TECHNICAL REPORT

ISO/IEC TR 13345

First edition
1994-07-15

Industrial automation systems — Specification of subsets of the protocol for ISO/IEC 9506

Systèmes d'automatisation industrielle — Spécification de sous-ensembles du protocole pour l'ISO/CEI 9506

Reference number
ISO/IEC TR 13345:1994(E)

Contents

Page

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.

The main task of technical committees is to prepare International Standards, but in exceptional circumstances a technical committee may propose the publication of a Technical Report of one of the following types:

— type 1, when the required support cannot be obtained for the publication of an International Standard, despite repeated efforts;

— type 2, when the subject is still under technical development or where for any other reason there is the future but not immediate possibility of an agreement on an International Standard;

— type 3, when a technical committee has collected data of a different kind from that which is normally published as an International Standard ("state of the art", for example).

Technical Reports of types 1 and 2 are subject to review within three years of publication, to decide whether they can be transformed into International Standards. Technical Reports of type 3 do not necessarily have to be reviewed until the data they provide are considered to be no longer valid or useful.

ISO/IEC TR 13345, which is a Technical Report of type 2, was prepared by Technical Committee ISO/TC 184, *Industrial automation systems and integration,* Subcommittee SC 5, *Architecture and communication.*

This document is being issued in the type 2 Technical Report series of publications (according to subclause G.4.2.2 of part 1 of the ISO/IEC Directives, 1992) as a "prospective standard for provisional application" in the field of industrial automation because there is an urgent need for guidance on how standards in this field should be used to meet an identified need.

This document is not be regarded as an "International Standard". It is proposed for provisional application so that information and experience of its use in practice may be gathered. Comments on the content of this document should be sent to the ISO Central Secretariat.

A review of this type 2 Technical Report will be carried out not later than two years after its publication with the options of: extension for another two years; conversion into an International Standard; or withdrawal.

Annex A forms an integral part of this Technical Report.

Industrial automation systems — Specification of subsets of the protocol for ISO/IEC 9506

1 Scope

ISO/IEC 9506-1 and ISO/IEC 9506-2 were published in 1990. Since publication, considerable experience has been gained by implementors identifying defects and other shortcomings. One major shortcoming of ISO/IEC 9506-2 is that although it allows subsetting of the protocol in accordance with parameters declared at the time of association establishment, no clear notation for describing the subsets is present. Moreover, for some combinations of parameters, there are possible ambiguities in the specification of the subset.

MMS prescribes a procedure, used at the time of association establishment, in which sets of parameters are exchanged for the purpose of identifying the services that may be performed during the association. The effect of identifying these parameters is such that the protocol available to be used during the association is limited to a proper subset of the entire protocol specified in ISO/IEC 9506-2. The parameters exchanged are of two types:

a) those that are announced by the two MMS users, the service conformance building block (CBB),

b) those that are negotiated between the two MMS users, the parameter CBB. These CBBs are proposed by the association initiator and either accepted or rejected by the association responder. Negotiation always works to reduce the set proposed, never to augment it.

Declaration of support of any service CBB requires inclusion of the protocol related to that service in the protocol set to be used on that association. Support of a parameter CBB usually results in the inclusion of some optional fields within the protocol of some service request or response. However, in some cases, support of a parameter CBB implies support of one or more service CBBs, regardless of whether or not support for those service CBBs has been declared. It is in such situations that ambiguity may arise. This Technical Report specifies a uniform technique for resolving such ambiguities.

It introduces a formal notation to describe the subsets in an unambiguous form, and a complete specification of the MMS protocol using this new notation. The notation also supports the protocol of the Companion Standards. The protocol of the Robot Companion Standard (ISO/IEC 9506-3), of the NCC Companion Standard (ISO/IEC 9506-4), and of the Process Industries Companion Standard (the subject of future International Standard ISO/IEC 9506-6) are included.

Annex A specifies the File Management Protocol, the subject of annex C of ISO/IEC 9506-2.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this Technical Report. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this Technical Report 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, *Industrial automation systems — Manufacturing Message Specification — Part 1: Service definition.*

ISO/IEC 9506-2:1990, *Industrial automation systems — Manufacturing Message Specification — Part 2: Protocol specification.*

ISO/IEC 9506-1/Amd.1:1993, *Industrial automation systems — Manufacturing Message Specification — Part 1: Service definition AMENDMENT 1 Data exchange.*

ISO/IEC 9506-2/Amd.1:1993, *Industrial automation systems — Manufacturing Message Specification — Part 2: Protocol specification AMENDMENT 1 Data exchange.*

ISO/IEC 9506-1/Amd.2:—[1], *Industrial automation systems — Manufacturing Message Specification — Part 1: Service definition AMENDMENT 2 Conditioned service response.*

ISO/IEC 9506-2/Amd.2:—[1], *Industrial automation systems — Manufacturing Message Specification — Part 2: Protocol specification AMENDMENT 2 Conditioned service response.*

ISO/IEC 9506-3:1991, *Industrial automation systems — Manufacturing Message Specification — Part 3: Companion standard for robotics.*

ISO/IEC 9506-4:1992, *Industrial automation systems — Manufacturing Message Specification — Part 4: Companion standard for numerical control.*

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

3 Protocol subset specification

3.1 Notation

The notation being introduced by this technical report has the form of a preprocessor language in which ASN.1 is embedded. It is very similar in concept to the macro preprocessor for the C language. There are only three commands used in this notation:

- IF (<list of arguments>)

- ELSE

- ENDIF.

