Working Implementation Agreements for Open Systems Interconnection Protocols: Part 5 - Upper Layers

Output from the June 1994 Open Systems Environment Implementors' Workshop (OIW)

SIG Chair: **James Quigley, Hewlett Packard**
SIG Editors: **Debbie Britt, NCTS** **Laura Emmons, Telenex**

Foreword

This part of the Working Implementation Agreements was prepared by the Upper Layers Special Interest Group (ULSIG) of the for Open Systems Environment Implementors' Workshop (OIW). See Part 1 - Workshop Policies and Procedures in the "Draft Working Implementation Agreements Document" for the workshop charter.

Text in this part has been approved by the Plenary of the above-mentioned Workshop. This part replaces the previously existing chapter on this subject.

Only the pages that were changed in June 1994 are being printed. Please refer to the March 1994 Working Document for additional information.

Future changes and additions to this version of these Implementor Agreements will be published as a new part. Deleted and replaced text will be shown as struck. New and replacement text will be shown as shaded.

Table of Contents

Part 5 - Upper Layers 1

0 Introduction 1

1 Scope 1

2 Normative References 1

3 Status 1

4 Errata 1
4.1 ISO Defect Solutions 1
4.2 Technical Corriagenda and Defect Reports 1
4.3 Defect Registers 2
4.4 Exception Handling 2

5 Association Control Service Element 2
5.1 Introduction 2
5.2 Services 2
5.3 Protocol Agreements 2
5.3.1 Application Context 2
5.3.2 AE Title 2
5.3.3 Peer Entity Authentication 2
5.4 Abort APDU 3
5.5 Connectionless 3

6 ROSE 3

7 RTSE 3

8 Presentation 3
8.1 Introduction 3
8.2 Service 3
8.3 Protocol Agreements 3
8.3.1 Transfer Syntaxes 3
8.3.2 Presentation Context Identifier 3
8.3.3 Default Context 4
8.3.4 P–Selectors 4
8.3.5 Provider Abort Parameters 4
8.3.6 Provider Aborts and Session Version 4
8.3.7 CPC–Type 4
8.3.8 Presentation–context–definition–result–list 4
8.3.9 RS–PPDU 4
8.4 Presentation ASN.1 Encoding Rules 4
8.5 Presentation Data Value (PDV) 5

8.6 Connection Oriented 5
8.7 Connectionless 5

9 Session 5
9.1 Introduction 5
9.2 Services 5
9.3 Protocol Agreements 5
9.3.1 Concatenation 5
9.3.2 Segmenting 5
9.3.3 Reuse of Transport Connection 6
9.3.4 Use of Transport Expedited Data 6
9.3.5 Use of Session Version Number 6
9.3.5.1 Selection of session version 6
9.3.5.2 User data in session version 2 6
9.3.6 Receipt of Invalid SPDUs 6
9.3.7 Invalid SPM Intersections 6
9.3.8 S–Selectors 6
9.4 Connectionless 7

10 Universal ASN.1 Encoding Rules 7
10.1 Tags 7
10.2 Definite Length 7
10.3 External 7
10.4 Integer 7
10.5 String Types 7
10.6 Extensibility 7

11 Additions to ISP on Common Upper Layer Requirements 8
11.1 Service 8
11.2 Provider Abort Parameters 8
11.3 Concatenation 8
11.4 Segmenting 8
11.5 Reuse of Transport Connection 8
11.6 Use of Transport Expedited Data 8

12 Character Sets 8

13 Conformance 9

14 Specific ASE Requirements 9
14.1 FTAM Phase 2 9
14.2 MHS 9
14.3 DS Phase 1 9
14.4 Virtual Terminal 9
14.5 MMS 9
14.6 Transaction Processing 9
14.7 Network Management 10
14.8 Remote Database Access 10

Annex A (normative)

Object Identifier Register 14
A.1 Register Index 14
A.2 Object Identifier Descriptions 14

Annex B (informative)

Recommended Practices 15

Annex C (informative)

Backward Compatibility 16

Annex D (normative)

Working Draft of new ISP on CL-CULR Specification 20

Annex E (informative)

Upper Layer SIG Registered Questions List 21

Part 5 - Upper Layers

Editor's Note - All references to Stable Agreements in this section are to Version 8.

Editor's Note - Clauses 1 through 12 will be replaced by appropriate references to ISP 11188-1 (Common Upper Layers Requirements).

0 Introduction

(Refer to Stable Agreements Document)

1 Scope

(Refer to Stable Agreements Document)

2 Normative References

(Refer to Stable Agreements Document)

3 Status

This version of the upper layer agreements is under development.

4 Errata

4.1 ISO Defect Solutions

(Refer to Stable Implementation Agreements).

4.2 Technical Corriagenda and Defect Reports

(Refer to Stable Implementation Agreements).

4.3 Defect Registers

(Refer to Stable Implementation Agreements).

4.4 Exception Handling

(Refer to Stable Implementation Agreements).

5 Association Control Service Element

5.1 Introduction

(Refer to Stable Agreements Document)

5.2 Services

(Refer to Stable Agreements Document)

5.3 Protocol Agreements

5.3.1 Application Context

(Refer to Stable Agreements Document)

5.3.2 AE Title

(Refer to Stable Agreements Document)

5.3.3 Peer Entity Authentication

(Refer to Stable Agreements Document)

5.4 Abort APDU

(Refer to Stable Agreements Document)

5.5 Connectionless

(Refer to Stable Agreements Document)

6 ROSE

(Refer to Stable Agreements Document)

7 RTSE

(Refer to Stable Agreements Document)

8 Presentation

8.1 Introduction

(Refer to Stable Agreements Document)

8.2 Service

(Refer to Stable Implementation Agreements).

8.3 Protocol Agreements

8.3.1 Transfer Syntaxes

(Refer to the Stable Agreements Document)

8.3.2 Presentation Context Identifier

(Refer to Stable Agreements Document)

8.3.3 Default Context

(Refer to Stable Agreements Document)

8.3.4 P-Selectors

(Refer to the Stable Agreements Document)

8.3.5 Provider Abort Parameters

(Refer to Stable Implementation Agreements).

