

INTERNATIONAL STANDARD

ISO/IEC
9506-4

First edition
1992-12-15

Industrial automation systems — Manufacturing Message Specification —

Part 4:

Companion standard for numerical control

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

Partie 4: Norme d'accompagnement pour la commande numérique



Reference number
ISO/IEC 9506-4:1992(E)



Documents
Document Sharing Hub

Table of Contents

Foreword x

Introduction xi

1 Scope 1

2 Normative References 1

3 Definitions 2

3.1 Alarm 2

3.2 Device 2

3.3 Numerical Control (NC) 2

3.4 Numerical Controller 2

3.5 NC-CS (NC Companion Standard) 3

3.6 NC-CS user 3

3.7 NC system 3

3.8 Operator Interface Device 3

3.9 Offset 3

3.10 Axis 3

3.11 Coordinate Axes 3

3.12 Zero Offset 3

3.13 Axis Value Transformation 3

3.14 Coordinate System 4

3.15 Actual Coordinate System 4

3.16 Coordinate Transformation 4

3.17 Geometry Transformation 4

4	Abbreviations	4
5	NC Application Description	5
5.1	NC Specific Model	5
5.1.1	NC System State Model	7
5.1.2	The NC Device Control Model	9
5.1.3	NC Information Model	15
5.1.4	Offset Model	16
5.1.5	NC Alarm Model	20
5.1.6	NC Data Store Model	21
5.2	NC Specific Functions	22
5.2.1	Data Transfer	22
5.2.2	NC Data Store Management	23
5.2.3	Process management	23
5.2.4	Interaction with an operator	24
5.2.5	Alarm Processing	24
6	NC Specific Context Mapping	25
6.1	Mapping Of The NC Specific Model To The VMD Object	25
6.1.1	Mapping Of The NC System To The VMD Model	25
6.1.2	NC VMD	28
6.2	Definition Of NC Specific Objects Which Map To Domains	29
6.2.1	NC Domain Object	29
6.2.2	Device Domain	29
6.2.3	Process Information Domain	29
6.2.4	Program Domain	29
6.2.5	Data Domain	29
6.2.6	NC Data Store Domain	30
6.2.7	Mapping Of Other Objects To Domains	32
6.3	Definition Of NC Specific Objects Which Map To Program Invocation	32
6.3.1	NC Program Invocation Object	32

6.3.2	Controlling Process Program Invocation	32
6.4	Definition Of NC Specific Objects Which Map To Other MMS Abstract Objects	35
6.4.1	Definition Of NC Specific Objects Which Map To Named Variables	35
6.4.2	Definition Of NC Specific Objects Which Map To Named Variable List Objects	37
6.4.3	Definitions Of NC Specific Objects Which Map To Named Type Objects	38
6.4.4	Definition Of NC Specific Objects Which Map To Event Condition, Event Action And Event Enrollment Objects	39
6.4.5	Mapping of the Local/Remote State to MMS objects	40
6.5	Definition Of New MMS Abstract Objects To Support Other NC Specific Objects	40
7	Services and Protocol	41
7.1	Numerical Control application context definition	41
7.2	Numerical Control abstract syntax definition	41
7.3	Use of the MMS services	41
7.3.1	Numerical Control specific ASN.1 module definition	41
7.3.2	Program invocation management services and protocol	42
7.3.3	Other productions	45
7.4	Definition and use of Application-specific Services	48
7.5	The Initiate Service and Protocol	49
7.5.1	Definition of NC-specific Initiate Service	49
7.5.2	Definition of NC-specific Initiate Protocol	50
7.6	End of Module	50
8	Standardized NC Specific Objects	51
8.1	Domain Objects	51
8.1.1	Device Domain	51
8.1.2	Program Domain	51
8.1.3	Process Information Domain	52
8.1.4	Tool Data Domain	52
8.1.5	Means Of Production Data Domain	52
8.1.6	Setup Data	53

8.1.7	Statistical Data	53
8.1.8	Probe Data Table	54
8.1.9	NC Data Store Domain	54
8.2	Program Invocation Objects	55
8.2.1	N_Referencing	55
8.3	Named Variable Objects	55
8.3.1	NC VMD Specific Named Variable Objects	55
8.3.2	NC Domain Specific Named Variable Objects	56
8.4	Scattered Access Objects	66
8.5	Named Variable List Objects	66
8.5.1	NC VMD-specific Named Variable List Objects	66
8.5.2	NC Domain-specific Named Variable List Objects	66
8.6	Named Type Objects	67
8.6.1	Data Store Entry Type	67
8.7	Semaphore Objects	68
8.7.1	Control Token	68
8.8	Operator Station Objects	68
8.9	Event Condition Objects	68
8.10	Event Action Objects	69
8.11	Event Enrollment Objects	69
8.12	Journal Objects	69
8.13	Other NC Specific Objects	69
9	Conformance	70
9.1	Conformance Class Description	70
9.1.1	Minimum Requirements For The Functional Group Data Transfer	70
9.1.2	Minimum Requirements For The Functional Group NC Information	71
9.1.3	Minimum Requirements For The Functional Group Process Management	71
9.1.4	Minimum Requirements For The Functional Group Operator Interaction	71
9.1.5	Minimum Requirements For The Functional Group Alarm Processing	71

9.1.6 Minimum Requirements For The Functional Group NC Data Store Management 71

9.2 Restrictions On MMS Optional Parameters 72

9.3 Conformance To Standardized Objects 72

9.4 Addition To MMS PICS 73

9.4.1 PICS For Functional Classes 73

9.4.2 PICS For Standardized Objects 74

Annex A: Dynamic Download and Coded Domains (Normative) 77

Annex B: Extension of ISO 9506-4 To Include NC Milling Machine Functionality
(Normative) 79

B Introduction 79

B.1 Scope 79

B.2 References 79

B.3 Definitions 79

B.4 Symbols and Abbreviations 79

B.5 Application Description 79

B.5.1 Tool Magazines, Tool Holders, Tools, Edges 80

B.6 NC Milling Specific Context Mapping 82

B.6.2 Objects Which Map To Domains 82

B.6.4.1 Objects which Map to Named Variables 83

B.8 NC Milling Specific Standardized Objects 87

B.8.2 Standardized Domains 87

B.8.3 Named Variable Objects 87

B.9 Conformance 90

B.9.4 PICS for standardized objects 90

Annex C: Extension Of ISO 9506-4 To Include Turning Machine Functionality
(Normative) 91

C Introduction 91

C.1 Scope 91

C.2 References 91

C.3 Definitions 91

C.4	Symbols and Abbreviations	91
C.5	Application Description	92
C.5.1	Part Setup	92
C.5.2	Taper Trims	92
C.6	NC Turning Specific Context Mapping	93
C.8.3	Named Variable Objects	95
C.9	Conformance	96
C.9.4	PICS for standardized objects	96
Annex D:	Extension Of ISO 9506-4 To Include Flexible System Functionality (Normative)	97
D	Introduction	97
D.1	Scope	97
D.2	References	97
D.3	Definitions	97
D.4	Symbols and Abbreviations	97
D.5	Application Description	98
D.5.1	Parking locations, partitions	98
D.5.2	Pallets, partitions, fixtures, parts	99
D.6	Flexible Systems Specific Context Mapping	101
D.6.2	Objects Which Map To Domains	101
D.6.4.1	Objects Which Map To Named Variables	101
D.8	Flexible Systems Specific Standardized Objects	106
D.8.1	Domain Objects	106
D.8.3	Named Variable Objects	106
D.9	Conformance	111
D.9.4	PICS for standardized objects	111
Annex E:	Application Examples (Informative)	112
E.1	Example TURNING CENTER	112
E.1.1	Application Description	112

E.1.2 The Turning Center NC VMD 113

E.2 Example Milling Center In A Flexible Manufacturing System 119

E.2.1 Application Description 119

E.2.2 The Milling Center NC VMD 121

E.3 Examples of Offsets 124

E.3.1 Example 1: Zero Offset 124

E.3.2 Example 2: Axis Value Transformation 125

E.3.3 Example 3: Coordinate Transformations/Active Part Coordinate System 126

E.3.4 Example 4: Geometry Transformation 127

List of Figures

Figure 1: NC System in the Manufacturing Environment	5
Figure 2: NC System Components	6
Figure 3: NC System State Model	7
Figure 4: Local/Remote State	8
Figure 5: NC Device Control Model	9
Figure 6: NC Device Control Model (Single Device)	10
Figure 7: NC Device Control Model (3 independent Devices)	11
Figure 8: NC Device State Model	12
Figure 9: NC Controlling Process State Model	13
Figure 10: NC Information Model	15
Figure 11: Offset Classification Hierarchy	16
Figure 12: Zero Offset and Virtual Zero Point	17
Figure 13: Sequence of Primitive Axis Value Transformations	18
Figure 14: Coordinate Transformations	18
Figure 15: Sequence of Primitive Geometry Transformations	20
Figure 16: NC Data Store Model	22
Figure 17: NC VMD Model	26
Figure 18: Data Store Mapping	31
Figure 19: Transitions of the Controlling Process state as a result of Program Invocation services	33
Figure 20: NC VMD Model (2 Program Invocations, 1 Device Domain)	34
Figure 21: NC VMD Model (1 Program Invocation, 2 Device Domains)	35
Figure 22: NC Milling Machine	80
Figure 23: Example of Tool Magazine, Tool Holder, Tools, Edges	81
Figure 24: Tool Data Object	83
Figure 25: Turning Machine	92
Figure 26: Milling Machine In A Flexible Manufacturing System	98
Figure 27: Example for a Parking Location with 4 Partitions	99
Figure 28: Example of a Pallet with 4 Partitions	99
Figure 29: Example of a Parking Location with a Pallet	100
Figure 30: Example of Parking Location, Partition, Pallet, Fixture, and Part Relationship.	100
Figure 31: Means of Production Data Object	101
Figure 32: 4 Axes Turning Center	113
Figure 33: Turning Center NC VMD	114
Figure 34: Milling Center in Flexible System	120
Figure 35: Milling Center NC VMD	122

List of Tables

Table 1 - Correlation between Device States and Controlling Process States . .	14
Table 2 - Status Coherence	28
Table 3 - Relation between the Local/Remote State and the Value of the Variable N_Remote	40
Table 4: NC-specific CS parameters for Initiate	49

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

International Standard ISO/IEC 9506-4 was prepared by Technical Committee ISO/TC 184, *Industrial automation systems and integration*, Subcommittee SC 1, *Physical device control*.