1) To be published.

The IF command requires an argument list enclosed in parentheses; the arguments are the names of the conformance building blocks, either service or parameter. One or more such arguments must appear. If there is more than one argument, the arguments are separated by one or more spaces. The argument is treated as a boolean variable that has the value true if the corresponding service or parameter building block is supported as a result of the MMS Initiate exchange. If there is one argument, then the lines following the IF statement up to the ELSE statement or to the matching ENDIF statement (if no ELSE statement appears) are to be included in the resulting ASN.1 definition if and only if the conformance building block so named in supported. If there is more than one argument, the lines following the IF statement are to be included if any of the conformance building blocks in the argument list is supported. (This can be thought of as a 'logical OR' function of the conformance building blocks.)

IF statements may be nested to any depth; the effect of

```
IF ( x )
IF ( y )
```

is to include the lines following these commands if and only if both x and y are true, that is if conformance block x and conformance block y are both included. (This can be thought of as a 'logical AND' function of the conformance building blocks.)

The ELSE statement may appear to allow ASN.1 statements to be included if a conformance building block is not true. Its use is similar to the normal use of ELSE in programming languages.

The ENDIF statement is used to end the scope of an IF statement or ELSE statement. Each IF statement must have a matching ENDIF statement.

3.2 Determination of the effective protocol

The protocol effective for any specific combination of service and parameter CBBs can be determined by the following procedure:

a) For each service CBB and parameter CBB declared or negotiated by the Initiate exchange, set the corresponding argument equal to true.

b) Process the entire ASN.1 module specified in clause 4 and in Annex A. For each IF statement, evaluate the argument.

i) If any of the elements in the argument is true, retain the statements following the IF statement up to a matching ENDIF or ELSE statement, if present. Discard the statements following the ELSE statement up to the matching ENDIF.

ii) If all the elements of the argument are false, discard the statements following up to the matching ELSE or ENDIF. If an ELSE statement is present, retain the statements following it to the matching ENDIF.

iii) Discard the IF statement, its matching ENDIF, and the ELSE, if present. The result should be an ASN.1 module devoid of IF, ELSE, and ENDIF statements.

c) In each production replace any occurrence of a comma followed immediately by a right brace with a right brace; i.e., delete such commas.

d) Form an ASN.1 working module of productions containing only the first production (i.e. the production MMSPDU from 4.1).

e) Add to the ASN.1 working module any productions referenced in the working module that are not already contained in that module.

f) Repeat step e) until no new productions are added.

The resulting ASN.1 module is the module that is effective for this combination of CBBs. Receipt of a PDU which is not consistent with this module should result in a reject.

3.3 Protocol for companion standards

In addition to the protocol of ISO 9506-2, the protocol for all the amendments and companion standards has also been included in this report. In order to generate the protocol for a companion standard, the set of parameter CBBs has been augmented with three new symbols:

csr - true if this is an instance of the companion standard for robots,

csnc - true if this is an instance of the companion standard for NC equipment,

cspi - true if this is an instance of the companion standard for the process industries.

Except for the fact that the companion standards define a different abstract syntax that must be negotiated using presentation negotiation, each companion standard acts just like a parameter CBB.

In order to accomodate these companion standards, the list of service and parameter CBBs is augmented by the service and parameter CBBs from those standards.

The complete list of service CBBs is:

status
getNameList
identify
rename
read
write
getVariableAccessAttributes
defineNamedVariable
defineScatteredAccess
getScatteredAccessAttributes
deleteVariableAccess
defineNamedVariableList
getNamedVariableListAttributes
deleteNamedVariableList
defineNamedType
getNamedTypeAttributes
deleteNamedType
input
output
takeControl
relinquishControl
defineSemaphore

deleteSemaphore
reportSemaphoreStatus
reportPoolSemaphoreStatus
reportSemaphoreEntryStatus
initiateDownloadSequence
downloadSegment
terminateDownloadSequence
initiateUploadSequence
uploadSegment
terminateUploadSequence
requestDomainDownload
requestDomainUpload
loadDomainContent
storeDomainContent
deleteDomain
getDomainAttributes
createProgramInvocation
deleteProgramInvocation
start
stop
resume
reset
kill
getProgramInvocationAttributes
obtainFile
defineEventCondition
deleteEventCondition
getEventConditionAttributes
reportEventConditionStatus
alterEventConditionMonitoring
triggerEvent
defineEventAction
deleteEventAction
getEventActionAttributes
reportEventActionStatus
defineEventEnrollment
deleteEventEnrollment
alterEventEnrollment
reportEventEnrollmentStatus
getEventEnrollmentAttributes
acknowledgeEventNotification
getAlarmSummary
getAlarmEnrollmentSummary
readJournal
writeJournal
initializeJournal
reportJournalStatus
createJournal
deleteJournal
getCapabilityList
fileOpen
fileRead
fileClose
fileRename
fileDelete
fileDirectory
unsolicitedStatus
informationReport
eventNotification
attachToEventCondition
attachToSemaphore
conclude
cancel
getDataExchangeAttributes
exchangeData
vMDStop
vMDReset
select
alterProgramInvocationAttributes
initiateUnitControlLoad
unitControlLoadSegment
unitControlUpload
startUnitControl
stopUnitControl
createUnitControl
addToUnitControl
removeFromUnitControl
getUnitControlAttributes
loadUnitControlFromFile
storeUnitControlToFile
deleteUnitControl
defineEventConditionList
deleteEventConditionList
addEventConditionListReference
removeEventConditionListReference
getEventConditionListAttributes
reportEventConditionListStatus
alterEventConditionListMonitoring
defineAccessControlList
getAccessControlListAttributes
reportAccessControlledObjects
deleteAccessControlList
changeAccessControl