Editor's Note -

8.3.6 Provider Aborts and Session Version

(Refer to the Stable Agreements Document)

8.3.7 CPC-Type

(Refer to the Stable Agreements Document)

8.3.8 Presentation-context-definition-result-list

(Refer to the Stable Agreements Documents)

8.3.9 RS-PPDU

(Refer to the Stable Agreements Documents)

8.4 Presentation ASN.1 Encoding Rules

(Refer to the Stable Agreements Document)

8.5 Presentation Data Value (PDV)

(Refer to the Stable Agreements Document)

8.6 Connection Oriented

(Refer to the Stable Agreements Document)

8.7 Connectionless

(Refer to Stable Agreements Document)

9 Session

9.1 Introduction

(Refer to Stable Agreements Document)

9.2 Services

(Refer to Stable Agreements Document)

9.3 Protocol Agreements

9.3.1 Concatenation

(Refer to Stable Implementation Agreements).

Editor's Note -

9.3.2 Segmenting

(Refer to Stable Implementation Agreements).

Editor's Note -

9.3.3 Reuse of Transport Connection

(Refer to Stable Implementation Agreements).

Editor's Note -

9.3.4 Use of Transport Expedited Data

(Refer to Stable Implementation Agreements).

Editor's Note -

9.3.5 Use of Session Version Number

9.3.5.1 Selection of session version

(Refer to the Stable Agreements Documents)

9.3.5.2 User data in session version 2

(Refer to the Stable Agreements Document)

9.3.6 Receipt of Invalid SPDUs

(Refer to the Stable Agreements Document)

9.3.7 Invalid SPM Intersections

(Refer to the Stable Agreements Document)

9.3.8 S-Selectors

(Refer to the Stable Agreements Document)

9.4 Connectionless

(Refer to Stable Agreements Document)

10 Universal ASN.1 Encoding Rules

10.1 Tags

(Refer to the Stable Agreements Document)

10.2 Definite Length

(Refer to the Stable Agreements Document)

10.3 External

(Refer to the Stable Agreements Document)

10.4 Integer

(Refer to the Stable Agreements Document)

10.5 String Types

(Refer to the Stable Agreements Document)

10.6 Extensibility

(Refer to the Stable Agreements Document)

11 Additions to ISP on Common Upper Layer Requirements

11.1 Service

(Refer to Stable Agreements Document)

11.2 Provider Abort Parameters

(Refer to Stable Agreements Document)

11.3 Concatenation

(Refer to Stable Agreements Document)

11.4 Segmenting

(Refer to Stable Agreements Document)

11.5 Reuse of Transport Connection

(Refer to Stable Agreements Document)

11.6 Use of Transport Expedited Data

(Refer to Stable Agreements Document)

12 Character Sets

(Refer to part 21 -- a new chapter expressly for character sets.)

13 Conformance

(Refer to Stable Agreements Document)

14 Specific ASE Requirements

14.1 FTAM Phase 2

(Refer to Stable Agreements Document)

14.2 MHS

(Refer to Stable Agreements Document)

14.3 DS Phase 1

(Refer to Stable Agreements Document)

14.4 Virtual Terminal

(Refer to Stable Agreements Document)

14.5 MMS

(Refer to Stable Agreements Document)

14.6 Transaction Processing

(Refer to Stable Agreements Document)

14.7 Network Management

(Refer to Stable Agreements Document)

14.8 Remote Database Access

~~(Refer to Stable Agreements Document)~~

Table Profile requirements list proforma

	Item / variable	Compliant choice	Specification's choice		Constraint / value
			Client	Server	
1	*Establishment-initiator*	m; o; i	M	I	Both shall not be "i".
2	*Establishment-responder*	m; o; i	I	M	
3	*Establishment-responder-reject*	m; o; i	I	M	The value shall be "i" if *Establishment-responder* has the value "i"
4	*Normal-data-requestor*	m; o; i	M	M	Both may be "i".
5	*Normal-data-acceptor*	m; o; i	M	M	
6	*Release-requestor*	m; o; i	M	I	Both may be "i".
7	*Release-acceptor*	m; o; i	I	M	
8	*Authentica*tion	m; o; i	O	O	
9	*Application-context-negotiation*	m; o; i	O	O	
10	*Transport-expedited*	m; o; i	I	I	

11	Number of presentation-contexts required	1 or more	2	2	The value chosen includes the presentation-context used for ACSE PDUs.
12	ISO/IEC ISP 11188-1 compliance?[1]	yes	Yes	Yes	If the answer is not "yes", the referencing specification may not claim mOSI compliance
13	Status values for all open (*) parameters (see table D.2)	all "m"; all "o"; all "i"; or "mixed"	Mixed	Mixed	If the answer is "mixed" (i.e. not all "m" and "–", or not all "o" and "–", or not all "i" and "–"), details shall be given in table D.2

[1] See clause 2 and ISO/IEC ISP 11188-1, annex B.

Table Open parameters (*)

	Referenced table (in annexes A, B and C)	Parameter	Specification's statement — Sender [a]	Specification's statement — Receiver [a]	Constraint / value
1	A.6.1	Calling AE title			Includes both the AP title and AE
2	[AARQ]	Called AE title			qualifier for each
3		Calling invocation ids			Includes both the AP invocation identifier and the AE invocation
4		Called invocation ids			identifier for each
5		User Information			
6	A.6.2	Responding AE title			Includes both the AP title and AE qualifier
7	[AARE]	Responding Invocation Identifiers			Includes both the AP invocation identifier and the AE invocation identifier
8		User Information			
9	A.6.3	Reason			
10	[RLRQ]	User Information			
11	A.6.4	Reason			
12	[RLRE]	User Information			

13	A.6.5 [ABRT]	User Information			
14	A.7.1 [AARQ and	Form 1 (Directory name)			For Receiver, compliant answer is "m" or "–". That is, if AE titles
15	AARE]	Form 2 (Object id+integer)			are supported for receiving, both forms are mandatory.
16	B.4.1 [CP]	Default context name			May be used for simple encoding of ACSE and user PCI
17	B.4.3 [CPR]	Default context result			

[a]Compliant answer for each row is "m", "o", "i" or "–", unless indicated otherwise.

Annex A (normative)

Object Identifier Register

A.1 Register Index

(Refer to Stable Agreements Document)

A.2 Object Identifier Descriptions

(Refer to Stable Agreements Document)

Annex B (informative)

Recommended Practices

(Refer to Stable Agreements Document.)