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*
- *Part 4: Companion standard for numerical control*

Annexes A to D form an integral part of this part of ISO/IEC 9506. Annex E is for information only.

Introduction

ISO/IEC 9506-1 defines the MMS services, a wide variety of services which are useful for the interworking of many types of manufacturing and process control devices when used in open communication systems conforming to the OSI Model (ISO 7498). However, these MMS services by themselves only provide a messaging environment for abstract, generic types of such control devices.

In order to convey device-specific semantics for real manufacturing equipment such as NC machines, robots, programmable controllers, process control systems and so forth, many of the MMS services have optional parameters which have not been detailed in MMS. MMS also specifies that any of these parameters may only be used in the context of a so-called "companion standard" designed for specific devices by the appropriate, recognized standardizing organization.

This part of ISO/IEC 9506 is intended to be such a companion standard to ISO/IEC 9506-1 and ISO/IEC 9506-2. Specifically, it defines the numerical control semantics to the manufacturing message specification. It should be used when numerically controlled manufacturing systems or devices are connected to a communication network conforming to the OSI Model and employing the MMS service and protocol.

Part 4 of ISO/IEC 9506 hereinafter shall be variously referenced as "this standard", "this International Standard", "this part of ISO/IEC 9506", or simply "ISO/IEC 9506-4".

The definitions and specifications of the numerical control specific semantics in this part of ISO/IEC 9506 follow the requirements and guidelines of annex A of ISO/IEC 9506-1.

NOTE 1 This part of ISO/IEC 9506 does not constrain the mapping from real to virtual numerically controlled manufacturing devices, nor does it specify individual implementations or products. This standard also recognizes that safe operation of NC devices is paramount, and that any operations described or defined herein are permissible only if complying with the relevant safety standard(s) and regulations.

Industrial automation systems — Manufacturing Message Specification —

Part 4:

Companion standard for numerical control

1 Scope

This part of ISO/IEC 9506 extends the concepts and principles defined in ISO/IEC 9506-1 and ISO/IEC 9506-2.

This part of ISO/IEC 9506, as a companion standard to Manufacturing Message Specification, describes the semantics of numerically controlled manufacturing devices and equipment. Specifically, this part of ISO/IEC 9506

- a) describes the use of Manufacturing Message Specification services in NC specific applications,
- b) describes the model of an NC device in terms of its application specific functions and how these functions map onto the attributes of a virtual manufacturing device (VMD),
- c) provides the NC specific syntax for those MMS services which are applicable to NC operations, and which also allow companion standard specific parameters,
- d) defines "standardized" names for NC specific objects,
- e) defines NC specific conformance in conjunction with the MMS service and parameter conformance building blocks (CBBs) required.

Furthermore, it should be noted that in addition to this part of ISO/IEC 9506, within ISO and IEC other companion standards have been developed or such work is in progress, in the application areas of robots and robotic systems, programmable control systems, and process control systems.

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 indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO 841:1974, *Numerical control of machines – Axis and motion nomenclature*.

ISO 2806: –¹⁾, *Numerical control of machines – Vocabulary*.

1) To be published. (Revision of ISO 2806:1990.)

ISO/IEC 9506-4: 1992(E)

ISO 7498:1984, *Information processing systems – Open Systems Interconnection – Basic Reference Model*.

ISO/TR 8509:1987, *Information processing systems – Open Systems Interconnection – Service conventions*.

ISO/IEC 8824:1990, *Information technology – Open Systems Interconnection – Specification of Abstract Syntax Notation One (ASN.1)*.

ISO/IEC 9506-1:1990, *Industrial automation systems – Manufacturing Message Specification – Part 1: Service Definition*.

ISO/IEC 9506-2:1991, *Industrial automation systems – Manufacturing– Message Specification – Part 2: Protocol Specification*.

3 Definitions

For the purposes of this part of ISO/IEC 9506 the following definitions apply.

3.1 Alarm

The immediate indication or notification of the occurrence of a severe problem (malfunction) or abnormal condition within the NC system, usually causing an interruption of the operation, which should be resolved externally (such as through operator intervention).

3.2 Device

The whole or part of the machinery or electro-mechanical equipment employed in manufacturing or other processes. Devices may be machine tools, functional parts of machine tools, workpiece handling equipment, or any similar apparatus. In this context, a device shall be visible to an NC-CS user as a separate unit which may be controlled independently by the NC-CS user, such as started and stopped.

3.3 Numerical Control (NC)

Automatic control of a manufacturing or other process performed by one or more device(s), usually involving motion mechanisms, and making use of preprogrammed numeric data, which may be introduced while the operation is in progress.

3.4 Numerical Controller

An electronic control system designed to perform the numerical control (NC) functions of manufacturing or other processes. It usually consists of a computer, file store(s), both analogue and digital input and output interfaces, and one or more operator interface device(s).

3.5 NC-CS (NC Companion Standard)

This part of ISO/IEC 9506.

3.6 NC-CS user

The part of an application process which conceptually invokes the NC Companion Standard. In the context of this part of ISO/IEC 9506, this refers to the external, or "host", processor which communicates with, and exerts some measure of remote control over the NC system.

3.7 NC system

A workstation which consists of a numerical controller and one or more devices.

3.8 Operator Interface Device

An operator interface device provides the means for the NC-CS user to communicate with the NC system operator. Conversely, the same interface may be used by the NC system operator to control the NC system in the LOCAL CONTROL mode.

3.9 Offset

Term which may refer to such general concepts as zero offset, axes value transformation, coordinate transformation, or geometry transformation.

3.10 Axis

Refers to a real, guiding (linear or rotary) axis of a machine, which also reflects one dimension in the work space of the machine.

3.11 Coordinate Axes

See Coordinate System.

3.12 Zero Offset

The relationship (in one dimension) between the origins of a real machine axis and the corresponding part program coordinate.

3.13 Axis Value Transformation

(One dimensional) transformation (sequence of such primitives as translation, mirroring, and scaling) of a point on a real machine axis in terms of the corresponding part coordinate.

3.14 Coordinate System

Defined in ISO 841

NOTE - The Coordinate System may also be uniquely defined by three orthogonal coordinate axes. Always right-handed

3.15 Actual Coordinate System

The actual coordinate system used by the NC.

3.16 Coordinate Transformation

Expression of the spatial relationship (location and orientation) between two independent coordinate systems.

3.17 Geometry Transformation

The application of a sequence of primitive transformations (translation, rotation, scaling, mirroring) to a point in space in terms of the actual coordinate system.

In addition to the definitions in this clause, the definitions of clause 3 of ISO/IEC 9506, part 1 apply, which in addition to MMS-specific definitions lists a number of terms defined in ISO 7498, the OSI Basic Reference Model, in ISO/TR 8509, the OSI Service Conventions, and in ISO 8824, the Abstract Syntax Notation One Specification. All of these definitions apply to this standard by reference.

NC specific definitions may also be found in ISO 2806.

4 Abbreviations

The following abbreviations are used in this part of ISO/IEC 9506:

ASN.1	Abstract Syntax Notation One
CBB	Conformance Building Block
Cnf	confirm
CS	Companion Standard, often used with NC: NC-CS
Ind	indication
M	mandatory parameter
MMS	Manufacturing Message Specification
NC	Numerical Control or Numerical Controller
OSI	Open Systems Interconnection
PICS	Protocol Implementation Conformance Statement Proforma
Req	request
Rsp	response
VMD	Virtual Manufacturing Device

5 NC Application Description

This chapter describes the model and functionality of an NC system in a manufacturing communication environment.

5.1 NC Specific Model

An NC system is a workstation, where one or more devices are controlled by a Numerical Controller (NC). Such devices may consist of any mechanical or electro-mechanical equipment depending on their individual application. An NC system may be connected to other stations through one or more communication channel(s).

In a factory, several NC-CS users may communicate with a single NC system, as shown in Figure 1.

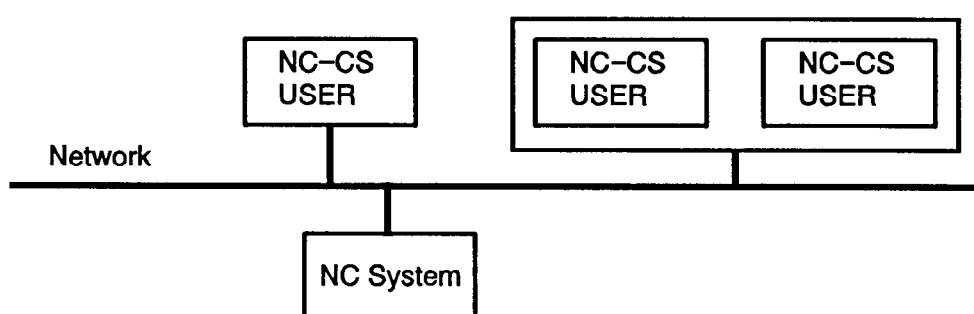


Figure 1: NC System in the Manufacturing Environment

An NC system may be described by models of its component parts. Figure 2 models the component parts. These component parts may describe aspects of the numerical controller, the device, or both.