The complete list of parameter (and parameter-like) CBBs is:

str1
str2
vnam
valt
vadr
vsca
tpy
vlis
real
cei
des
dei
recl
csr
csnc
cspi

4 Protocol module

This clause presents the entire protocol module using the macro notation described in Clause 3. It includes all of the protocol for the companion standard for robots, ISO 9506-3, the companion standard for NC equipment, ISO 9506-4, and the companion standard for the process industries, ISO 9506-6. It also includes the protocol for the Data Exchange services and the Conditioned Service Response services.

4.1 MMSpdu

```
IF ( csr csnc cspi )
IF ( csr )
ISO-9506-MMS-ROBOT-1 {iso standard 9506 part(3) mms-robot-module-version1(2) }
ENDIF
IF ( csnc )
ISO-9506-MMS-NCCS-1 {iso standard 9506 part(4) mms-nccs-module-version1(2)}
ENDIF
IF ( cspi )
ISO-9506-MMS-PROCESS-1 {iso standard 9506 part(6) mms-process-module-version1(2)}
ENDIF
ELSE
ISO-9506-MMS-1 {iso standard 9506 part(2) mms-abstract-syntax-version1(1) }
ENDIF

DEFINITIONS ::= BEGIN

IMPORTS AP-title, AP-invocation-id, AE-qualifier,
   AE-invocation-id, Authentication-value
   FROM ISO-8650-ACSE-1
{iso standard 8650 abstract-syntax(2) acse-pdi(1)};

MMSpdu ::= CHOICE {
 confirmed-RequestPDU      [0] IMPLICIT Confirmed-RequestPDU,
 confirmed-ResponsePDU     [1] IMPLICIT Confirmed-ResponsePDU,
 confirmed-ErrorPDU        [2] IMPLICIT Confirmed-ErrorPDU,
IF ( unsolicitedStatus informationReport eventNotification )
 unconfirmed-PDU           [3] IMPLICIT Unconfirmed-PDU,
ENDIF
 rejectPDU                 [4] IMPLICIT RejectPDU,
IF (cancel)
 cancel-RequestPDU         [5] IMPLICIT Cancel-RequestPDU,
 cancel-ResponsePDU        [6] IMPLICIT Cancel-ResponsePDU,
 cancel-ErrorPDU           [7] IMPLICIT Cancel-ErrorPDU,
ENDIF
 initiate-RequestPDU       [8] IMPLICIT Initiate-RequestPDU,
 initiate-ResponsePDU      [9] IMPLICIT Initiate-ResponsePDU,
```

```
 initiate-ErrorPDU          [10] IMPLICIT Initiate-ErrorPDU,
 conclude-RequestPDU        [11] IMPLICIT Conclude-RequestPDU,
 conclude-ResponsePDU       [12] IMPLICIT Conclude-ResponsePDU,
 conclude-ErrorPDU          [13] IMPLICIT Conclude-ErrorPDU }
```

4.2 Confirmed-RequestPDU

```
Confirmed-RequestPDU ::= SEQUENCE {
 invokeID                 Unsigned32,
IF (attachToEventCondition attachToSemaphore )
 listOfModifier           SEQUENCE OF Modifier OPTIONAL,
ENDIF
                          ConfirmedServiceRequest,
IF ( csr cspi )
                          [79] CS-Request-Detail OPTIONAL
                          -- shall not be transmitted if value is the value
                          -- of a tagged type derived from NULL
ENDIF
 }
```

4.3 Unconfirmed-PDU


```
Unconfirmed-PDU ::= SEQUENCE {
                          UnconfirmedService,
IF ( cspi )
                          [79] CS-Unconfirmed-Detail OPTIONAL
                          -- shall not be transmitted if value is the value
                          -- of a tagged type derived from NULL
ENDIF
 }
```


4.4 Confirmed-ResponsePDU

```
Confirmed-ResponsePDU ::= SEQUENCE {
 invokeID                    Unsigned32,
                          ConfirmedServiceResponse,
IF ( csr cspi )
                          [79] CS-Response-Detail OPTIONAL
                          -- shall not be transmitted if value is the value
                          -- of a tagged type derived from NULL
ENDIF
 }
```