = TRUE
 Attribute : Brakes On (TRUE, FALSE)
Attribute : Upper Bound - Floating_point
Attribute : Lower Bound - Floating_point
Attribute : Joint Servo
 Attribute : Actual Joint Value - Floating_point
Attribute : Path Planner
 Attribute : USER-BASE - Pose
Attribute : Desired TOOL-USER - Pose
Attribute : Speed Factor - Floating_point
Attribute : Programmed Speed - Floating_point
Attribute : Acceleration Factor - Floating_point
Attribute : Programmed Acceleration - Floating_point
Attribute : End Effector
 Attribute : ID Number
Attribute : Tool Descriptor
Attribute : TOOL-MICS - Pose

Note: Robot terminology normally considers an axis to represent a joint and an associated link. Link information is a local matter. For the purposes of this abstract model joints and axes are identical.

6.2.2.2 Local Control

This attribute, of type boolean, indicates whether (TRUE) or not (FALSE) the robot arm is in local control.

6.2.2.3 Device Power On

This attribute, of type boolean, indicates whether (TRUE) or not (FALSE) the robot arm has power applied to it.

6.2.2.4 Device Calibrated

This attribute indicates whether the robot arm as a unit is calibrated. This attribute has the value CALIBRATED if all joints are calibrated, CALIBRATING if any joint is calibrating, otherwise NOT-CALIBRATED.

6.2.2.5 Motion Enabled

This attribute, of type boolean, indicates whether (TRUE) or not (FALSE) the robot arm (or any part of it) will move if a valid command signal is presented to the path planner. The robot arm may be moving autonomously if and only if the Motion Enabled attribute is TRUE.

6.2.2.6 Number Of Joints

This attribute, of type integer, indicates the number of joints of the robot arm.

6.2.2.7 BASE-WORLD

This attribute, of type Pose, indicates the value of the transformation of the Base Coordinate System with respect to the World Coordinate System.

6.2.2.8 Servomechanism**6.2.2.8.1 MICS-BASE**

This attribute, of type Pose, indicates the actual value of the transformation of the Mechanical Interface Coordinate System with respect to the Base Coordinate System. It is derived from the Actual Joint Value attribute and implementation specific parameters.

6.2.2.8.2 Ordered List Of Joint Description**6.2.2.8.2.1 Joint Type**

This attribute indicates whether the joint mechanism of the joint is revolute or prismatic.

ISO/IEC 9506-3: 1991(E)

6.2.2.8.2.2 **Calibrated**

This attribute indicates whether the joint is CALIBRATED, NOT-CALIBRATED, or CALIBRATING.

6.2.2.8.2.3 **Joint Brakes**

This attribute, of type boolean, indicates whether (TRUE) or not (FALSE) brakes exist for this joint. If brakes exist (Joint Brakes is TRUE), then the attribute Brakes On exists. The Brakes On attribute, of type boolean, indicates whether (TRUE) or not (FALSE) the brakes are applied to restrict motion.

6.2.2.8.2.4 **Upper Bound**

This attribute, of type floating point, indicates the upper limit of travel for the joint.

6.2.2.8.2.5 **Lower Bound**

This attribute, of type floating point, indicates the lower limit of travel for the joint.

6.2.2.8.2.6 **Joint Servo - Actual Joint Value**

This attribute, of type floating point, indicates the actual value of the joint parameter. The value will be the linear position or angular position depending upon the Joint Type.

6.2.2.9 **Path Planner**

6.2.2.9.1 **USER-BASE**

This attribute, of type Pose, indicates the value of the transformation of the User Coordinate System with respect to the Base Coordinate System.

6.2.2.9.2 **Desired TOOL-USER**

This attribute, of type Pose, indicates the value of the desired pose of the Tool Coordinate System with respect to the User Coordinate System.

6.2.2.9.3 Speed Factor

This attribute, of type floating point, identifies the override speed factor to be applied to the Programmed Speed input to the path planner.

6.2.2.9.4 Programmed Speed

This attribute, of type floating point, identifies the magnitude of the programmed speed of the MICS coordinate system with respect to the Base coordinate system as designated in the robot program.

6.2.2.9.5 Acceleration Factor

This attribute, of type floating point, identifies the override acceleration factor to be applied to the Programmed Acceleration input to the path planner.