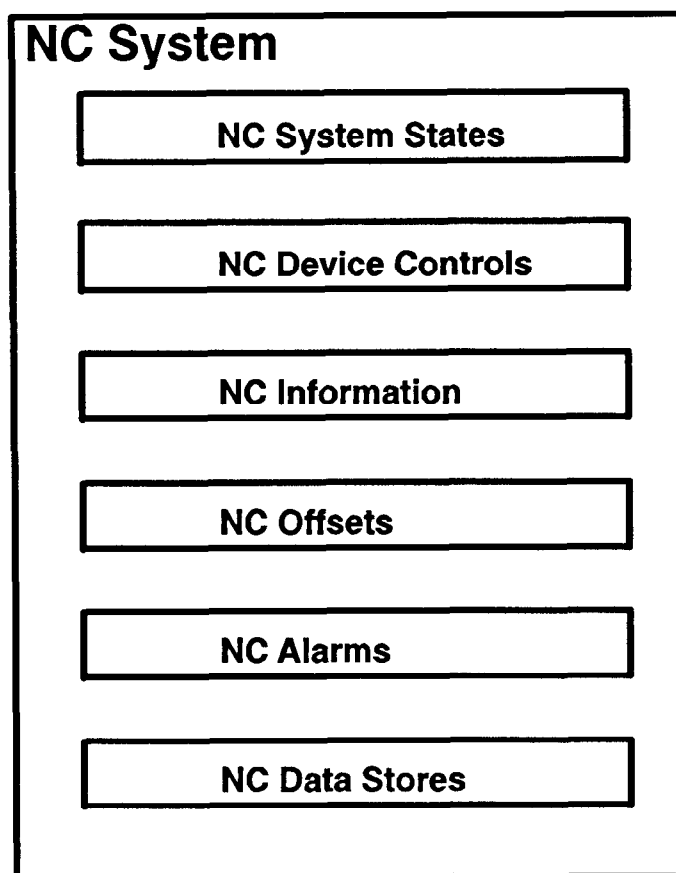


Figure 2: NC System Components

The models in this section describe the elements of the NC application area from a communication point of view. They include the following models:

- the NC system state model,
- the NC device control model,
- the NC information model,
- the NC offset model,
- the NC alarm model,
- the NC data store model.

5.1.1 NC System State Model

The NC System State Model describes the global state of an NC system, as visible to an NC-CS user. Also, it represents a mechanism which allows coordination between local users and NC-CS users (remote users), as well as the coordination between multiple NC-CS users.

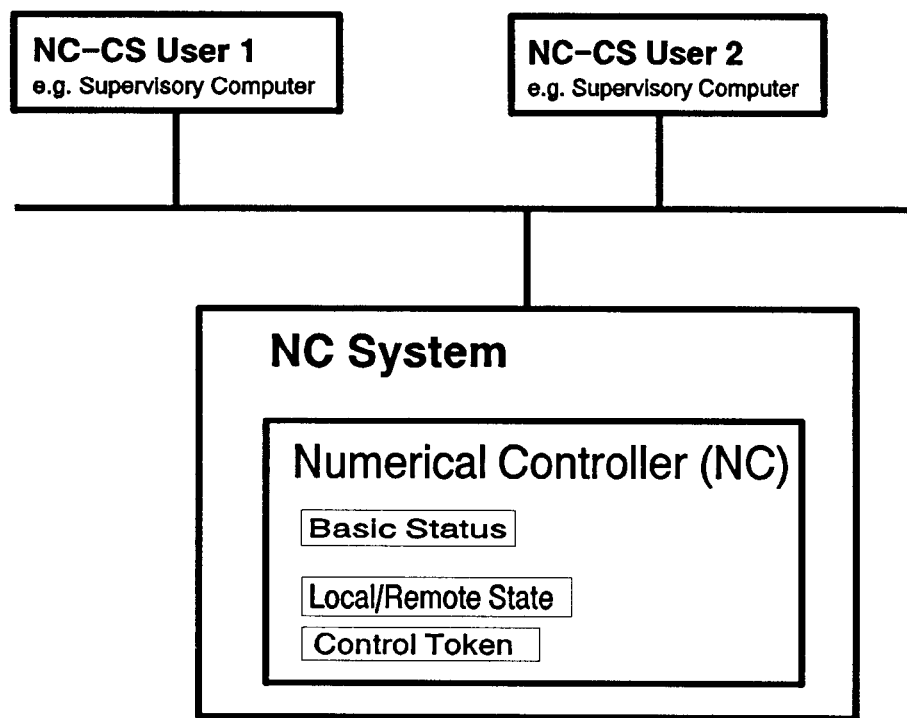


Figure 3: NC System State Model

5.1.1.1 Basic Status

The Basic Status describes the overall working condition of the NC system.

5.1.1.1.1 State Descriptions

- a) NORMAL PRODUCTION - The NC system is fully operable.
- b) LOCAL-ACTION REQUIRED - A component of the NC system has failed or is not operable

5.1.1.2 Local/Remote State

In the factory environment an NC system may be controlled by an NC-CS user (remotely) or locally, as shown in figure 4.

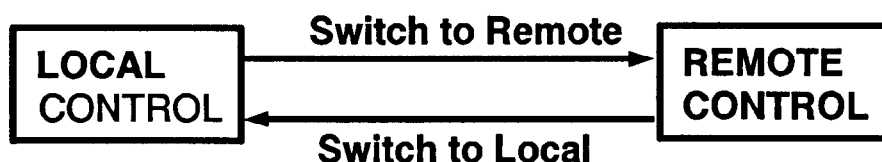


Figure 4: Local/Remote State

5.1.1.2.1 State Descriptions

- a) **LOCAL CONTROL** - In this state, the NC operation is under control of a local user. For a transition to the REMOTE CONTROL mode, local action by the operator is required. ISO/IEC 9506 part 4 does not include any states or operations under LOCAL CONTROL.

NOTE - Even in LOCAL CONTROL some remote operations may be possible. For example, an NC-CS user may inform the NC system about the next part to be made. The choice of remote operations which may be allowed in LOCAL CONTROL is implementation dependent, and should be closely related to security. This choice may even be changed as a result of operator decisions.

- b) **REMOTE CONTROL** - In this state, the NC system is controlled by a remote user. Any operator intervention which modifies the state of the controlling process, or which results in a safety violation shall force the transition to LOCAL CONTROL.

5.1.1.2.2 State Transitions

- a) **Switch to Remote** - This transition shall only be the result of local action and shall enable remote control of the NC system. While this transition may also depend on the state of the Controlling Process, the exact conditions for this shall be at the discretion of the system implementer.
- b) **Switch to Local** - This transition shall change the NC system from REMOTE CONTROL to LOCAL CONTROL. It may take place during any of the Device states and Controlling Process states, and the resultant states of the Device(s) and/or Controlling Process(es) shall be implementation dependent.

5.1.1.3 Control Token

An NC system may be controlled by several NC-CS users. Such a multiple client configuration requires a mechanism for taking and relinquishing control of the NC system. Without this capability, there is no means for preventing two or more NC-CS users from attempting to control the NC system simultaneously.

5.1.2 The NC Device Control Model

The NC Device Control Model may be viewed as a number of interacting components, such as one or more Devices, a Numerical Controller, an Operator Interface Device, and one or more Controlling Processes, as illustrated in figures 5 through 7.

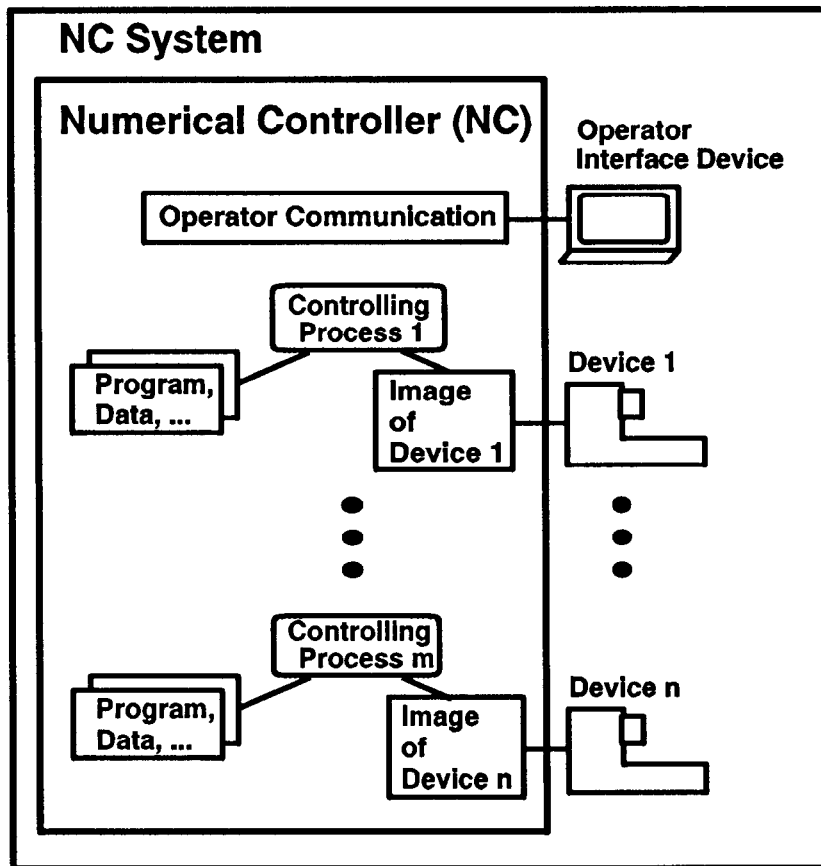


Figure 5: NC Device Control Model

5.1.2.1 Devices

Devices represent those mechanical or electro-mechanical parts of an NC system which are visible to an NC-CS user as a separate entity which may be controlled separately, such as started or stopped.

NOTE - The manner in which devices are delineated within a system shall be arbitrary. While one implementation may consider as devices such overall entities as a milling machine or tool warehouse, another one may subdivide a system into smaller units, such as spindle, rotary table, or tool changer.