Annex C (informative)

Backward Compatibility

Version & Section		
Issue	Changed	Backward Compatibility
Restrictions on minimum number of octets implementations shall be able to receive.	V1E2 5.5.3.2	Interworking problems may occur, since implementations could send more than 128 octets. [An implementation that conforms to versions previous to V1E2 as an initiator and V3E1 as a responder will be able to interoperate.]
Agreements on AE Title, AP Title, and AE Qualifier changed.	V1E3 section 5.5.3.3 & V1E4 section 5.5.3.3	Interworking problems may occur between implementations that expect different forms of AP Title and AE Qualifier to be used. [Implementations that accept any form of these parameters will interwork with initiators that conform to earlier versions.]
Restrictions on encoding of "Presentation Context Identifier."	V2E1 section 5.8.3.3	Interworking problems may occur since implementations could encode negative numbers. [An implementation that conforms to versions previous to V2E1 as a responder and V3E1 as an initiator will be able to interoperate.]
Mode selector as first element in set	V1E4 section 5.6.3.4	This will cause interworking problems for those implementations that don’t encode "mode selector" as the first element in the set. [An implementation that conforms to versions previous to V1E4 as an initiator and V3E1 as a responder will be able to interoperate.]

Version & Section		
Issue	Changed	Backward Compatibility
Restrictions on encoding of "protocol version" and "presentation requirements."	V2E1 section 5.8.4.2	This will cause interworking problems for those implementations expecting "protocol version" and "presentation requirements" to be encoded in the primitive form. [An implementation that conforms to versions previous to V2E1 as an initiator and V3E1 as a responder will be able to interoperate.]
Restrictions on encoding of "presentation selector."	V2E1 section 5.8.4.3	This will cause interworking problems for those implementations expecting "presentation selector" to be encoded in the primitive form. [An implementation that conforms to versions previous to V2E1 as an initiator and V3E1 as a responder will be able to interoperate with either version.]
Use of default values for Minor syncpoint changed.	V2E3 section 5.11.1.1.1	No backwards compatibility
Addition and deletions of abstract syntaxes.	V2E1 section 5.11.1.3.1	No backwards compatibility
Value for session functional unit "resynchronize" changed.	V2E4 section 5.11.1.4.1	No backwards compatibility
Restrictions on inclusion of "Transfer-syntax-name" in CP PPDU and CPC type.	V3E1 section 5.8.6	Interworking problems will occur for those implementations that expect "Transfer-syntax-name" parameter to be present in the PDV-List even though one transfer syntax was negotiated. [An implementation conforming to V3E1 as an initiator and versions previous to V3E1 as a responder will be able to interoperate.]

Issue	Version & Section Changed	Backward Compatibility
Encoding restrictions on ASN.1 INTEGER type describing PCI.	V3E1 section 5.10.4	Interworking problems will occur since implementations conforming to previous versions could encode PCI integer lengths greater than 4. [Responders that accept integers describing PCI that are encoded in greater than 4 octets and Initiators that conform to V3E1 will be able to interoperate.]
Encoding restrictions on BIT STRING, OCTET STRING, and CHARACTER STRING.	V3E1 section 5.10.5	Implementations that conform to previous versions can expect these strings to have nested constructed encodings and therefore interworking problems will occur. [Responders that accept nested constructed encodings and Initiators that conform to V3E1 will be able to interoperate.]
No extra trailing bits allowed in BIT STRING.	V3E1 section 5.10.6	Interworking problems will occur when implementations that conform to previous versions send extra trailing bits. [Responders accepting extra trailing bits and Initiators that conform to V3E1 will be able to interoperate.]
Restriction on usage of "token item field" and "user data."	V3E1 section 5.9.3.1	Interworking problems will occur since implementations that conform to V1E1 do not expect the "token item field" to be encoded when a category 0 SPDU is concatenated to a category 2 SPDU.
Restrictions on CPC-type values when multiple transfer syntaxes are proposed.	V2E2 section 5.8.3.9	Interworking problems may occur between initiators that send CPC-type values and receivers that do not examine them.

Version & Section		
Issue	Changed	Backward Compatibility
References to ISO 8649 and ISO 8650 changed.	V1E3 section "References."	Interworking problems will occur for those implementations that conform to ISO DIS 8649 and 8650. V1E3 references IS versions of 8649 and 8650.
References to ISO 8326, ISO 8327, ISO 8822, and ISO 8823 changed.	V1E4 section References.	Interworking problems will occur for those implementations that conform to 8326/DAD2, 8327/DAD2, DIS 8822, and DIS 8823. V1E4 referenced 8326/AD2, 8327/AD2, IS 8822, and IS 8823.
AE Title changed according to Amendment 1 to ISO 8650.	V3E1 section 5.5.3.2	Interworking problems will occur between initiators that use AE-title- form 1 and responders that accept only AE-Title-form 2.
Restrictions on usage of "direct references" in ABRT APDU.	V3E1 section 5.5.4	Interworking problems will occur for those implementations that expect the "direct reference" parameter to be included in the ABRT APDU. [An implementation that conforms to V3E1 as an initiator and versions previous to V3E1 as a responder will be able to interoperate.]

Annex D (normative)

Working Draft of new ISP on CL-CULR Specification

Editor's Note - This annex contains the draft proposed text for a new ISP on connectionless upper layer facilities. Electronic readers should refer to the Word for Windows 2.0 formatted document culr4.doc.

Annex E (informative)

Upper Layer SIG Registered Questions List

ULSIG Registered Question List

(1) Summary: Herb Falk's question on ACSE Association Info.

Source: Herb Falk

Date Raised: 26 April, 1993

Issue: Copy of message follows:

The problem is specifically that the ACSE "Association-information", which is an ASN.1 EXTERNAL, has taken the CHOICE of octet-aligned. The ISO specifications and NIST stable agreements seem to be clear on this matter. We will try to explain them as best we can. A hard copy of the Presentation-Connect PDU follows on a separate page. Note that the item circled and marked "1" is the beginning of the PDV-list. Note "2" is the beginning of the Presentation Data List encoded as Single-ASN1-type. Note "3" is the beginning of the Association-Information encoded as an EXTERNAL. Note "4" is the beginning of the External encoding tagged as octet-aligned.

Please reference page 31 of ISO specification ISO-8823 (IS). At the top of the page is found a definition for the PDV-list. Legal presentation data values are a CHOICE of { Single-ASN1-type, octet-aligned, and arbitrary}. This CHOICE is further qualified in section 8.4.2.5, on the following page, to say that the single-ASN1-type shall be used if the PDV-list contains exactly one presentation data value. The ACSE Assocaite-Request PDU shown in the trace has exactly one presentation data value, therefore this encoding rule applies. The PDU conforms to this specification and may be verified in note "2" to be the value 0xA0.