6.2.2.9.6 Programmed Acceleration

This attribute, of type floating point, identifies the magnitude of the programmed acceleration of the MICS coordinate system with respect to the Base coordinate system as designated in the robot program.

6.2.2.10 End Effector

6.2.2.10.1 ID Number

This attribute, of type object identifier, identifies uniquely the end effector from all other end effectors within a certain set.

6.2.2.10.2 Tool Descriptor

This attribute provides a text description of the end effector.

6.2.2.10.3 TOOL-MICS

This attribute, of type Pose, indicates the value of the transformation of the Tool Coordinate System with respect to the Mechanical Interface Coordinate System.

ISO/IEC 9506-3: 1991(E)

6.2.3 Robot auxiliary device resource

6.2.3.1 Auxiliary attribute model

All physical resources separate from the robot arm resource are modeled as auxiliary device resources. The Auxiliary Device object describes the attributes of the physical resource.

Object: Auxiliary Device

Attribute : Local Control (TRUE, FALSE)

Attribute : Device Power On (TRUE, FALSE)

Attribute : Device Calibrated (CALIBRATED, NOT-CALIBRATED, CALIBRATING)

6.2.3.2 Local Control

This attribute, of type boolean, indicates whether (TRUE) or not (FALSE) the auxiliary device resource is in local control.

6.2.3.3 Device Power On

This attribute, of type boolean, indicates whether (TRUE) or not (FALSE) the auxiliary device resource has power applied to it.

6.2.3.4 Device Calibrated

This attribute indicates whether the auxiliary device resource as a unit is calibrated. If calibration does not apply to a particular auxiliary device, then this attribute does not exist.

6.3 Robot specific objects that map to Program Invocations

6.3.1 Robot specific Program Invocations

The model of a Program Invocation is augmented to allow the concept of interworking of two related Program Invocations in the same VMD. In such a pair of interworking Program Invocations, one Program Invocation is referred to as the Controlling Program Invocation, the other as the Controlled Program Invocation. The two Program Invocations are modeled as having an inter-process communication channel available to them over which they pass messages which provide both data and synchronization of the processes. However, such messages are not visible to the remote MMS user except insofar as they cause the state of the Program Invocation to be altered.

The relationship between these two types of Program Invocations is a many to one relationship. A Controlling Program Invocation can control many Controlled Program Invocations; a Controlled Program Invocation can be controlled by only one Controlling Program Invocation at any specific time.

6.3.2 Extensions to the Program Invocation object model

6.3.2.1 Program Invocation attribute model

The model of the Program Invocation given by ISO/IEC 9506-1 shall be augmented by the following attributes:

Object: Program Invocation

Attribute: All MMS Attributes

Attribute: Error Code - Integer

Attribute: Control (CONTROLLING, CONTROLLED, NORMAL)

Constraint: Control = CONTROLLING

Attribute: List of References to Controlled Program Invocations

Attribute: Program Location - String

Attribute: Running Mode (FREE-RUN, CYCLE-LIMITED, STEP-LIMITED)

Constraint: Running Mode = CYCLE-LIMITED

Attribute: Remaining Cycle Count - Integer

Constraint: Running Mode = STEP-LIMITED

Attribute: Remaining Step Count - Integer

Constraint: Control = CONTROLLED

Attribute: Reference to Controlling Program Invocation

6.3.2.2 Error Code

This attribute, of type integer, shall identify the last recorded error of the Program Invocation execution. A value of zero indicates that no error has been recorded. The meaning of other values is a local matter, as is the method of resetting this attribute.

6.3.2.3 Control

This attribute indicates whether (CONTROLLING) this Program Invocation is intended to be coupled to another Program Invocation as being in control, or whether (CONTROLLED) this Program Invocation, after coupling, normally receives control information from another Program Invocation (the Controlling Program Invocation). If neither case applies, this attribute shall have the value NORMAL.

When two Program Invocations are coupled in this manner, the service indications for the Start and Resume service requests received by a Controlling Program Invocation can cause state transitions in the Controlled Program Invocations which are coupled to the Controlling Program Invocation. (See 7.3.3.3 and 7.3.3.5) The result reported by a Controlling Program Invocation can contain information which reflects the effect of the service request on the Controlled Program Invocations.