5.1.2.2 Image of Device

The Numerical Controller contains an image of every Device. The image fits with some mechanics and some resources of the Device configured when the NC System is set up. The Device Image includes for example the actual state of the Device or the number of axis.

5.1.2.3 Controlling Process

Controlling Processes represent the execution of the user programs, which are linked to part programs and data such as tool corrections and fixture offsets. They may also be linked to execution information, such as the current execution state, the line of program currently being processed, and so on. Controlling Processes are linked to devices, and they may be created dynamically by the NC-CS user.

5.1.2.4 Operator Interface Device

An operator interface device provides the means for the NC-CS user to communicate with the NC system operator. Physically, the same interface may also be used by the NC system operator to control locally the NC system.

5.1.2.5 Single Device NC System

The general model of an NC system may consist of one or more devices and one or more Controlling Processes. A simple numerical controller, however, may support only a single device and a single Controlling Process.

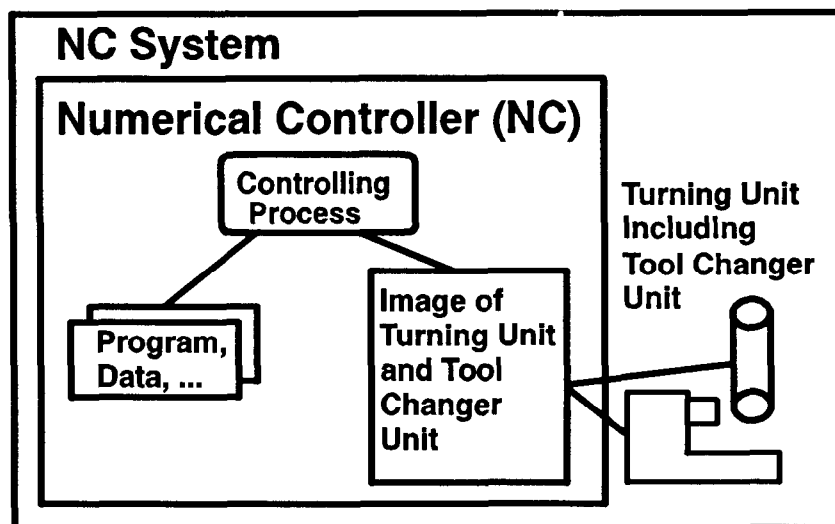


Figure 6: NC Device Control Model (Single Device)

For example figure 6 shows an NC system consisting of a single device and a single Controlling Process. The device image represents a turning unit and a tool changer. The Controlling Process links programs and data to the device.

5.1.2.6 Multiple Device NC System

Other numerical controllers may be able to control several, usually interacting devices. An example of such a system is shown in figure 7, which models three separate devices, two turning units, and a tool warehouse. The NC-CS user may act on each of these devices independently.

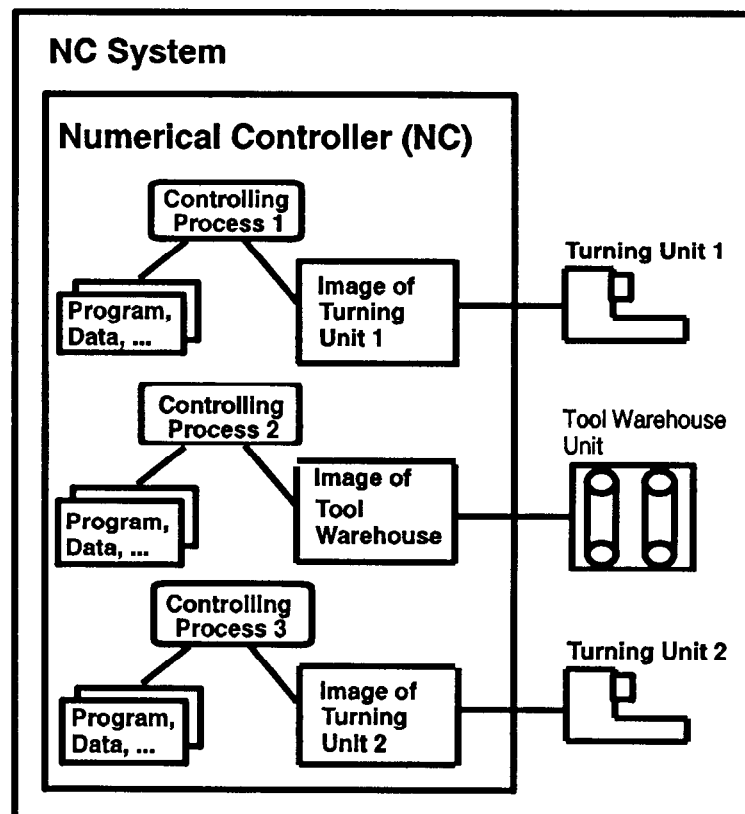


Figure 7: NC Device Control Model (3 independent Devices)

The operation of the NC device is described by the Device State Model and the Controlling Process State Model.

5.1.2.7 The Device State Model

The Device State Model describes the state of the device. The image of the device includes the Device State.

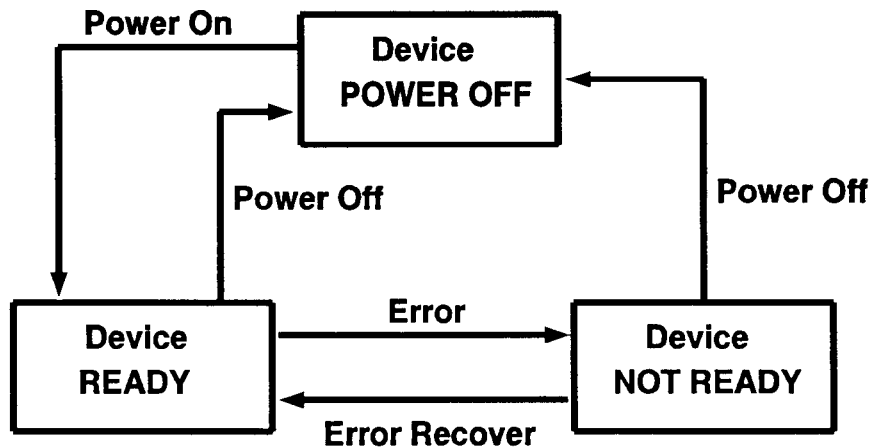


Figure 8: NC Device State Model

5.1.2.7.1 State Descriptions

- a) **POWER OFF** - In this state, all device functions are inactive except for those pertaining to safety (e.g. brakes, clamps). No power should be applied to the device.
- b) **READY** - In this state, power is applied to the device, that is, axes drives, spindle drives, hydraulic pumps, etc.
- c) **NOT READY** - This state may be caused by an error or a stop condition. If caused by an error condition, only an action which acknowledges and removes the error condition may cause an exit from this state. It is assumed that the NC provides sufficient information for proper diagnosis.

5.1.2.7.2 State Transitions

- a) **Power On** - This is associated with energizing the device, and represents the transition from **POWER OFF** to **READY**.
- b) **Power Off** - This transition puts the device into the **POWER OFF** state, which shall halt all operations of the device.
- c) **Error** - This transition puts the device from the **READY** state into the **NOT READY** state.
- d) **Error Recover** - This transition returns the device to the **READY** state.

NOTE - The above device state model is simplified. In reality, such transitions as **Power On** and **Power Off** may be rather complex procedures.

5.1.2.8 The Controlling Process State Model

The Controlling Process state model provides the user with the information about the execution state of the user program. A Controlling Process may control one or more devices. For an NC-CS user it is necessary to be aware of the program, and the way the program acts on the devices.

Therefore, the Controlling Process state model describes the execution states of the user program, such as start, stop, and so on.

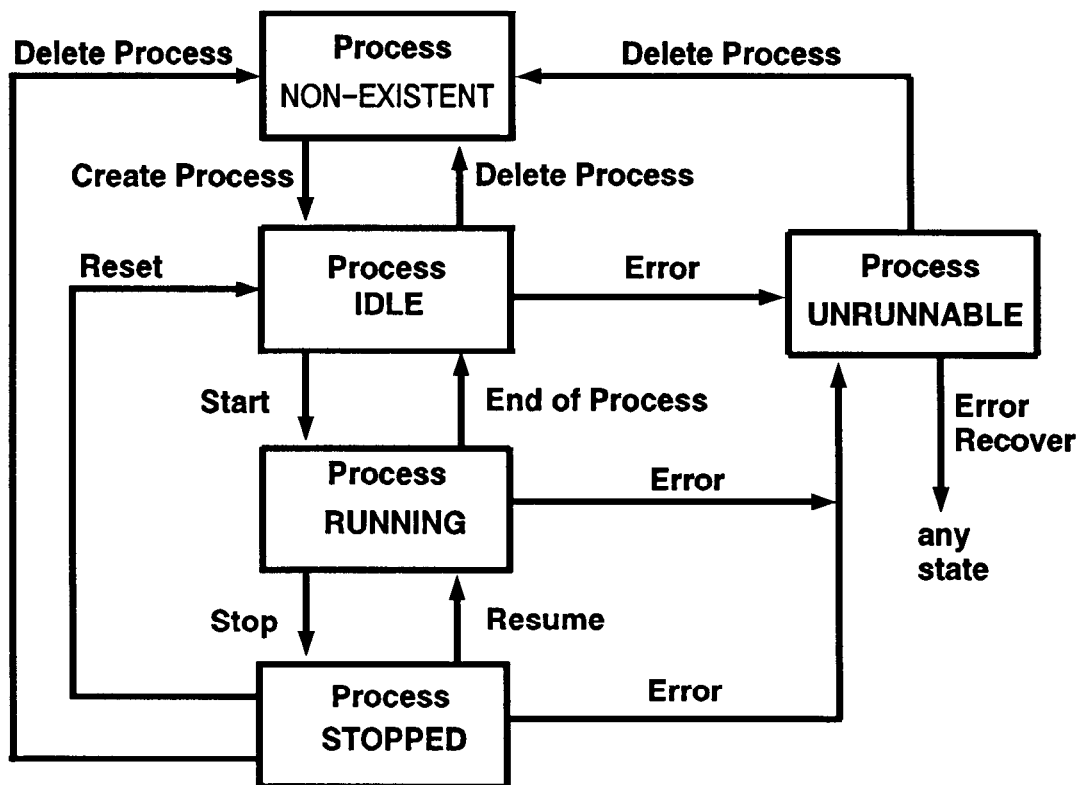


Figure 9: NC Controlling Process State Model

The Controlling Process may be created dynamically by the NC-CS user. Such creation usually entails the binding of a device to the Controlling Process, and the selection of user programs and data. To operate any device, at least one Controlling Process is needed. The binding of some Controlling Processes and their devices may be static.