Please refer to page 18 of ISO specification 8650 for a description of the AARQ-apdu. Towards the bottom of the page there is a description of 'user-information'. It states that 'user-information' is IMPLICIT 'Association-information' OPTIONAL. 3 pages later in the same specification is the definition for 'Association-information'. It states that an 'Association-information' field may only be a SEQUENCE OF EXTERNAL. An EXTERNAL is not defined in the ACSE Protocol specification. It is found in the ASN.1 Protocol Specification ISO 8824.

Please refer to ISO specification 8824 (Abstract Syntax Notation One) page 23 for a description of the EXTERNAL. Section 34.7 of 8824 says that:

"If the data value is the value f a single ASN.1 data-type, and if the encoding is an integral number of octets, then the sending implementation shall use any of the encoding choices:

single-ASN1-type
octet-aligned
arbitrary"

According to ISO 8824 it would be legal to send "Associate-information" as octet-aligned at note "4". However, we believe that there is an implementation agreement on this CHOICE of encoding. If you look at the NIST stable agreements on page 12 in section 10.3 there is an implementors agreement on which choice to use in the EXTERNAL. The second sentence in that paragraph reads as follows:

"If a data value to be encapsulated in an EXTERNAL type is an instance of a single ASN.1 type encoded to the basic encoding rules for ASN.1 then the option "single-ASN1-type" shall be chosen as encoding."

We believe that this sentence is why the byte in note "4" should be the value 0xA0 instead of 0x81. This seems to be self-explanatory. However, to make sure that we are not taking this sentence out of context or misinterpreting it, we have placed a call to the Upper Layers chairman of NIST and are asking for a clarification.

Remember that NIST stable agreements are not binding which means that the Computrol MMS is still within the guidelines for this encoding at the current time. But also be advised that these stable agreements are being moved into the upper layer agreements within the next year.

"1" 31 81 84 a0 03 80 01 01 a2 7d 81 02 04 00 82 02 00 01 a4 23 30 10 02 01 01 06 05 28 ca 22 02 01 30 04 06 02 51 01 30 0f 02 01 03 06 04 52 01 00 01 30 04 060 02 51 01 88 02 06 00 89 03 05 40 00 61 45 30 43 02 01 03
"2" a0 3e 60 3c a1 07 06 05 28 ca 22 01 01
"3" be 31 28 2f 06 02 51 01 02 01 01
"4" 81 26 a8 24 80 02 20 00 .....

Responses: From Laura Emmons (laurae@ar.telenex.com) May 10 1993:
I took a look at Herb Falk's defect report and I don't think there is any problem with any of the standards or our position on the use of the EXTERNAL data type. His description of the encoding of the encoding of his layer 6 header seems to be irrelevant. If the MMS-InitiateRequest is a single ASN.1 element (I haven't seen this protocol, but it seems that it is), then the data value of the instance of the Association-information element should be encoded as a single-ASN1-type. Therefore, in his pdu Note 4 should be an 0xA0.

Solution: The data value of the instance of the Association-information element should be encoded as a single-ASN1-type.

Status:	OIW:	Closed - Accepted 03/94
	EWOS:	
	AOW:	

(2) Summary: PGI PI issue from Japan

Source: Jun Yamaguchi (junichi@vnet.ibm.com)

Date Raised: July 22, 1993

Issue: Copy of message follows:
I have a question about ISO 8327. I would like you to clarify an interpretation of this standard.

Base standard states "PGI units and PI units within the same nesting level shall be ordered in increasing value of their PGI and PI codes." in the clause 8.2.6 of ISO 8327.

There are several interpretations for this statement:

1. PGI units shall be ordered in increasing value of their PGI codes. PI units in the same PGI unit shall be ordered in increasing value of their PI codes. PI units without PGI code have the same nesting level with PGI units, and this kind of PI units and PGI units shall be ordered in increasing value of their PGI and PI codes.

2. PGI units shall be ordered in increasing value of their PGI codes. PI units in the same PGI unit shall be ordered in increasing value of their PI codes. PI units without PGI code shall be ordered in increasing value of their PI codes. There are no relationship between PGI units and PI units about the order.

3. PGI units shall be ordered in increasing order of their PGI codes. PI units in the same PGI unit shall be ordered in increasing value of their PI codes. PI units without PGI code have no relationship with other units. So, this kind of PI units may be placed in any position.

Which interpretation is correct, or all wrong?

Responses: From Bob Baker (baker@uxdp5.Tredydev.Unisys.com) July 26:
I reviewed Jun Yamaguchi's session question which you forwarded to the OIW members. We had the same question years ago when we were implementing our Session layer, and I talked with Kim Banker at the time. He was very helpful and we finished our implementation based on his suggestions.

We believe interpretation #1 is the only correct interpretation of the session specification. This interpretation is consistent with what Kim told us and also with our implementation...Interpretations #2 and #3 would permit any of the PI codes which have no PGI code to be present after PGI 193 (User Data) in an SPDU. This is annoying at best, and would probably cause many implementations severe problems.

From Andrew Chandler (a.chandler@xopen.co.uk) August 17
My interpretation is as follows (essentially this is interpretation 1 above):

PGI units shall be ordered in increasing value of their PGI codes.

PI units in the same PGI unit shall be ordered in increasing value of their PI codes.

PGI units and PI units at the same level of nesting shall be ordered in icreasing value of their PGI and PI codes.

Solution: Interpretation 1 is correct.

Status: OIW: Closed - Accepted 09/93
EWOS:
AOW:

25

(3) Summary: Encoding FTAM single PDV list

Source: Kevin Bohan (0004141431@mcimail.com)

Date Raised: July 29, 1993

Issue: Copy of message follows:
I have a question as to what is meant in section 8.5 of the NIST Stable Agreements.

Proginet has an FTAM product that sends back an F-Begin-Group-Response, F-Deselect-Response, F-Close-Response, F-End-Group-Response.

This is done using a single PDV list. We have encoded this PDV-List using the single-ASN1-type. The remote site is kicking this out and they claim that this is not valid.

Is this Valid?

Responses: From Klaus Truoel (truoel@gmd.de) Mar 17, 1994:
Section 8.5 in Part 5 of the OIW Stable Implementor's Agreements (as well as the FTAMISP 10607-1, 7.3 and CULR-1, 7.10 state:
"... shall be encoded either as a single PDV-list (using the octet-aligned choice) or as a series of PDV-lists, ..."
The FTAM Stable Agreements have the same text.
Therefore, encoding as grouped PDUs using a single PDV-list and the single-ASN1-type of encoding is not valid.

Solution: Encoding grouped PDUs using a single PDV-list with the single-ASN1-type choice of encoding is not valid.

Status: OIW: Closed - Accepted 03/94
EWOS:
AOW:

(4) Summary: Ed Kelley question on whether FTAM can directly use P-U-ABORT.

Source:

Date Raised:

Issue:

Responses:

Solution:

Status: OIW: Open - pending
EWOS:
AOW:

(5) Summary: new MMS issue on CUL for Security

Source: MMS SIG

Date Raised: 16 September, 1993

Issue: Copy of liason:

The MMS SIG is investigating the use of various OSI protocols and features for achieving different security requirements for MMS. With further discussion with the Security SIG, it appears that concepts in GULS are adequate for our needs. In particular, the use of the ACSE Functional Unit for Authentication.

As it is likely, that all of the SIGs will need similar requirements for upper layers, we are asking for you to investigate the common needs and, if warrented, develop a version of the Common Upper Layer Requirements that address security.

Responses:

Solution:

Status: OIW: Open - pending
EWOS:
AOW:

(6) Summary: Gary Williams issue on p-u-abort on bad encoding.

Source:

Date Raised: 9 September 1993

Issue: The problem is that we believe that there is a possible contradiction between clause 7.9 of Draft Version 12 of pDISP 11188-1, 1993-01-22 (ISP:Common Upper Layer Requirements) which states:

"If a received PPDU contains improperly encoded data values (including data values embedded with the user data field of a PPDU) and if an abort is issued, then either an ARU or an ARP shall be issued."

and ISO 8823: 1988, clause's 6.4.4.2 and 6.4.4.3 which state that the only response is a P-P-ABORT.

The information that we require is how to start the procedure to address this issue, possibly obtain a contact name, or how to get in touch with he/she in order to resolve the issue.

Responses: From Klaus Truoel (truoel@gmd.de) Aug 8, 1993:
The current draft of Common Upper Layer Requirements is draft 14, and it will hopefully get the approval as PDISP by the Regional Workshops in Sept and Oct. Of course, after that approval it will not be too late to fix bugs if there are any.

The clause which you are questionning is the same also in the latest version. Actually, it is a clause which is in that document (and in the European FTAM ENVs) since many years. It passed several ISO ballots,reviews and discussions with ISO experts.

The reason behind that clause, as far as I can remember the history, is the often discussed problem, which OSI layer would be responsible to detect "improperly encoded data values". Is it the presentation layer or can it in many cases only be done by the application ? In the latter case, the application would initiate the Abort and that would result in an ARU. This is what the clause expresses.

And, by the way, the clauses in ISO 8823 which you reference, specify "if possible". Sometimes it may not be possible if only the application can detect the bug.

As I myself am the editor of the PDISP, you may send all comments or questions to me. In case you are not satisfied with my above explanation and if you want to raise the issue to a broader audience for consideration, I am prepared to take the issue with me to the forthcoming OIW (beginning of Sept.)

and to EWOS (Oct.).

Solution: In the above reference to the base standard 8823, the terms "Invalid or unexpected PPDU" and "Invalid parameter" are both defined as PCI errors. The standard does not discuss the handling of errors found on user-data. Therefore there is no issue. However, it is expected that X3T5 will submit a ballot comment on CULR-1 asking that clarifying text be added to 7.9 staing that "(via an A-ABRT)" be added after the phrase "an ARU".

Status: OIW: Closed - Accepted 03/94
EWOS:
AOW:

(7) Summary: X/Open ROSE PCI must be in BER.

Source:

Date Raised:

Issue:

Responses:

Solution:

Status: OIW: Open - pending
EWOS:
AOW:

(8) Summary: Discrepency in usage of Exception Report SPDU causes conformance testing problems..

Source: Lee Chastain, JITC (lee@huachuca-jitcosi.army.mil)

Date Raised: 14 December 1993

Issue: ISO 8327, sec. 7.25: The EXCEPTION REPORT SPDU is used to report that a protocol error has been detected within the SPM.

ISO 8327, sec. A.4.1.2: NOTE It should be noted that sending an EXCEPTION REPORT SPDU may lead to an SPM deadlock. It is therefore advised to send the ABORT SPDU rather than the EXCEPTION REPORT SPDU, especially in the case of protocol errors.

If an implementor were to take the above advice, under what circumstances would their Session ever send an ER?

ISO 8327-2 (PICS) A.7.12.2/1 If you claim support for the Exceptions functional unit, the requirement to send the ER is mandatory. This appears to be in contradition to the note in the standard.

Given that the note above is not an error in the standard, shouldn't the capability to send the ER be optional rather than mandatory?

Responses: A discrency exists between ISO/IEC 8327 parts 1 and 2. ISO/IEC 8327-1 1994 (E), sec. 7.27 states: the EXCEPTION REPORT SPDU is used to report that a protocol error has been detected within the SPM. The note in section A.4.1.2 of the same document states: It should be noted that noted that sending an EXCEPTION REPORT SPDU may lead to an SPM deadlock. It is therefore advised to send the ABORT SPDU rather than the EXCEPTION REPORT SPDU, especially in the case of protocol errors.

In ISO/IEC DIS 8327-2, sec. A.7.12.2/1, if support is claimed for the exceptions functional unit, the requirement to send the ER SPDU should be optional.

It appears that there is no need for the ER SPDU in 8327. Further study should be given to the elimination of the ER SPDU.

Solution: A defect report will be submitted sugessting the following change to ISO/IEC DIS 8327-2, sec. A.7.12.2:

Exceptions functional unit

SPDU	Status Sdr	Status Rcv	Support Sdr	Support Rcv	Comment
Exception Report (ER)	c32	c33			
Exception Data (ED)	c33	c33			

c32: if [S-FU(EXCEP)] then o else n/a
c33: if [S-FU(EXCEP)] then m else n/a

Status: OIW: Closed - Accepted 03/94
EWOS:
AOW:

(9) Presentation PPDUs

Summary: Discrepancy between ISO 8823-2 and ISP 10607-1 in the Context Identifier List parameter of the RS and RSA

Source: Lee Chastain, JITC (lee@huachuca-jitcosi.army.mil)

Date Raised: 6 January 1994

Issue: There appears to be a discrepancy between ISO 8823-2 and ISp 10607-1 in the Presentation Context Identifier List parameter of the RS and RSA PPDUs.

In the ISO (A.7.13/1 and A.7.14/1) the parameter is conditional, which can evaluate to mandatory. The ISP, however, claims the base standard has it as optional and for the profile makes it out of scope. Which is correct? Do we need (or already have) a defect report?

Responses: From Noel Albertson (noel@nsilsun1.Tredydev.Unisys.com):
It appears that the ISP is incorrect in stating the 'Presentation Context Identifier List' as optional in the base standard, since it would be required if the implementation supports the 'Context Restoration' functional unit. However, "outside of scope" may still be the correct status for the profile since (I believe) the functional unit is also outside the scope.

Solution: ISP 10607-1 table A.2 column D under resynchronization does have a typo, and should be conditional. Since the new version of the ISP does not have a column D, the typo has been fixed.

Status: OIW: Closed - Not an issue 03/94
EWOS:
AOW:

(10) Summary: Need for clarification on when Presentation Context negotiation is complete during association establishment using RTSE.

Source: Brenda Troisi, Tandem Computers (troisi_brenda@tandem.com)

Date Raised: 20 January 1994

Issue: The issue is concerned with whether the absence of direct reference from the EXTERNAL type of the RTSE APDU (user data of AARQApdu - RTORQApdu) in an association connect request should be considered as an error, or whether the indirect reference alone is sufficient.

The answer to the above question revolves around the related question: "When is presentation context negotiation complete?" This is the question raised to the OIW.

The current OIW response is as follows:
Presentation context negotiation is complete when the presentation "confirm" is complete, and the negotiated context result is conveyed to the Presentation user.
Note that the Presentation user may choose to reject the result.

Company A's interpretation:
Company A's RTSE implementation is a client of ACSE and sends its RTORQ-apdu in the ACSE AARQ user-information field. The RTSE implementation uses a single transfer syntax, BER, and performs the encoding/decoding of the EXTERNAL that will be passed as user-information in the AARQ PDU. Given the similarity of EXTERNAL and PDV-list, the author apparently concluded that the rukes for the PDV-list applied to the EXTERNAL as well (see Attchment 3). Therefore, the author did not include the direct-reference in that EXTERNAL.

Company B's interpretation:
Company B agrees that the standard contains no definitive description. They have apparently discounted the similarity between PDV-list and the EXTERNAL, decided that X.208 section 34.6 is the only relevant text, and that negotiation is complete when the responder issues the CPA or CPR PPDU. Thus negotiation is incomplete at the point that the P-CONNECT indication's user data is decoded and they want the EXTERNAL value to contain the direct reference field. Since Company A's RTORQ-apdus don't, they reject connect requests from Company A's MTA.

Implication of the OIW decision:
The OIW place the completion of negotiation after the point of time under discussion. In fact, it is later than either company chose. If they choose to recognize the authority of this group to make this ruling, then Company A's AARQ/CP PPDU is not in accordance with their decision and ought to change. In fact, from reading the OIW statement, both company's AARE/CPA PPDUs

could be considered wrong even though their implementations of these PDUs match that in all the implementations we could find to check.

Remaining ambiguity:
Both Company B and the OIW state that construction of the 3 Presentation Connection PDUs precede completion of presentation context negotiation. Each considers only one side of the connection (e.g. the responding side cannot know when the confirm is delivered). Both statements put together could be the correct answer - that is, the two parties to the connection each have a different view of when it is complete. But neither statement addressed the issue of encoding vs. decoding. It would seem most logical that the state of negotiation at the time of encoding should be the correct one to consider. Yet that is not what anyone does for the AARE!

ATTACHMENT 3: Details on the ambiguity

Relevant sections of the standards:
The problem occurs on association establishment. The application's connect (or bind) primitive becomes the content of the user-information field of the ACSE AARQ-apdu. This field is defined in X.227 as:

```
user-information IMPLICIT Association-information
```

and that data type is

```
Association-information ::= SEQUENCE OF EXTERNAL
```

the EXTERNAL data type is defined in X.208 section 34:

```
EXTERNAL ::= [UNIVERSAL 8] IMPLICIT SEQUENCE
{
        direct-reference OBJECT IDENTIFIER OPTIONAL,
        indirect-reference INTEGER OPTIONAL,
        data-value-descriptor Object Descriptor OPTIONAL,
        encoding CHOICE
        {
                single-ASN1-type [0] ANY,
                octet-aligned     [1] IMPLICIT OCTET STRING,
                arbitrary          [2] IMPLICIT BIT STRING
        }
}
```

X.208 section 34.6 explains that the presence of the direct-reference field in this type is conditional on the state of the presentation context negotiation (i.e. whether or not the negotiation is complete). Unfortunately, neither this text nor X.216/X.226 (Presentation specs) definitively state when negotiation is complete. In addition, the text does not state whose point of view is relevant. That is, do you decide what value to use based on the state of negotiation when the value is encoded or when it will be decoded?

The entire AARQ-apdu is then passed as user data in the P-Connect req. The P-Connect req. user data becomes the user data field of the CP PPDU and that is defined in X.226 as a choice of Simply-encoded -data or Fully-encoded-data. Fully-encoded-data is the case that applies to this discussion:

```
Fully-encoded-data ::= SEQUENCE OF PDV-list
PDV-list ::= SEQUENCE {
        Transfer-syntax-name OPTIONAL,
        Presentation-context-identifier,
        presentation-data-values ::= CHOICE {
                single-ASN.1-type  [0] ANY,
                octet-aligned      [1] IMPLICIT OCTET STRING,
                arbitrary          [2] IMPLICIT BIT STRING
        }
}
```

Section 8.4.2.7 states "The transfer-syntax-name component of a PDV-list value in a CP PPDU shall be present when more than one transfer syntax name was proposed for the presentation context of the presentation data values. The NIST/OIW Stable Implementor's Agreements on Upper Layers (see 12/92 edition, Volume 1, Chapter 5, section 8.6) goes on to say that the transfer syntax name shall be present IF AND ONLY IF more than one transfer syntax name was proposed. [Since the definition of the PDV-list and EXTERNAL fields look the same, and, in the case of an AARQ, an EXTERNAL is embedded in a PDV-list, one could extrapolate that this rule would apply to both data types. Does it? Should it?]

Implementation Status

A quick check was made against interworking protocol traces that we had handy to see what various (FTAM, X.500, X.400) implementations and conformance testers are doing in this area. For the AARQ/CP PPDU, 3 of the 4 possible combinations of absence/presence of the two transfer syntax components were found. Everyone implemented a single combination (no transfer syntax components) in the AARE/CPA PPDU BUT this particular combination is in contradiction of the OIW clarification in Appendix 1. That is, everyone unanimously decided to consider negotiation complete when encoding the EXTERNAL in the AARE rather than upon delivery of the P-Connect Conf.

Status in the Standard's World

These issues have long been known and discussed but no definitive action has been taken. According to a member of the ISO Presentation group, the similarity between the EXTERNAL and PDV-list data type has been discussed in the past and the (unrecorded) consensus appears to be that the rules to PDV-list do not apply to EXTERNAL despite their similarity. A defect report was filed against X.208/8824 in June 93 which questions the text in X.208 section 34.6. The implementor's groups seem to be the best available forum for addressing this issue.

Proposed Solution:

This is clearly a complex issue which is hard to understand and hard to explain. Simplification would enhance clarity. A simple solution would be to alter the context of X.208 section 34.6 to refer to the text ruling the construction of the Presentation PDV-list. And then to clarify that PDV-list description. There are two possibilities for the clarification of the PDV-list:

- Most unambiguous: the transfer syntax component is always (and only) included in any exchange (all 4 primitives and both PDUs - a list could be included for clarity) which can only modify the Defined Context Set (DCS). This DCS. This would be a departure from current practice and the OIW statement.
- Perhaps better: the transfer syntax component may always be included but is only used when necessary i.e. when the PCID does not unambiguously identify a transfer syntax because the DCS is incomplete and more than one transfer syntax was defined. This is closer to the current practice but perhaps harder to explain to anyone unfamiliar with the DCS.

In either case, the value of the field would not be checked when it was not used which would introduce a level of tolerance to achieve backward compatibility.

The solution should both clarify the issue AND accomodate existing implementations. Unfortunately, it is probably too late (both for this round of standardization and with respect to all the extant implementations) to achieve this level of clarification. So the ambiguities will remain and be a source of confusion and question amongst implementors (while researching this issue, I found 4 other companies who shared our confusion). Thus a recorded statement from the OIW acknowledging the ambiguities with some resolution would be helpful. At a minimum, implementors should be encouraged to make their protocol implementations "tolerant" in this murky area. I would define tolerance in the following way: "absence or presence of the transfer syntax component in the PDV-list or ACSE user data in primitives/PDUs which define or modify the DCS should not be considered a protocol violation unless the corresponding PCID does not resolve to a single transfer syntax at the required time.

Responses: A concensus at the June OIW was the following:

On the Negotiation question:
The transfer syntax rule is stated in the Stable Agreements Part 5 - 8.6.

After long discussions, talks with experts and more research, we were convinced that "prior agreement" means agreements between the implementors (through standards and implementors agreements). It does not mean actual negotiation between systems. Since the base standards and agreements can specify a single transfer syntax, use of the direct-reference component would not be appropriate.

On the question of EXTERNAL VS PDV-list:
These entities are different and should not be confused.

From: Laura Emmons (laurae@ar.telenex.com):
>When is Presentation layer negotiation complete?
According to the state tables in X.226 for the initiator of a connection, it is complete after the P-CONcnf is issued to the PS-user; for the responder of a connection, it is complete wheneither the CPA or CPR is sent.

>The issue is concerned with whether the absence of direct reference from the >EXTERNAL type of the RTSE APDU (user data of AARQApdu - RTORQApdu) in an >association connect request should be considered as an error, or whether the indirect >reference alone is sufficient.

I could not find an EXTERNAL data type in the RTORQApdu so I assume you mean the EXTERNAL component of the Association-information component of the AARQ-apdu. The use of the transfer syntax name described in 8.6 of Part 5 of the SIA is in reference to the PDV-list of the CP PPDU and has absolutely nothing to do with this issue. The critical text is in X.208 34.6. It states "If presentation layer negotiation is not complete, an object identifier value is also needed which identifies the encoding rules (transfer syntax) used for the encoding. Where presentation layer negotiation is in use, and where the direct-reference OBJECT IDENTIFIER element is allowed or is required to carry such a value, this shall be identified by a comment associated with the use of the EXTERNAL notation, otherwise the field shall be absent." It does not say that the object identifier value identifying the transfer syntax ***must*** be the direct reference component. Since there is no comment associated with the notation for the Association-information element in the ACSE, and since the transfer syntax was conveyed in the presentation context identifier list, the direct reference component of this particular EXTERNAL should not be present.

Solution: The direct reference component of the EXTERNAL in the Association-information element of both the AARQ and AARE APDUs should not be present. Its absence is not an error.

Status: OIW: Closed - Accepted 13 June 1994
EWOS:
AOW:

38

(11) Summary: Issue on Checkpoint Size and RTTR APDU

Source: Brenda Troisi, Tandem Computers (troisi_brenda@tandem.com)

Date Raised: 3 February 1994

Issue: About Checkpoint Size and RTTR APDU - 2/3/94

THE QUESTION:

RTTR APDUs are used to convey the X.400 P1 MPDU. If Checkpointing is used, then the MPDU will be conveyed in one or more RTTR APDUs the size of each of which is constrained by the Checkpoint size (if checkpointing is not in use, a single, unlimited size RTTR APDU is used). A question has arisen about the meaning of the checkpoint size value and its relationship to the RTTR APDU. Specifically, should that size include or exclude the encoding of the OCTET STRING. X.228 is not crystal clear on this point and the text could be interpreted in at least two different ways.

RELEVANT FACTS:

This issue came up during the 1988 study period and was addressed by the 1984 Implementor's Guide. The first paragraph of X.410 section 4.3.2 (Transmitting an APDU) is the relevant text and it was amended as indicated in the "[]" below (this clarification appeared as bullet E28 in the 1984 IG). Remember that in 1984, RTS was mapped directly to session, hence the RTS APDU is sent as an SSDU.

"The maximum SSDU size [(same as the maximum checkpoint size)], as stated previously, will have been negotiated during the connection phase. The sending RTS must submit in S-DATA.request, SSDUs that conform to that agreement."

In X.228, this text becomes section 8.2.1.2.1 (use of P-Data user data):

"The maximum User data size (number of octets of the RTTR APDU value) will have been negotiated during the association-establishment procedure. The sending RTPM shall submit User data that conforms to that agreement."

Question: why was this clarifying text not included in 1988? Was it considered to be self-evident or was an explict repudiation of this position intended? If the latter, then should that fact be noted in X.228 Annex B (differences between X.228 and X.410-1984)?

One problem here seems to be the interpretation of the "value" in this previous paragraph. One possibility is to take it literally and equate "value" with "content" (as in type/length/value -> identifier/length/content) of the BER encoding. Another possibility is to interpret it as the right hand side of the ASN.1 definition for RTTRapdu e.g. OCTET STRING.

As I sit here late tonite, I realize that I cannot find any mention of the OCTET STRING in the 1984 X.410 standard. Perhaps that is the key to this. If the OCTET STRING is new in 1988, then a position compatible with the 1984 IG bullet is to say that the 1984 content of the SSDU has now become the content of the 1988 OCTET STRING and therefore, the checkpoint size now applies to the content of the OCTET STRING. That is, don't count the OCTET STRING encoding in the checkpoint size. This raises a new question which is "how is 1984 compatibility achieved with that OCTET STRING in place?".

OTHER RELEVANT SECTIONS OF X.228:

The RTTR APDU is defined as:

RTTRapdu ::= OCTET STRING

The content of the OCTET STRING is all or a piece of the MPDU which is encapsulated in the OCTET STRING encoding. The RTTRapdu is then passed in the P-Data request to the other side.

section 7.3.2 (APDUs used in transfer procedure):

"The RTSE-user APDU value is transformed into the encoded-APDU-value and vice versa by means of the local syntax-matching services. The transfer procedure uses the RT-TRANSFER (RTTR) APDU. The transfer procedure supports segmenting and reassembling of the encoded-APDU-value into/from one or more RTTR APDUs.

An encoded-APDU-value is transferred as a single RTTR APDU if checkpointing is not used. Otherwise, the encoded-APDU-value is transferred as a series of RTTR APDUs, the maximum size (ie number of octets forming the RTTR APDU value) of each being the negotiated checkpoint-size. The concatenation of the RTTR APDU values is the encoded-APDU-value."

Again, there is the question of how literally to take the term "value" in the phrase "number of octets forming the RTTR APDU value". There is also room for interpretation of the last sentence. One can take the "concatenation" phrase literally or one can take it as a logical concatenation (e.g. strip the octet string and then concatenate). Another possible area of confusion is usage of the term APDU. When does it refer to the user data passed in the RTS primitive (i.e. the data that will be encapsulated in the RTS APDU) and when does it refer to the user data which will be passed in the presentation primitive (i.e. the encoded RTS APDU). It would seem most logical for it to always refer to the latter case since the former case would be be better referred to as UserData.

X.228 Sect 6.2.2 Use of Presentation-service p386

"The RTPM requires access to local syntax-matching-services provided by the presentation-service provider. This syntax-matching-service consists of:

a) an encoding service enabling the transformation from the local representation of an APDU value into an encoded-APDU-value of type OCTET-STRING the value of which is the representation of the APDU value specified by the negotiated transfer syntax;
b) a decoding service enabling the transformation from an encoded-APDU-value into the local representation of the APDU value.

If X.410-1984 mode or simple encoding is used by the Presentation Layer, the APDU value is encoded as ASN1.type ANY. If full encoding is used by the Presentation Layer, the APDU value is encoded as ASN.1 type EXTERNAL. (For X.410-1984 mode, single encoding and full encoding see Recommendation X.226)."

Note: None of these sections appear to distinguish between normal and X.410-1984 mode when dealing with checkpoint size. Thus the two cases must be dealt with in the same way. Given that X.410-1984 mode is to be bitwise compatible with a 1984 X.410 implemenation....

Responses: Concensus from the June OIW:
There is no room for interpreting the term "value" used above. Value is defined in X.208 12.7. The value of the RTTRapdu is the whole OCTET STRING value, i.e. identifier, length, and contents.

Solution: See above

Status: OIW: Closed - Accepted 13 June 1994
EWOS:
AOW:

(12) Summary: RTS (X.228) Ambiguity on CallingSSUseReference

Source: Charlie Combs, MCI

Date Raised: 11 March 1994

Issue:

Introduction

Implementations of the 1988 version of the X.400 series of CCITT Recommendations are becoming operational. It has been discovered through interworking testing that some implementations are not compatible with 1984-based implementations. The incompatibility is due to an ambiguity in the *Reliable Transfer: Protocol Specification* Recommendation X.228 (1988). This contribution identifies the interworking problem and proposes a solution. It is hoped that the OIW will agree on a solution at the March 1994 meeting.

Problem Description

Recommendation X.410 (1984) as amended by *X.400-Series Implementor's Guide*(version 5, clarification E17) states in section 4.2.1 (S-CONNECT) the following for the Session Connection Identifier:

"Note 1 - The initiating RTS will supply a Session Connection Identifier, which will be used to uniquely identify the connection. The identifier is formed of the following components: Calling SS-user Reference, Common Reference, and, optionally, Additional Reference Information. The identifier is returned unchanged by the responding RTS, except that the Calling SS-user Reference supplied by the initiator is conveyed as the Called SS-user Reference.

Each Component, when present, will contain a data element of the appropriately named type from the following definitions:

CallingSSUserReference ::= SSAPAddress -- of the initiator

CommonReference ::= UTCTime

AdditionalReferenceInformation :: T61String

The syntax of the SSAP address T.61 String in the **CallingSSUserReference** is not defined by this Recommendation, but is a local matter. This information shall only be used to compare two Session Connection Identifiers octet by octet. Note: this allows both X.409 encoding (Identifier, Length, and Contents) and Session Layer Protocol rules (Contents only) for the Session Connection Identifier."

Recommendation X.228 (1988) Annex B (Differences between this Recommendation and Recommendation X.410-1984) states the intent of the X.410-1984 mode is to support 1984-vintage implementations with the statement:

"In X.410-1984 mode this Recommendation and its use of ACSE and Presentation service is bit-compatible to Recommendation X.410-1984 under consideration of the clarifications and errata of the X.400-series Implementors Guide V.5."

Responses:

Solution: Therefore, it is concluded that Recommendation X.228 section 8.1.1.1.3.5 (Session connection identifier) is contradictory and should, as a minimum, be interpreted as:

```
CallingSSuserReference  ::= CHOICE{ --local matter, solely in X.410-1984
mode --
                        OCTET STRING --solely in normal mode --}
```

CommonReference ::= UTCTime

AdditionalReferenceInformation ::= T61String

Additionally, it needs to be made clear that, in the context of the **PConnect** and **PAccept** (in 1984, **RTORQapdu** and **RTOACapdu** in 1988), the present definition of **SessionConnectionIdentifier** is not altered by the above interpretations.

Defect Report on X.228 is being submitted.

Status: OIW: Closed - accepted
EWOS:
AOW: