

Information on

Industrial automation systems – Manufacturing Message Specification

ISO 9506; first edition 2000-08-15

By Neal Laurance (Editor ISO 9506) and
Karlheinz Schwarz (Convenor ISO TC 184 SC 5 WG 2)

20 July 2001

1 Introduction and Background

The present publication of ISO 9506 (first edition 2000-08-15) has several problems; some of them are related to the fact that the wrong version of 9506 was published. This problem is most likely Neal's fault, although he still does know how it happened. As a result, the published document contains two kinds of errors that should have been corrected had the proper version been published.

This note is to inform you about the situation, give details and proposals on how to solve the problems.

2 Problems found in the current official version

2.1 Protocol errors

Numerous protocol errors are present in the published document. For example, the word "CHOICE", a key word in ASN.1, is spelled with lower case letters in several places. Also, there are several missing commas in the ASN.1. These errors invalidate the published ASN.1 on syntactic grounds.

2.2 Agreed changes not implemented

Several agreed to changes were not implemented. In particular, the extension of the character strings to permit Kanji did not occur. Also, the words "deprecated" still appear in the annex on Scattered Access even though there was general agreement that they should be removed.

In the interim between the publication and the present, other errors have emerged. In part, these problems have been identified based on attempts by Bancroft Scott at OSS Nokalva to compile the MMS protocol. As a result of our correspondence with Bancroft, it has also come to our attention that there is a continuing problem with respect to PER and MMS. Since we think that PER may become more significant for MMS in the future, we think we need to review this situation as well. The resolution of these problems also involve the problems mentioned above. The next paragraphs review these two problem areas.

Object Model

The object model was introduced into the second edition of MMS in order to add precision to the informal object models of the first edition. They used the facilities of Part 2 of ASN.1,

(ISO 8824-2). However, the intent of the ASN.1 designers for the object models was to provide a representation of constraints to the protocol that could be checked at run time by the encoders and decoders. MMS did not use object models in this way. Rather, it used the object models to provide a reference model for the service procedures. Put another way, although the object models are a formal part of the ASN.1 protocol, the resultant PDUs do not depend on them syntactically. You can remove the object models from the ASN.1 module without an effect on the protocol.

As a result of attending a course in ASN.1 at OSS Nokalva, Neal has learned that the object models in MMS have serious deficiencies with respect to the intent of the ASN.1 standard. We propose to change these object models in order to bring them into conformance with the semantics of the ASN.1 standard. Even then, the protocol will not depend directly on the models.

It is possible that, with sufficient work, many (perhaps all) of the constraints in the MMS standard could be put into the constraint language of ISO 8824-2 and ISO 8824-4 so that automatic encoders and decoders could be generated directly from the protocol. However, the resulting syntax would be so much more complex that it would not be recognized by many of the present practitioners of MMS technology. As we move to more automated development of MMS systems, the use of such automated tools as the OSS ASN.1 compiler generator may make this change desirable.

PER Compatibility

A more serious problem, one not anticipated by the ballot, is that this version of MMS, and probably the first version as well, depending on how it is being used, is not compatible with the PER encryption rules. The problem is that PER reorders the elements of a choice and elements in a sequence. Great care was taken in MMS to ensure that the tags of all elements remain the same through the various development stages. However, PER ignores the tags. It provides its own indexing. And, in order to maintain PER compatibility, (1) one cannot remove alternatives from a CHOICE, and (2) one cannot replace a primitive type by an alternative of a CHOICE. This problem arises in two principle areas in the MMS syntax.

In order to extend the Identifier to handle non English characters, we have replaced.

```
Identifier ::= VisibleString (FROM .... etc.)
```

with

```
Identifier ::= CHOICE {  
    VisibleString (FROM..... etc.),  
    ExtendedString }
```

This will work in BER but fail in PER.

And the present treatment of protocol subsets in which we remove alternatives from the service vectors (ConfirmedServiceRequest and ConfirmedServiceResponse and others) results in incompatible PER protocols.

The first question we must ask ourselves: is this problem significant? Again, we have to examine two cases. The first has to do with respect to compatibility of version 1 with version 2.