5.1.2.8.1 State Descriptions

- a) NON-EXISTENT - This is the state of a Controlling Process prior to its creation.
- b) UNRUNNABLE - This state is the result of an error. Recovery from this state requires the explicit action of an operator.
- c) IDLE - In this state the Controlling Process is ready to execute, that is, all required initialization have been completed.
- d) RUNNING - In this state the Controlling Process is fully operational and executing user program commands.
- e) STOPPED - This state may be the result of a stop command, or any other stop condition. From this state, after stopping, the user program may resume execution normally.

5.1.2.8.2 State Transitions

State transitions may be initiated from a number of sources, such as the NC-CS user, the operator, the user program, or the Device. Also, it is presumed that the manufacturer may define certain conditions which will inhibit these transitions, while providing sufficient information to properly diagnose the situation.

- a) Create Process - This transition creates a Controlling Process by linking user programs, data, and devices.
- b) Delete Process - This transition deletes a Controlling Process, by deleting all such links between programs, data, and devices.
- c) Start - This transition starts the execution of the Controlling Process, assuming that all prerequisites have been fulfilled.
- d) End of Process - This transition completes the execution of the Controlling Process. This transition usually corresponds to the end of the program.
- e) Stop - This transition stops execution of the program. It may be caused by a stop command from the NC-CS user, and it puts the Controlling Process into the STOPPED state. Execution may be restarted by the Resume transition.
- f) Resume - This transition restores the RUNNING state.
- g) Error - This transition shall suspend the Controlling Process and cause it to be UNRUNNABLE. The cause of the error shall be removed before the process may be recovered from this state.
- h) Error Recover - Removing the cause of a previous error by local or remote action shall result in this transition from the UNRUNNABLE state to any state.
- i) Reset - This transition puts the Controlling Process into the IDLE state.

5.1.2.8.3 Correlation To The Device State

Since the states of a device and its associated Controlling Process states are interdependent they will influence each other. For instance, when a device is switched off or enters the NOT READY state, then there should be a Controlling Process state transition.

The following table describes the correlation between device states and Controlling Process states.

Table 1 - Correlation between Device States and Controlling Process States

Device State	Possible Controlling Process States
POWER OFF	NON-EXISTENT, UNRUNNABLE, IDLE
READY	any state
NOT READY	NON-EXISTENT, UNRUNNABLE, STOPPED, IDLE

5.1.3 NC Information Model

Basic Status, Device Status, and the Controlling Process Status provide the NC-CS user with general information about the state of the NC system. However, to provide the more detailed information about the NC system required for remote control by the NC-CS user, this part of ISO/IEC 9506 includes the additional types of information described here by the NC Information Model.

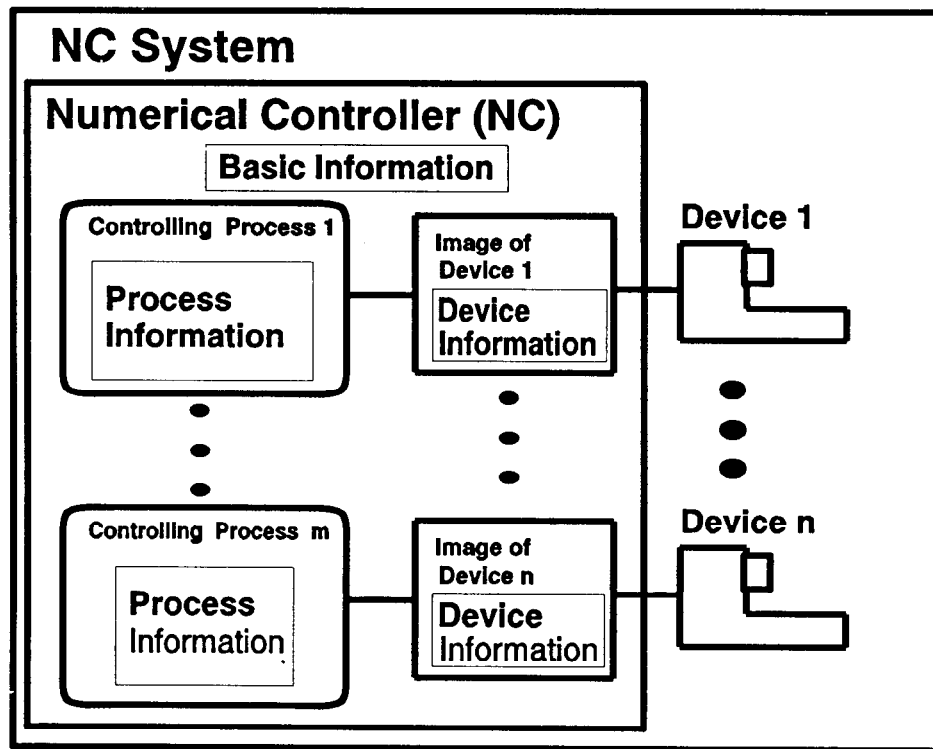


Figure 10: NC Information Model

5.1.3.1 Basic Information

This encompasses information about the Numerical Controller, such as controller voltage or controller error information.

5.1.3.2 Device Information

This encompasses information about the device, such as hydraulic pressure or device voltage.

5.1.3.3 Process Information

This encompasses information about the Controlling Process, e.g. the currently executing program block.

5.1.4 Offset Model

An important part of the operation of an NC system is the capability to handle axis transformation, zero offsets, and other coordinate transformations. In stand-alone, conventional NC systems most of this functionality is handled by the machine operator, as well as by user programs. The purpose of the Offset Model is to provide the NC-CS user with these functions.

The Offset Model shall provide the NC-CS user with all the geometric transformation capabilities required for remote control. Offsets represent the transformations of the location descriptions of points which are usually associated with the machined part or workpiece. The Offset Model distinguishes two types of such transformations, one dimensional and three dimensional, that is, in terms of one axis, or in terms of Cartesian coordinates. Both types further distinguish between (1) transformation of the reference origin, and (2) transformations of axis values or geometry. For better visualization of the Offset Model refer to figures 11 through 15.

For tutorial information on this, see (informative) Annex E.

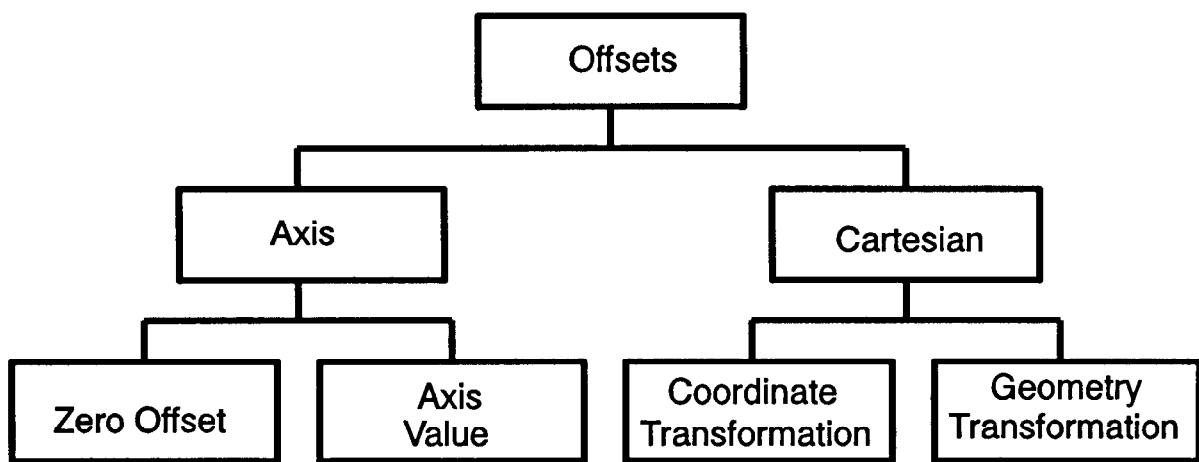


Figure 11: Offset Classification Hierarchy

5.1.4.1 Zero Offset

A zero offset shifts the virtual origin (zero point) of a physical machine axis (see figure 12). This virtual origin is the reference for the part program.

Zero offsets equally apply to linear and rotary axes.

virtual origin = real origin + zero offset 1 + zero offset 2 ... + zero offset n

The number of zero offsets supported in an NC is implementation specific, and shall be stated by the appropriate PICS entry.

For rotary axes the permissible range shall be specified (e.g. 0..360, -180..180, ...).

Axis with virtual origin

- Translation T(1)
- Scaling S(2)
- Mirroring M()
- Translation T(2)

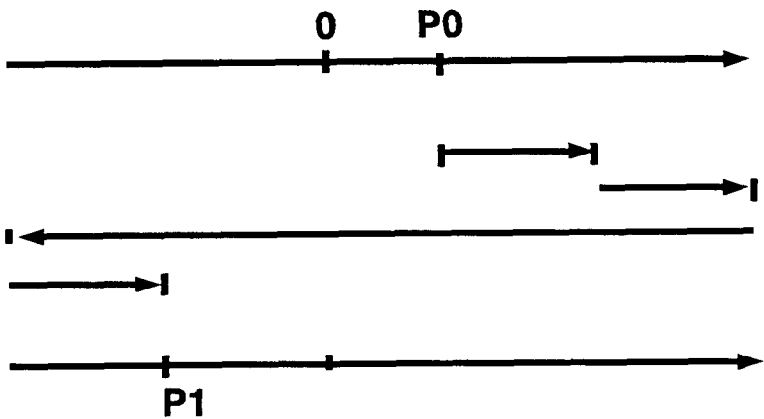


Figure 13: Sequence of Primitive Axis Value Transformations

5.1.4.3 Coordinate Transformations

A coordinate transformation defines the position and rotation of one coordinate system in terms of another coordinate system. Such transformation should consist of a rotation (Euler angles) and a vector, in accordance with the conventions of ISO 841.

Objects defined in this part of ISO/IEC 9506 which may represent pallets, fixtures, parts, or partitions may have offset attributes which refer to coordinate transformations. An example of coordinate transformations is shown in figure 14.

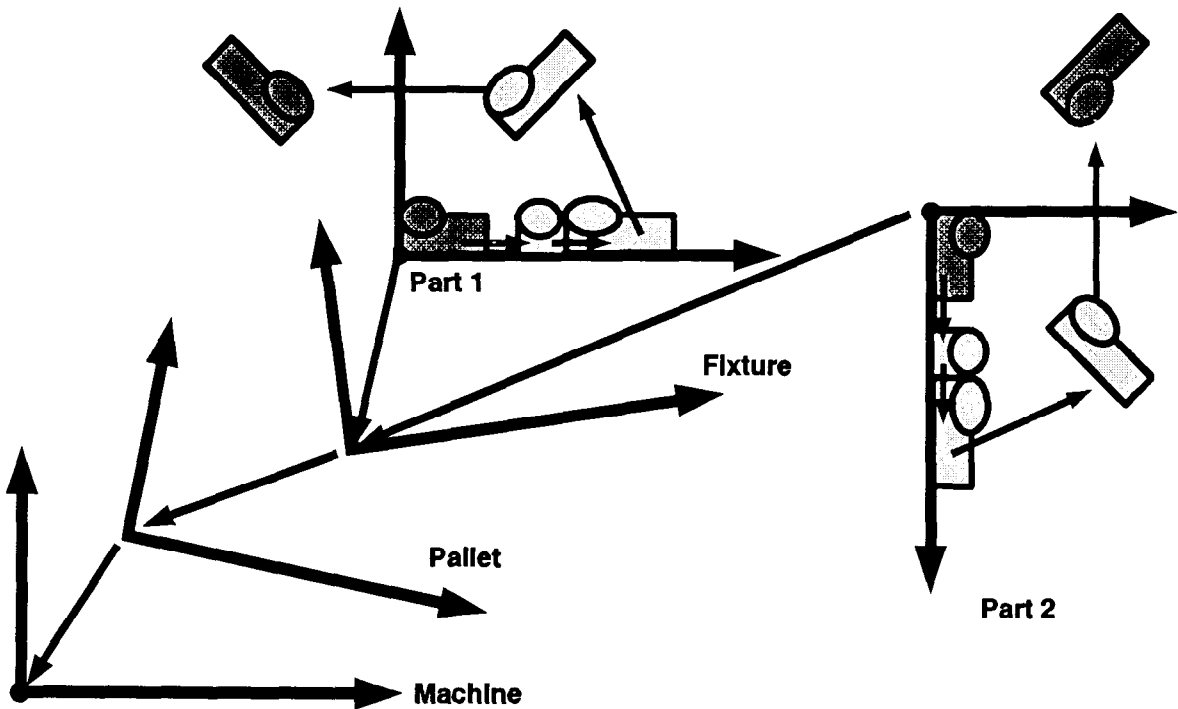


Figure 14: Coordinate Transformations

User programs are written for workpieces or parts in terms of their own specific coordinate systems. To execute correctly, sufficient information about such a coordinate system is required within the NC. This information shall be contained in an object known as the "Active Program Coordinate System".

5.1.4.4 Geometry Transformation

Geometry Transformation defines the transformation applied to the coordinates of a point in three dimensional space, to change its reference from the original coordinate system, usually the "program coordinate system", to some target coordinate system, usually the coordinate system of the machining environment. The effect of geometry transformations is the change of location, orientation, or geometry of the workpiece produced by the part program.

The following defines the primitives used in geometry transformations. The original coordinate system is assumed to be in terms of the coordinates x, y, and z:

Translation:

Translation of point P0 with vector v
 primitive T(v): $P1 = T(v) * P0$ (matrix notation)

Scaling:

Scaling of the x,y,z components of P0 with scale factors s1, s2, s3.
 primitive S(s1,s2,s3): $P1 = S(s1,s2,s3) * P0$ (matrix notation)

Rotation:

Sequence of three rotations (Rotation defined by ISO 841), defined in the following order:

1. Rotation about x axis with angle a
2. Rotation about y axis with angle b
3. Rotation about z axis with angle c

primitive R(a,b,c): $P1 = R(a,b,c) * P0$ (matrix notation)

Mirroring:

M (xy, yz, xz)

For xy, yz and xz only the two values 0 and 1 are valid.

0 means no mirroring about the specified plane.

1 means mirroring about the specified plane.

Examples:

mirroring about the xy plane:	M(1,0,0)
mirroring about the yz plane:	M(0,1,0)
mirroring about the xz plane:	M(0,0,1)
mirroring about the xy and yz plane:	M(1,1,0)

Mirroring by itself is commutative. Otherwise, any sequence of primitive operations is order dependent. Following is a commonly used notation which reflects the order (see figure 15):

$$P1 = T(1,0,0) * R(10,20,30) * M(1,0,0) * P0$$

^{^first primitive}
^{^second primitive}
^{^third primitive}

NOTE 1 - When using the mirroring primitive, the shape of a workpiece remains the same. For example, if the program describes a cylinder, then after a mirroring operation the cylinder remains a cylinder. When using the scaling primitive with negative scale factors, the shape changes. Thus, the cylinder program, after scaling with the scale factor -1, will produce a hole in the work piece.

NOTE 2 - This standard does not limit the number of primitive transformations, which shall be implementation specific.

NOTE 3 - Complex geometric transformations may be expressed by a sequence of primitives.

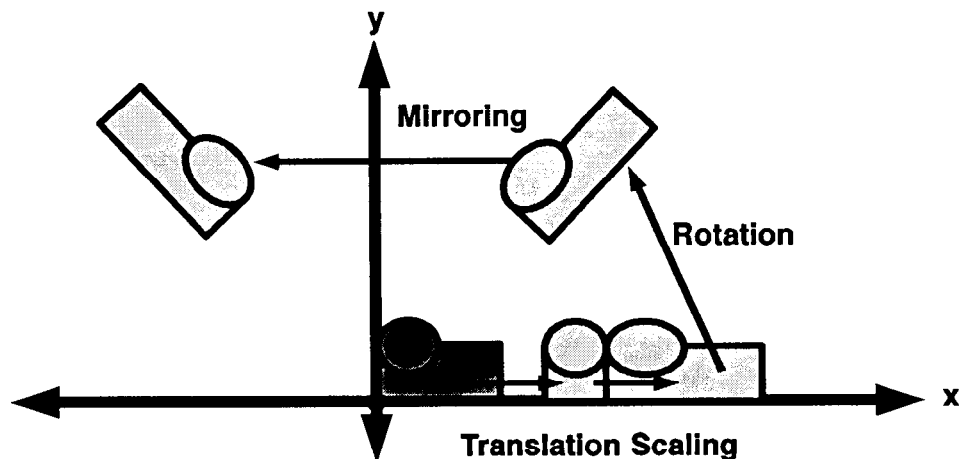


Figure 15: Sequence of Primitive Geometry Transformations

5.1.5 NC Alarm Model

In order to properly perform its task within a manufacturing environment, an NC shall have the ability to signal faults to an NC-CS user.

NOTE - Such faults may be measuring system faults, control unit temperature problems, control unit battery problems, and so on. Typically, within an NC there exists a list of hundred or more fault conditions, each of which may cause an alarm.

The monitoring and signaling of alarms is represented by the NC Alarm Model. If an Alarm occurs and alarm monitoring is enabled, an NC Alarm Notification is sent to the NC-CS user. The NC Alarm Notification includes the following information:

NC Alarm Name - Every alarm which is to be signaled to an NC-CS user is uniquely identified by its NC Alarm Name

NCAlarmState - This parameter represents the current state of Alarm Monitoring (DISABLED, IDLE, ACTIVE).

NC Alarm State = DISABLED indicates that Alarm monitoring is disabled. When alarm monitoring is disabled, a NC Alarm Notification may be sent to the NC-CS user (NC Alarm State = DISABLED) to inform the NC-CS user.

NC Alarm State = IDLE indicates that the monitored alarm is not active or the alarm has ceased to exist.

NC Alarm State = ACTIVE indicates that the monitored alarm is active.

NC Alarm Severity - This parameter represents some measure of severity for NC-CS user receiving the NC Alarm Notification.

NC Alarm Time - This parameter contains the time (date and time or a time sequence identifier) at which the transition in the NC alarm state was detected.

NC Alarm Acknowledgment Rule - This parameter indicates how the NC-CS user shall acknowledge the receipt of the NC Alarm Notification.

NC Alarm Acknowledgement Rule = NONE indicates that there are no acknowledgements required.

NC Alarm Acknowledgement Rule = ACK-ACTIVE - Acknowledgement is required for Alarm Notifications indicating the NC Alarm State ACTIVE. For the NC Alarm State IDLE, acknowledgements are allowed but not required.

NC Alarm Acknowledgement Rule = ACK-ALL - Acknowledgements are required for Alarm Notifications indicating the ACTIVE or IDLE Alarm State.

NOTE - The Alarm Acknowledgement Rule may be different for every alarm. For NC Systems not supporting the acknowledgement of alarms, this parameter shall have the value NONE.

5.1.6 NC Data Store Model

Most numerical controllers provide data storage, as well as the capability of remote selection, inspection, and/or retrieval of any data required for NC operations. Such data typically consist of user programs and operational parameters, such as tooling, fixturing, process, and other parameters. They are usually generated remotely, and transferred to the NC either via a communication network, or by local means. The Data Store Model defines the organization of the data store, and the manner in which the data may be viewed.

While this part of ISO/IEC 9506 does not specify any particular data store hierarchy, it does provide for multiple data stores. Such data stores may be within or external to the numerical controller. External store(s) are included in this model so that they may be visible to the NC-CS user.

NOTE - ISO/IEC 9506-4 only defines the model of a logical data storage system from the viewpoint of an NC-CS user, and the mapping of such model to a real data store is outside the scope of this part of ISO/IEC 9506.

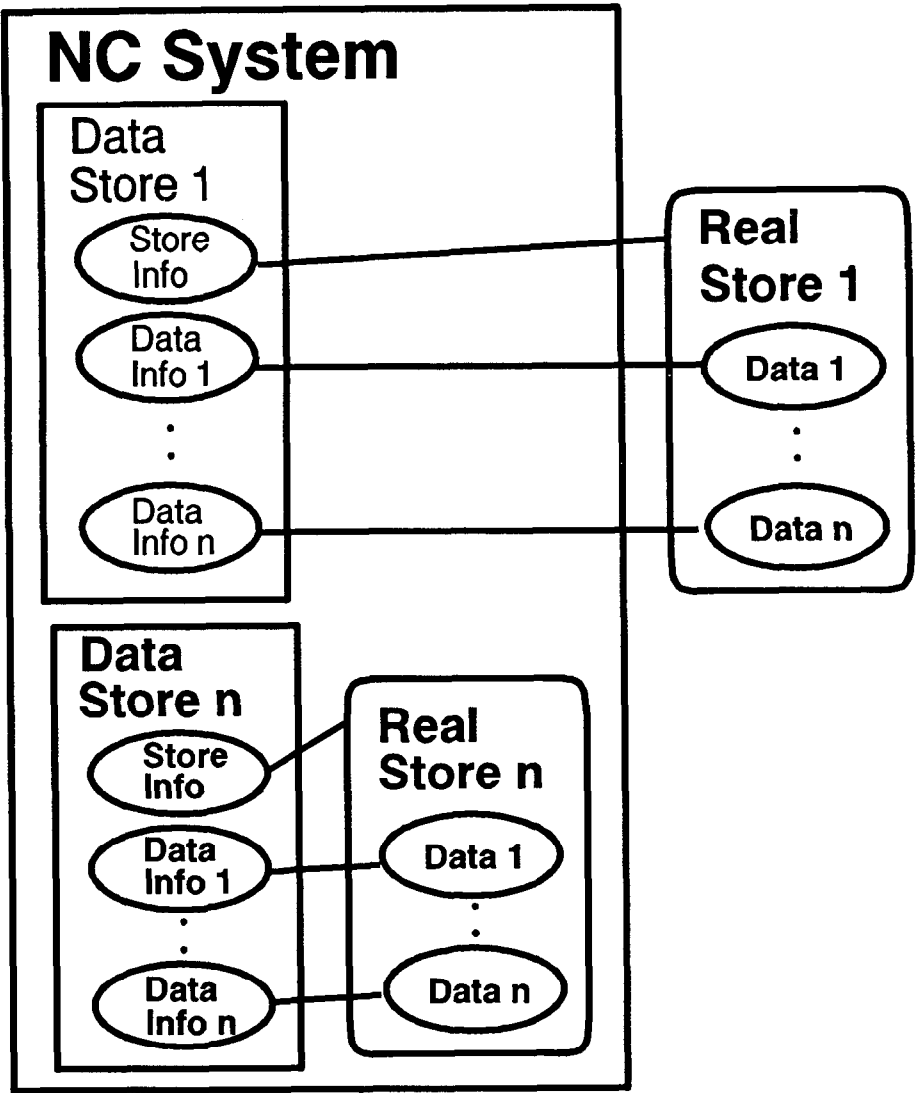


Figure 16: NC Data Store Model

5.2 NC Specific Functions

This part of ISO/IEC 9506 does not define any functional classes of NC systems. Instead, it defines the following, general NC specific functions, any of which may or may not be supported by a particular NC system. The mapping of these functions to MMS is defined in Chapter 6.

5.2.1 Data Transfer

Numerical controllers conforming to this part of ISO/IEC 9506 are able to transfer data from the NC-CS user to the NC system and back. There are 3 different ways to transfer data to and from an NC system:

- a) **Block Transfer to Datastore** - The transfer of "bulk" data between the NC-CS user and the NC system (Download and Upload).
- b) **Block Transfer Dynamic** - The NC system executes the data during transfer with minimal buffering (Dynamic Download).
- c) **Variable Transfer** - In this way data are viewed as variables and the data transfer is done by reading or writing these variables.

The NC-CS user may also direct the NC system to load or store data from a remote fileserver.

Some data, such as tooling data, may be handled either as data blocks or as variables, or both, depending on the complexity of a particular application. Simple numerical controllers may handle tooling data using block transfer, and more complex systems may use either block transfer or variable transfer, or both. Variable data structures to handle a variety of applications are defined in the normative Annexes to ISO/IEC 9506-4.

5.2.2 NC Data Store Management

The NC Data Store Model provides to the NC-CS user the information which addresses the following data management capabilities:

- a) **Storage Organization** - the names of the NC-CS user programs in the data store(s).
- b) **Version Control** - the origin and/or version attribute of NC-CS user programs in the data store(s).
- c) **Memory Usage** - the amount of data store space allocated to NC-CS user programs.
- d) **Memory Location** - the kind of storage medium represented by the data store.
- e) **Data Type** - the format of NC-CS user programs in the data store.
- f) **Protection Schema** - the integrity, or accessibility of the data in the data store.
- g) **Deletion of data store entries.**

5.2.3 Process management

For remote control of the NC system, the following functions which relate to program control are required:

- a) **Select a machine program for execution.**
- b) **Start the execution of a Controlling Process. (Remote cycle start)**
- c) **Pause and resume a Controlling Process.**
- d) **Stop the execution of a Controlling Process.**

5.2.4 Interaction with an operator

The operator interface device provides the means for the NC-CS user to communicate with the NC system operator.

5.2.5 Alarm Processing

In order to signal faults to the NC-CS user, NC systems should have the capability to:

- a) mask/unmask alarms
- b) signal alarms to the NC-CS user
- c) receive alarm acknowledgments
- d) provide summary information of active alarms

6 NC Specific Context Mapping

This clause relates the model of the NC system which has been defined in clause 5, to the abstract model of the Virtual Manufacturing Device (VMD) defined in MMS. For the purpose of inter-operability the uniform application of the abstract concepts in MMS to real systems is essential. While this part of ISO/IEC 9506 may not anticipate every variant of NC System existing or yet to be developed, it provides general guidelines for associating real systems with the abstract model, which should be applicable to any conforming NC system.

6.1 Mapping Of The NC Specific Model To The VMD Object

This clause shows how different elements of NC systems relate to the MMS abstract concepts of VMD, Domains, Program Invocations, and Variables etc. It is equally applicable to (simple) NC systems which map to a single VMD, as well as those which map to multiple VMDs. The distinction depends largely upon the application.

6.1.1 Mapping Of The NC System To The VMD Model

This clause defines how the NC specific models of clause 5.1 map to the abstract objects defined in part 1 of this International Standard.

Figure 17 shows the major components of an NC VMD and their relation to each other. An NC VMD may include variables, domains, and one or more Program Invocations.

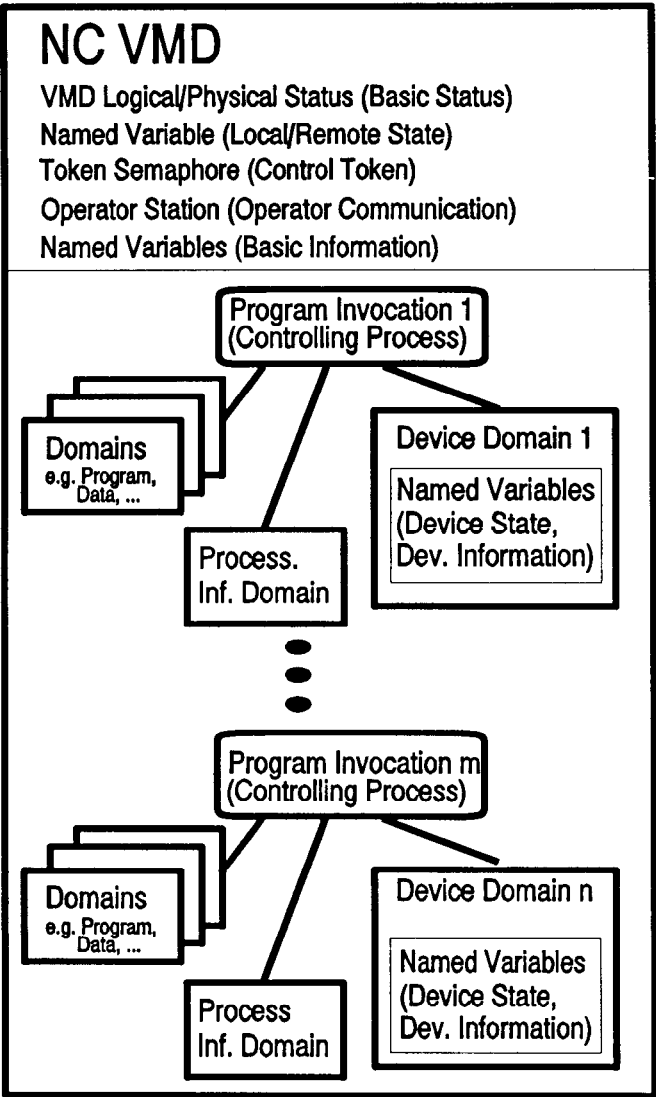


Figure 17: NC VMD Model

6.1.1.1 Basic Status

The Basic Status maps to the VMD Logical Status and to the VMD Physical Status.