We believe that there is no significant implementation of version one using PER; any PER problems are in the future. So, with respect to any backward compatibility problems, we only have to be concerned with BER. In that respect, we think the present approach is adequate. **Please, if anyone knows of any significant problems involving present implementations of PER, we will need to know it soon.** At the very least, we think we need to add

words to the Introduction indicating that the new version is not compatible with respect to the old version if PER is employed.

However, even looking forward, there are problems with the use of PER. These have to do with removing alternatives from CHOICES. As it stands now, the document says that each MMS user approaches an MMS session with a reduced syntax appropriate to the syntax that he is prepared to receive (as a server) or send (as a client). In the initial dialogue, the two parties negotiate down to a compatible intersection of their two syntaxes that will be used in that association.

Now, if by prior agreement, both parties approach the dialogue with the same subset of MMS, PER will work. However, in the general case, one party will have one subset of MMS in its implementation and the other party will have a different subset of MMS. That is, the CHOICE vectors will not be identical; the effective communication will be limited to the intersection of their two CHOICE vectors. The problem is that the indexes of the alternatives in the two CHOICE vectors will not be the same and the PER encoded communication will fail. (BER will work fine!).

We imagine that in most cases of MMS communication, there is really prior agreement about the protocol subset to be employed in the communication, and no problems will result. However, this is equivalent to saying that the MMS Initiate negotiation is unnecessary. Sooner or later, we will come upon the case in which the MMS Initiate negotiation is being usefully employed, and the resulting PER communication will fail. Do we fix this now or later?

In the case of expanding the content of the strings to include non English characters (European extended set, Hangul, etc.) there is another problem. If the two sides have not agreed a priori to employ non English characters, this communication will also fail. Therefore, the use of non English characters in character strings in general should be governed by another parameter support option bit. This should be included in the Initiate negotiation. This is an oversight from the original document.

3 Proposal to remedy the problems

3.1 Add a new Parameter Support Option bit

Let us call it "Exch" for extended character set.

Change the Identifier production to be

Identifier ::=

IF (exch)

CHOICE {

VisibleString (FROM etc.)(SIZE (1..32)),

UTF8String (SIZE(1..32))

}

ELSE

VisibleString (FROM ...etc,)(SIZE (1..32))

ENDIF

If the extended character set is not negotiated, the Identifier is the same as it was in version 1 and has the same PER encoding (as well as the same BER encoding).

Change the MMSString production in the same way.

3.2 Change the various CHOICE vectors that depend on the service and parameter support options

This proposal is more complex. It hasn't worked out completely yet.

```
ConfirmedServiceRequest ::= CHOICE {  
  IF (status)  
    status      [4] ConfirmedStatusRequest,  
  ELSE  
    status      [4] NULL,  
  ENDIF
```

That is, for each of the suppressed options in a CHOICE vector, instead of deleting the choice, we replace it by a NULL (or perhaps by a RejectPDU, We haven't yet thought that through.) However, the intent is to maintain the same number of choices in the vector in order that PER encoding is always the same.

As said above, this is solely to maintain the utility of the Initiate negotiation. If, in reality, we always rely on prior agreement rather than the Initiate dialogue, this is unnecessary; however, in that case we should remove the Initiate dialogue.

Please give us some guidance on these questions. In doing so, we must keep in mind not only the people who have already implemented MMS (they must be satisfied with the direction) but also those who will implement MMS in the future.

Thanks for your cooperation!

If you have any question, please feel free to contact either Neal or Karlheinz.

With Regards,

Neal Laurance
Laurance Associates
876 Heather Way
Ann Arbor, MI 48104-2734
USA

Phone +1 (7 34) 7 61 19 96
Fax +1 (7 34) 7 61 19 96
Email Neal.Laurance@ieee.org

Karlheinz Schwarz
Schwarz Consulting Company, SCC
Im Eichbaeumle 108
76139 Karlsruhe
Germany

Phone +49-721-684844
Fax +49-721-679387
Email schwarz@scc-online.de