6.1.1.2 Local/Remote State

The Local/Remote State maps to a VMD specific Named Variable Object with the name N_REMOTE.

6.1.1.3 Control Token

The control token maps to a Token Semaphore Object of the name N_CONTROL.

6.1.1.4 Device Image

A device image maps to a predefined domain object (Device Domain) with domain specific Named Variable objects which represent the physical resources and the state of the Device.

The device state maps to a Device Domain specific Named Variable N_DEV_STATE.

6.1.1.5 User Program

A user program is modeled as a domain object and is linked to a controlling process.

6.1.1.6 User Data

User data are modeled as domain objects and may be linked to a controlling process

6.1.1.7 Controlling Process

The Controlling Process is modeled as a Program Invocation object linked to the domain objects which are germane to the controlling process. The Controlling Process state maps to the Program Invocation state.

6.1.1.8 NC Information

The Basic Information, the Device Information and the Controlling Process Information are mapped to variables. The Basic Information maps to VMD specific Named Variables, the Device Information maps to device domain specific variables. The Process Information maps to domain specific variables. The Process Information domain which contains these variables is locally created with creation of the assigned Program Invocation and shall include the name of the Program Invocation.

Instead of accessing different variables separately, the NC-CS user may define Named Variable List objects consisting of the desired variables. This part of ISO/IEC 9506 provides Variable List objects for the Basic Information (Basic Information List), for the Device Information (Device Information List) and for the Process Information (Process Information List). The use of these objects, however, is optional.

6.1.1.9 NC Alarm Model

The NC Alarm Model maps to the MMS Event Model.

6.1.1.10 NC Data Store

The following object classes are used to provide the functionality of the data store as required by the NC Data Store Model:

- a) NC Domain Object Class:
 - NC Data Store Domain
(N_STORE_...)

- b) Named Variable Object Class:

ISO/IEC 9506-4: 1992(E)

- Data Store Information
(N_StoreType,
N_MaxStoreEntries,
N_MaxStoreCapacity,
N_CurrentEntries,
N_RemainingEntries,
N_CurrentCapacityUsed,
N_RemainingCapacity)
- Data Store Entry

c) Named Variable Type Object Class:

- Data Store Entry Type
(N_DataStoreEntry)

d) Named Variable List Object Class:

- Data Store Entry List
(N_EntryList)

6.1.1.11 Operator Interface Device

The Operator Interface Device maps to the Operator Station object in MMS. Several Operator Station objects may exist in an NC system.

6.1.2 NC VMD

6.1.2.1 NC VMD Object

Object: NC VMD

All MMS defined VMD attributes

Additional Detail - There are no additional attributes defined in this standard.

6.1.2.2 VMD Logical Status and VMD Physical Status

The VMD Logical and Physical Status produce the Basic Status according to the following table:

Table 2 - Status Coherence

Basic Status	VMD Logical Status	VMD Physical Status
NORMAL PRODUCTION	State Changes Allowed	Operational
LOCAL ACTION	any	Needs Commissioning, Inoperable REQUIRED Partially Operational

6.2 Definition Of NC Specific Objects Which Map To Domains

6.2.1 NC Domain Object

Object: NC Domain

All MMS defined Domain attributes

Additional Detail - There are no additional attributes defined in this standard.

6.2.2 Device Domain

The Device Domain represents the physical resources of a Device. It is a predefined domain which may be set up by the manufacturer, and which provides a naming space for domain specific Named Variables, such as Device State, Number of Axes, and so on.

6.2.3 Process Information Domain

The Process Information Domain contains the Process Information of the corresponding Controlling Process. This domain is locally created with creation of a Controlling Process and exists only during the lifetime of the same, that is, it is locally deleted with deletion of the Controlling Process.

The Attribute List of Domain References of the associated Program Invocation contains a reference to the Process Information Domain.

6.2.4 Program Domain

User programs usually contain a sequence of part execution steps and are coded in a high level language (such as ISO 6983). Such programs are modelled as domains, which provides the means to upload, download or delete user programs from the NC store, and enables an NC-CS user to assemble several user programs into one program invocation.

6.2.5 Data Domain

Data used by user programs may be modelled as domains or variables in domains, for example:

- Tool Data domain: manufacturer defined tooling information.
- Means of Production Data domain: manufacturer defined means of production information, such as fixture information.

Like program domains these objects may be uploaded, downloaded, or deleted from the NC store, and may be assembled, together with user programs, into a program invocation.

6.2.6 NC Data Store Domain

An NC Data Store domain object shall be created for each logical data store in the NC VMD. Such a domain shall contain certain domain specific objects, namely those which define the NC data store, and those which define domains assigned to the data store. The name of an NC Data Store domain shall begin with "N_STORE_...". The creation of an NC Data Store domain is a local matter. This standard does not map a logical data store to a physical data storage device.

The NC Data Store Domain list of subordinate objects includes the following:

- a) Data Store Information - corresponds to the "Store Info" element of the NC Data Store model (see figure 18).
- b) Data Store Entry Type - defines the content of "Data Info" element of the model.
- c) Data Store Entries - corresponds to the "Data Info" element of the model.
- d) Data Store Entry List - provides a list of Data Info element names.

The relationship between these objects and the NC Data Store model is shown in figure 18.

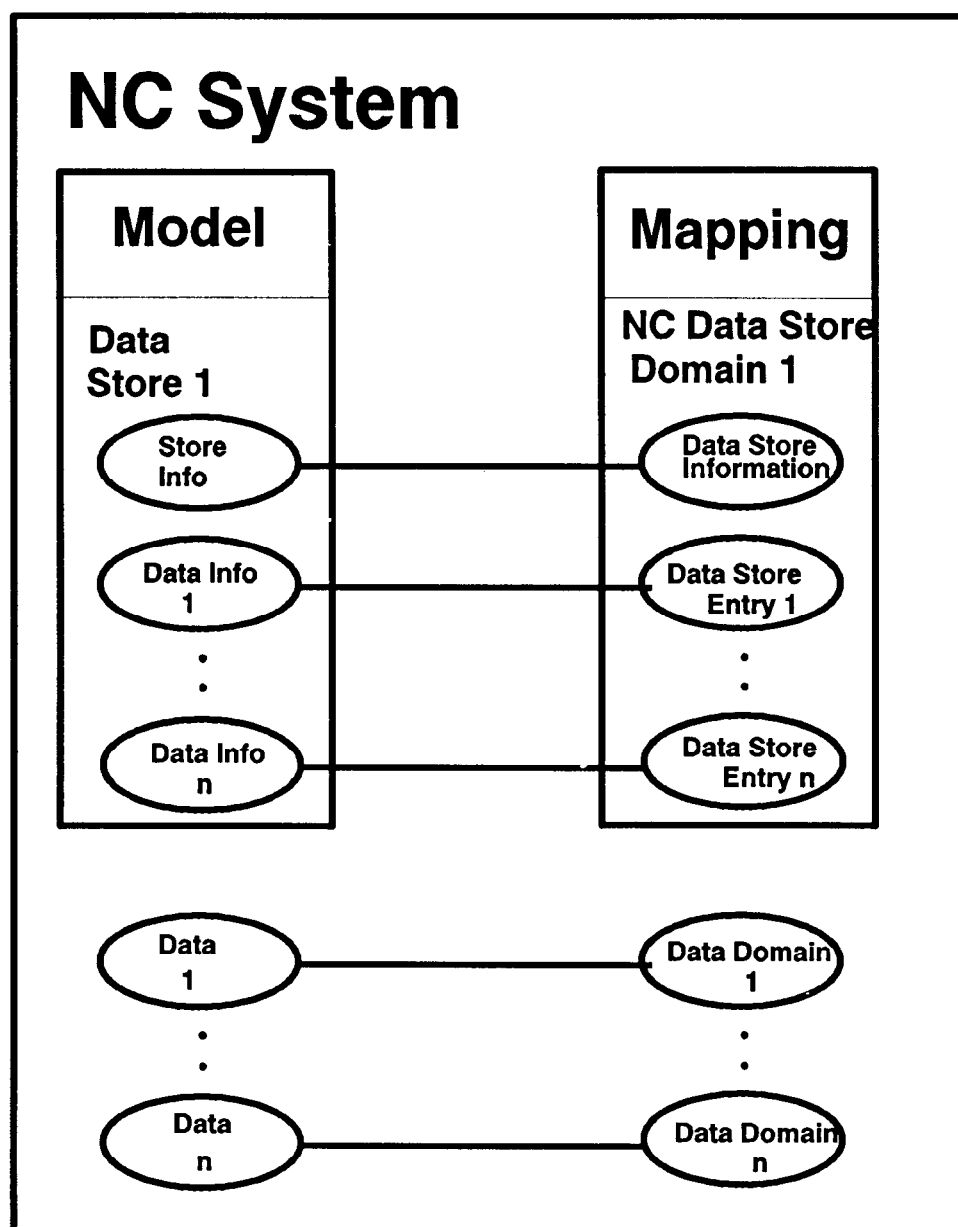


Figure 18: Data Store Mapping

The NC Data Store is normally not modified directly. Its objects are created and deleted locally as a result of the creation and removal of domains from the NC VMD. How the NC Data Store image is altered, that is, the creation of NC Data Store entries, and the assignment of objects which are NC Data Store domain specific, shall be a local issue. Following are the possible and visible side effects of an upload and download of domains in terms of the NC Data Store: