

SUMMARY OF VOTING ON

Letter Ballot Reference No: SC22 N3612

Circulated by: JTC 1/SC22

Circulation Date: 2003-08-12

Closing Date: 2003-12-10

SUBJECT: Summary of Voting on SC 22 N3612, CD Registration and Approval Ballot for ISO/IEC CD 11404, General purpose datatypes (GPD)

The following responses have been received on the subject of registration:

"P" Members supporting registration without comments

13 (China, Czech Republic, Denmark, Germany, Ireland, Italy, Republic of Korea, Netherlands, Norway, Russian Federation, Ukraine, UK, USA)

"P" Members supporting registration with comments

-

"P" Members not supporting registration

1 (Japan)

"P" Members abstaining

1 (Switzerland)

"P" Members not voting

10 (Austria, Belgium, Brazil, Canada, Egypt, Finland, France, DPR of Korea, Romania, Slovenia)

_____ end of registration ballot, beginning of NB comments _____

Japan

Japan NB disapproves SC 22 N 3612 as a CD, since it is not mature to be a CD.

SUMMARY OF VOTING ON

Letter Ballot Reference No: SC22 N3612

Circulated by: JTC 1/SC22

Circulation Date: 2003-08-12

Closing Date: 2003-12-10

SUBJECT: Summary of Voting on SC 22 N3612, CD Registration and Approval Ballot for ISO/IEC CD 11404, General purpose datatypes (GPD)

The following responses have been received on the subject of approval:

"P" Members supporting approval without comments
10 (China, Czech Republic, Denmark, Ireland, Italy, Republic of Korea, Netherlands, Norway, Russian Federation, Ukraine)

"P" Members supporting approval, with comments

-

"P" Members not supporting approval
2 (Japan, USA)

"P" Members abstaining
1 (Switzerland)

"P" Members not voting
12 (Austria, Belgium, Brazil, Canada, Egypt, Finland, France, Germany, DPR of Korea, Romania, Slovenia, United Kingdom)

_____ end of approval ballot, beginning of NB comments _____

Japan

Japan NB disapproves SC 22 N 3612 as a CD.

It has a lot of newly added syntax rules and semantic descriptions to extend the scope of original Language Independent Datatypes. Many of those additions are technically incomplete: some of the non-terminals are left undefined having no syntax rule; some are left undefined of their semantics; and some examples do not follow the defined syntax rules. It is hard to start technical discussion on the intended General Purpose features of GPD unless it is filled up with necessary definitions and descriptions on those issues.

1. value domain

It is intended to be used for metadatas and value domains as described in "Introduction" and has a normative reference to ISO/IEC 11179-3. It actually tries to refer to "URI-text" and "URI-to-value-domain" in the syntax rules of 8.1.2 and 8.1.3.

Yet it has no syntax rules for those non-terminals nor any semantical descriptions. We have no way except guessing relation to ISO/IEC 11179-3 or other sources.

2. program text conformance

The notion of "program text conformance" is not clear from the description of 4.4. Since there is no items in "3. terms and definitions",

"program text" might refer to usual programs written in, say, Fortran, Cobol, or else. Can such programs conform to this standard?

Are they necessary to conform to this standard in any sense?

It defines the non-terminal "program-text" in 7.6, and

"program text conformance" might be on this "program-text".

Anyhow, it needs clear definition of "program text" which is intended to conform to the standard.

3. extensibility of datatypes and value spaces

As one of means to cope with this new feature, it introduces

"provisions associated with datatypes" (6.9).

This notion of "provisions" is not defined well.

It needs further refinement of syntax rules, semantic descriptions and examples.

EXAMPLE of 6.9 shows:

```
normative MDE = provision( ... ),
normative ODE = provision( ... ),
normative XDE = provision( ... ),
normative address_label_standard =
record XDE
(
  name MDE: characterstring,
  address MDE: characterstring,
  city MDE: characterstring,
  state_province MDE: characterstring,
  postal_code MDE: characterstring,
  country_code ODE: characterstring,
),
```

and says MDE, ODE and XDE are shorthands of each provision. But there is no rules allowing such shorthands introduction.

From the syntax rule for "program-text", each statement beginning with "normative" and terminated with a comma might be a declaration and specifically a

normative-datatype-declaration.

```
normative-datatype-declaration =  
  "normative", type-identifier,  
  [ "(" formal-type-parameter-list, ")" ],  
  "=", type-definition ;
```

Though the declaration for "address_label_standard" seems to obey the rule, those for MDE, ODE and XDE don't obey the rule since "provision(...)" is a provision-statement that is not a type-definition.

Moreover, the type-definition for "address_label_standard" does not obey the syntax rule of "record-type" which is an aggregate-type which is a generated-type which is a type-specifier.

```
record-type = "record", { provision-statement },  
             "(" field-list ")" ;  
field-list = field { "," field } ;  
field = field-identifier ":" field-type ;  
field-identifier = identifier ;  
field-type = type-specifier ;
```

And, as you see, there is no space to write "MDE" or "ODE" in front of ":"

in a field by the rules shown above. Also "XDE" cannot be written next to "record" since the rule allows only direct insertion of provision-statements.

4. object identifier

Objects in the sense of Open Systems Interconnection are treated in 10.1.10 referring to ASN 1, and "objectidentifier-value" is defined to be either "ASN-object-identifier" or "collection-identifier". It seems to say that a collection-identifier is such as "ISO_10646 2" and an ASN-object-identifier is such as { iso(1) standard(0) 10646 part1(1) 2 }, where "2" is a registry-index which refers to a specific character collection.

The description given below "Value syntax:" is not clear if it defines meaning of "collection-identifier" in terms of ASN 1. It should be refined to clarify the intention with appropriate examples. For example, take the paragraph starting with "The keyword ISO_10646".

The keyword ISO_10646 refers to the collections defined in Annex A of ISO/IEC 10646-1:1993 and the collection designated is that collection whose "collection-number" is the value of registry-index.

The form of the object identifier value is:

```
{ iso(1) standard(0) 10646 part1(1) registry-index }
```

There is no referent of "the collection designated" in this sentence. It might take it granted that the hidden subject of the sentence is a given collection-identifier starting with "ISO-10646" and that any collection-identifier represents a collection of characters, though there is no explicit description nor definition. Moreover, there is no referent of "registry-index" in the sentence. Of course, there is the syntax rule for registry-index, which simply says it is a number. What is really necessary here is the referent of "registry-index" is the number specified at the end of a given collection-identifier. It might have tried to indicate such intention by putting it in italic. Such convention could not explain itself to the reader of the standard without clear description and definition.

5. reference to other standards

In definition of "identifier" (7.3.1) and "digit-string" (7.3.2) appear non-terminals "ISO/IEC-10176-extended-letter", "ISO/IEC-10176-extended-digit", and "ISO/IEC-10176-extended-special", which have no definition in this standard.

They need their definition.

It is no excuse in omitting their definition that they have self-explanatory names.

Note that the standard cannot refer to ISO/IEC TR 10176 in normative way since it is a TR that may not constitute any provision of a standard.

6. miscellaneous items

There are a lot of miscellaneous items including editorial ones. They are listed below in the order of section numbers.

<< Introduction, Support for semi-structured and unstructured data aggregates >>

may be unknown in advance (e.g., a compilation time),
but may be discovered and process at runtime

The word "a" in "a compilation time" would be a typo.

The word "process" should be changed to "processed".

<< 1, paragraph after bullets >>

datatypes and provides a means by which datatypes not defined

We think that "a" of "a means" should be removed.

<< 2 >>

ISO/IEC 8825:2002, Information technology --- ASN.1 Encoding Rules
ISO/IEC 8825:2002, Data elements and interchange formats ---
Information interchange - Representation of dates and times

One of these two citations must be erroneous.

Many standards and technical reports should be added to the reference list here. For example, TR 10176, ISO 2375 and ISO 7350 are referred to in the text of this Standard, but are not cited in this section.

<< 3.10 >>

generator that distinguish this datatype generator from other datatype generators and produce identical value spaces from identical parametric

The word "and" in this sentence was "which" in the previous version of this Standard. We think that "which" is much better.

<< 3.23 >>

The term "GPD datatype" is "general purpose datatype datatype" in its full form. This looks quite strange to us.

<< 3.35 >>

A notation "<<EBNF>>" appears but its meaning is not explained anywhere.

<< 3.50 >>

This section should be placed before 3.49. The word "regular" precedes "representation" in alphabetical order.

<< 4.1, bullet 4 >>

4. to the extent that the entity provides operations other than movement or translation of values, define

What is the meaning of "translation"?

<< 5.1, Table 5-1 >>

The asterisk notation for repetition seems unused in this Standard. We think that the explanation of this notation should be removed.

<< 5.1, Example 2 >>

letter: A B C D E F G H I J etc.
vowel: A E I O U
consonant: B C D F G H J K L M etc.

The font of the word "etc." looks wrong.

<< 6.2 >>

A value space is the collection of values for a given datatype.

This definition is somewhat different from the one given in 3.62.

A value space contains regular values and may contain sentinel values.

Is the term "sentinel values" appropriate in this context? We usually use the term referring to terminating values of loops.

<< 6.4 >>

Aggregate datatypes may be:

We think that "may be" is too weak here.

- conceptually structured, having both designators (i.e., access methods) and datatypes known prior to use of the aggregate datatype, or
- conceptually semi-structured, have either designators and datatypes known prior to use of the aggregate datatype, or
- conceptually unstructured, having neither designators and datatypes known prior to use of the aggregate datatype.

The first and the third bullets use "having", while the second bullet uses "have". The word "and" in the second bullet should be "or".

<< 6.6 >>

The set of characterizing operations for a datatype comprises those operations on, or yielding values of, the

The position of a comma is not correct. The above sentence should read as follows:

... operations on, or yielding, values of the ...

This is consistent with 3.9 and 3.10.

<< 6.6, Note 3 >>

- n-adic operations which map ordered n-tuples of values, each of which is of a specified datatype, which may be the given datatype or a parametric datatype, into values of the given datatype or a parametric datatype.

It is easy to misunderstand that this bullet is saying about n-adic operations where $n \geq 3$, since operations with 0, 1 or 2 parameters are mentioned in the previous bullets. We suggest to insert a condition " $n \geq 1$ " in this bullet.

<< 6.6, Note 5 >>

NOTE 5 IsEqual is always a characterizing operation on datatypes with the equality property.

The operation name is not "IsEqual" but "Equal", according to 6.3.1 and many sections in 8. A reference to 6.3.1 should be given at the end of this Note.

<< 6.9, last sentence >>

Defined provisions for a datatype, that datatype can only be a datatype family

We cannot understand this sentence. Probably, it should be removed.

<< 7.1, rule 4 >>

equivalent: a string literal that contains the one character quotation mark.

The sequence "the one" is strange.

<< 7.2 >>

A sequence of one or more space characters, horizontal tabs, end of line characters, or newline characters except within a character-literal or string-literal (see 7.3), shall be considered whitespace. Any use of this International Standard may define any other characters or sequences of characters not in the above

character set to be whitespace as well, such as vertical tabulators, end of page indicators, etc..

There are several inconsistencies, "horizontal tabs" vs "vertical tabulators", and "end of line characters" vs "end of page indicators".

- Any sequence of characters beginning with the sequence // (solidus, solidus) and terminating with the occurrence thereafter of a newline character sequence.

In the first sentence of this section, the term "end of line characters" is used instead of "end of line character sequence".

<< 7.3.1 >>

pseudo-letter-like = letter |
digit |
underscore ;

The first and the second alternative should be "letter-like" and "digit-like" we suppose. Otherwise, we can use extended letters as the first character of an identifier, but cannot as the second character. This is quite strange.

2. The identifier X in a component of a type-specifier (Y) refers to that

The parentheses around "Y" would easily be misunderstood as syntactical notation. We suggest to insert some words, e.g. "say", before "Y".

<< 8.1.2 >>

list-source-reference = identifier |
"", URI-text, "" ;

The term "URI-text" is not defined anywhere. The sequence ("") before and after "URI-text" should be (").

<< 8.1.3 >>

enumerated-value = enumerated-value-list |
URI-to-value-domain ;

The term "URI-to-value-domain" is not defined anywhere. Isn't it identical to "value-domain-source" defined in the previous section?

Equal(x, y: enumerated(enum-value-list)): boolean is true if x and y designate the same value in the enum-value-list, and false otherwise.

The font of "enum-value-list" in "in the enum-value-list" looks wrong.

<< 8.1.4, Note 5 >>

NOTE 5 The value space of a Character datatype is the character set, not the character codes, as those terms are

We understand that the name of a type should not be capitalized.

scope of this International Standard. Many uses of this International Standard , however, may require the association to

The space between "International Standard" and a comma should be removed.

<< 8.1.6 >>

radix(-factor) of the specified time-unit. time-unit, and radix and factor

This sentence seems to suggest subtraction of "factor" from "radix". It should be exponentiation, we think.

shall conform to ISO 8601:2000, Representation of dates and times.

In section 2, the same standard is referred to as "ISO/IEC 8601:2000". Probably, "IEC" should be inserted here.

InOrder(x, y: time(time-unit, radix, factor)): boolean is true if the point in time designated by x precedes

In general, the operation "InOrder" should mean "less than or equal". Here it seems to suggest "less than" operation, since the word "precedes" is used.

<< 8.1.10 >>

arithmetic - Part 1: Integer and real arithmetic. IEC 559:1988

Now, the correct standard number is 60559, not 559.

<< 8.4.2 >>

```
member = { "override" }, member-identifier, ":", member-type ;
```

In section 8.2.7, a non-terminal "override-qualifier" is defined. We do not understand why this non-terminal is not used here.

<< 8.4.6, Note 6 >>

```
where the non-existent sizen+1 is taken to be 1. And the  
Ord(x1, ..., xn)th
```

The font of "th" after "Ord(x1, ..., xn)" looks wrong.

<< 8.6 >>

```
provision-statement = "provision", "(", actual-param-list, ")" ;
```

The non-terminal "actual-param-list" is not defined anywhere. We suppose that the sections 8.6.1-8.6.4 give concrete forms of "actual-param-list" but this fact should explicitly be stated.

We think that the name "actual-param-list" is not appropriate. It should be "actual-parameter-list".

<< 9.4 >>

```
normative-datatype-declaration =  
  "normative", type-identifier,  
  [ "(" formal-type-parameter-list, ")" ],  
  "=", type-definition ;
```

The non-terminal "normative-datatype-declaration" is not referred to from any syntax rules. We suppose that it should be one of alternatives of "declaration" in section 9, or of "type-declaration" in 9.1.

<< 9.5.1 >>

```
import-type = "import", URI-or-type-identifier,  
  { "including" "(" select-list ")" |  
  "excluding" "(" select-list ")" } ;
```

The space before the closing parenthesis in the last line should be moved before the quote mark.

```
presented as source text of the datatype specification. If the  
including keyword is used, then only those elements in the source.
```

There are no verbs in this sentence.

<< 10.1.2 >>

Values: all Integer values v such that $0 \leq v$ and $v \leq \text{modulus}$.

Why is the value "modulus" included? We understand that the range is $0 .. \text{modulus}-1$. We suggest to write " $0 \leq v$ and $v < \text{modulus}$ ".

Properties: ordered, exact, numeric.

We think that the property "bounded" should also be included.

<< 10.1.4 >>

Properties: unordered, exact, non-numeric.

In 10.1.5, for "character string" type, the property "denumerable" is added to these. We think that the type "bit string" should also have this property.

<< 10.1.10 >>

Abstract Syntax Notation One (ISO/IEC 8824:1990).

In section 2, the year is "2002", not "1990".

nameandnumberform = identifier "(" numberform ")" ;
collection-identifier = registry-name registry-index ;

These rules use spaces as symbol concatenation marks. According to the definition of the syntax, commas should be used instead.

denoted by an ASN-object-identifier is that prescribed by ISO/IEC 8824:1990 Abstract Syntax Notation One.

The year should be "2002".

The keyword ISO_10646 refers to the collections defined in Annex A of ISO/IEC 10646-1:1993 and the

The year should be "2000".

<< 10.2 >>

When the generator generates an aggregate datatype, the aggregate properties described in clause 6.8

This paragraph should not be indented.

<< 10.2.1 >>

We see many inconsistencies in the usage of fonts in these sections. For example,

Properties: non-numeric, unordered, exact if and only if the element datatype is exact.

The operations "non-numeric", "unordered" and "exact" should be in bold.

<< 10.2.2 >>

Components: leaf shall be any datatype.

In 10.2.1, the word "may" is used instead of "shall".

Values: all finite recursive sequences in which every value is either a value of the leaf datatype, or a (sub-)tree itself.

Ultimately, every "terminal" value is of the leaf datatype.

The position of right parenthesis (beginning of a line) is strange.

<< 10.2.3 >>

Properties: ordered, exact, non-numeric.

The property "bounded" should be added.

<< Annex A >>

12 Annex A (informative): Character-set standards

We do not understand why this section has two numbers "12" and "Annex A". We understand that this section should not be numbered "12".

ISO/IEC 4873:1991 Information technology \ ISO 8-bit code for information interchange - Structure and rules for implementation

There is an empty line between the first and the second line of a reference. The same comment also applies to the 11 references which follow this one.

ISO 6861: - Information and documentation - Cyrillic alphabet coded character sets for historic

The year of the standard should be given. The same comment also applies to the 7 references which follow this one.

<< Annex D >>

This Annex is totally useless. It is the syntax summary of 1996 version of this Standard, and gives no useful information about this version.

USA

The U.S. DISAPPROVES the CD Consideration ballot. The US will change its vote to approve following successful resolution of the comments.

Template for comments and secretariat observations

Date: [2003-12-10](#)

Document: [US Comments on CD 11404](#)

1	2	(3)	4	5	(6)	(7)
MB ¹	Clause No./ Subclause No./ Annex (e.g. 3.1)	Paragraph/ Figure/Table/ Note (e.g. Table 1)	Type of comment ²	Comment (justification for change) by the MB	Proposed change by the MB	Secretariat observations on each comment submitted
USA-1	Clause 1, 6.9, 8.6.2.1, 8.6.4, Annex B and Annex C.		Major tech	<p>In the 1996 standard, implementation concerns are clearly relegated to the informative Annexes B and C. But this draft is confused about whether implementation concerns are normative aspects of datatypes or non-normative aspects. Implementation concerns are stated to be out-of-scope in Clause 1, but addressed normatively in several places.</p> <p>For example, 8.6.4 is a set of runtime annotations that Clause 1 says are out-of-scope for 11404.</p> <p>And the ordering of the components in an aggregate is identified in Annex C.7 as an implementation notion suitable for annotation. But 8.6.2 appears to specify a normative form of the same concept.</p> <p>The "placement recommendations" in Annex B seem to provide the same notion as the "scope" tag in 8.6.2, and the occurrence of provision-statement in the type declaration in 9.1 coincides with the directions in B.1.</p>	<p>Clarify the relationship of implementation notions to the <i>scope</i> of 11404.</p> <p>If necessary, repair Clause 1 and 6 to incorporate discussion of implementation properties as in-scope.</p> <p>Align 8.6 with Annex B and C, or delete and replace one of them.</p>	
USA-2	8.3.3	parameter-type	Major tech	<p>parameter-type = type-specifier is too restrictive. For many active programming languages, the forms of the "parameter type" for a formal parameter include more generalized specifiers that constrain the allowable datatypes to which the actual value can belong, such as: any-type, any-class, any-collection, any-collection of <parameter-type>, pointer to <parameter-type>, and possibly others.</p> <p>To support interface definitions, ISO 11404 must support these more generalized concepts in parameter-types.</p> <p>We need to have an understanding of how formal parameters can constrain the datatype/value-space of the actual parameter, and the U.S. will contribute a document in that area.</p>	<p>Provide support for "generalized"parameter-types, including at least any-type, any-class, any-collection, optional/default parameters to procedures, and optional elements/fields of records and arrays.</p> <p>The standard shall document some model of these concepts, provide corresponding syntactic elements, and define them to support the "generalized specifier" model.</p> <p>The semi-structured types described in 6.4 appear to be an "any-collection". The standard should define them with this approach.</p> <p>The standard shall also specify the effects of these specifications on mappings, similar to the specifications for support of datatype properties in 11.4</p>	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

Template for comments and secretariat observations

Date: [2003-12-10](#)

Document: [US Comments on CD 11404](#)

1	2	(3)	4	5	(6)	(7)
MB ¹	Clause No./ Subclause No./ Annex (e.g. 3.1)	Paragraph/ Figure/Table/ Note (e.g. Table 1)	Type of comment ²	Comment (justification for change) by the MB	Proposed change by the MB	Secretariat observations on each comment submitted
					There is clearly some relationship between this concept and the "normative datatype" concept in 6.9. That relationship should become clear when 8.6 is clarified (see USA-4).	
USA-3	8.4.2	1st para	Major tech	<p>This is an erroneous definition of class. It makes "class" a synonym for "record", with superficial changes to the syntax, but no significant difference in the computational model.</p> <p>The standard should define "class" in terms of a common OO conceptualization. For example, consider the definition of "class" in UML (ISO 19501-1) or the definition of "entity" in EXPRESS (ISO 10303-11).</p>	<p>The standard should define "class" in terms of an interface to an object.</p> <p>The characterizing operations should be consistent with the nature of classes and objects.</p> <p>At the end of the Components section, define the meaning of the keyword override. Suggested:</p> <p>"The keyword override shall not appear unless the class is being defined as an explicit subtype (see 8.2.5). The member-identifier following the keyword override shall be the identifier for a member of the base datatype for the explicit subtype. The member-type following the keyword override shall designate a subtype of the member-type that was declared for that member of the base datatype."</p> <p>Replace the Values section with:</p> <p>"Values: The value space is that of an unspecified state datatype, each of whose values denotes a single 'object' that supports the interface represented by the class. The values of a class datatype are atomic."</p> <p>Replace the Value-syntax section with:</p> <p>"Value-syntax: none. In general, values of a class datatype have no external representation."</p> <p>Delete the Aggregate-properties section.</p> <p>Revise the Subtypes section to read:</p> <p>"Subtypes: any class datatype whose members</p>	

¹ **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

² **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

Template for comments and secretariat observations

Date: [2003-12-10](#) Document: [US Comments on CD 11404](#)

1	2	(3)	4	5	(6)	(7)
MB ¹	Clause No./ Subclause No./ Annex (e.g. 3.1)	Paragraph/ Figure/Table/ Note (e.g. Table 1)	Type of comment ²	Comment (justification for change) by the MB	Proposed change by the MB	Secretariat observations on each comment submitted
					<p>include one member c corresponding to each member of the base datatype, such that:</p> <ul style="list-style-type: none"> - the member-identifier for the subtype member is identical to the member-identifier for the member of the base datatype, and - the member datatype of the member of the subtype is the same as, or is a subtype of, the member datatype of the member of the base datatype." <p>Under Operations:</p> <p>Strike "Aggregate".</p> <p>Strike the definitions of MemberReplace, MemberFunctionInvoke, and MemberFunctionOverride.</p> <p>Replace the Note at the bottom, with:</p> <p>Note 1 – Class models the object-oriented programming language concept with the same name.</p> <p>Note 2 – The characterization of class that distinguishes it from Pointer to Record, which is the typical implementation of Class, is the characterization of the allowable subtypes. A subtype of a Class datatype models the object oriented notion of "subtype" or "subclass". A subtype of a Class datatype can have <i>additional</i> attributes (members); a subtype of a Record cannot.</p> <p>Note 3 – An operation is represented by a member whose member-type is a procedure datatype. Invoking an operation associated with a value of a class datatype can be derived from the characterizing operations as:</p>	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

Template for comments and secretariat observations

Date: [2003-12-10](#)

Document: [US Comments on CD 11404](#)

1	2	(3)	4	5	(6)	(7)
MB ¹	Clause No./ Subclause No./ Annex (e.g. 3.1)	Paragraph/ Figure/Table/ Note (e.g. Table 1)	Type of com- ment ²	Comment (justification for change) by the MB	Proposed change by the MB	Secretariat observations on each comment submitted
					Invoke(MemberSelect(...))	
USA-4	8.6	All	Major tech	<p>Subclause 8.6 defines an annotation language apparently intended to permit specification of a large number of things, but the purpose and scope of that set of things is not stated. Those things must be sorted out and identified, and then a syntax consisting of a dozen well-defined "type-attributes", each of which specifies one of those things, should be proposed. The syntax given is far too general, with the consequence that most conforming utterances are extremely difficult to make sense of, and it is nearly impossible to reconcile it with the stated interpretations of the examples, which reflect the true intent.</p> <p>The example in 8.6.1 says that the syntax</p> <pre>record provision(obligation=permit, target=type, scope=recursiveidentifier, subset=defined) (...)</pre> <p>is interpreted as: "every data element in all aggregates is optional". However, according to the interpretation of the individual components in 8.6.x, we get:</p> <p>Obligation=permit means: "the provision is an optional requirement, i.e. 'may'".</p> <p>Target=type means it the provision applies to the datatype</p> <p>Scope=recursiveidentifier means it applies to all the 'identifiers', at any depth, in the datatype (definition)</p> <p>Subset=defined means that it applies to all the <something> (identifiers?) that are defined.</p> <p>So, per 8.6.x, the provision is interpreted: "All defined identifiers that appear anywhere in this record definition are optional (requirements)."</p> <p>The language never refers to "data elements" or "components", and it never says what is meant by an</p>	<p>Define the kinds of constraints and permissions that this language is intended to specify:</p> <ul style="list-style-type: none"> - optional fields - extensible records - extensible choices - extensible enumerations - constraints on array/list/set sizes - constraints on parametric values , etc. (It is not clear from 8.6 what all this list should include.) <p>Determine which of these are not possible (or not adequately expressible) with existing features. Then define a syntax consisting of a set of well-defined type-attributes, each of allows specification of one such constraint or permission, and define the explicit interpretation of each such type-attribute.</p>	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

Template for comments and secretariat observations

Date: [2003-12-10](#)

Document: [US Comments on CD 11404](#)

1	2	(3)	4	5	(6)	(7)
MB ¹	Clause No./ Subclause No./ Annex (e.g. 3.1)	Paragraph/ Figure/Table/ Note (e.g. Table 1)	Type of comment ²	Comment (justification for change) by the MB	Proposed change by the MB	Secretariat observations on each comment submitted
				<p>identifier being an 'optional requirement'. So the relationship to the interpretation stated in the text is obscure.</p> <p>In a similar way, reading the interpretation of the syntax elements of Example 2, gives: "All undefined identifiers that appear anywhere in this record definition are optional." But the text says it means: "this datatype may be extended with additional data elements." Who would have guessed?</p> <p>And example 3 indicates that a very powerful but redundant capability is being embedded in this syntax:</p> <pre>provision(obligation=require, target=type, scope=size, value=range(limit..*))</pre> <p>Which says per 8.6.1.5 that "a value of the size of the type in the range (limit..*) is required." Now, of course, the size of the type can be specified by the range-subtype constraint defined in 8.2. What is intended by the syntax in the example is that maxsize is a parameter (not a named constant) and its value is required to be at least 17, but that can be specified by defining the parametric type generator and typing the maxsize parameter properly. So this mechanism appears to be redundant.</p>		
USA-5	Clause 8,10		ge	Provide examples of all datatypes and generators		
USA-6	Title, many		ed		The term "general purpose" should be spelled "general-purpose", per the NODE and Webster.	
USA-7	Syntax, all		ed	<p>The font in which the syntax is written is very difficult to read. Is it mandated by ISO directives or JTC1 conventions? If not, is there some more readable choice?</p> <p>Observation: For use as a (Web) resource, the syntax rules should be tagged with a distinct paragraph style, so that presentation can be modified by XSSL style rules.</p>		

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

Template for comments and secretariat observations

Date: [2003-12-10](#)

Document: [US Comments on CD 11404](#)

1	2	(3)	4	5	(6)	(7)
MB ¹	Clause No./ Subclause No./ Annex (e.g. 3.1)	Paragraph/ Figure/Table/ Note (e.g. Table 1)	Type of comment ²	Comment (justification for change) by the MB	Proposed change by the MB	Secretariat observations on each comment submitted
USA-8	Introduction	p. x	ed	What is the intended relationship between the draft introduction and the 1996 introduction?		
USA-9	Introduction	p. x	te	The discussion of concepts that are in scope or out-of-scope belongs in clause 1, not in the Introduction.	Move the discussion of versioning, implementation options, namespaces, data representation, etc., into Clause 1, with clear indication of which are in-scope and which are out-of-scope. Where appropriate, provide forward references to the subclauses that support the in-scope features.	
USA-10	Clause 2		ed	ISO 8825 is nowhere referred to in the normative text.	Delete ISO 8825 (both parts) from the normative references	
USA-11	Clause 2		ed	ISO 11179 is nowhere referred to in the normative text.	Delete the normative references to ISO 11179	
USA-12	3.4		ed	The Note cannot be correct if versioning is supported.	Delete or clarify the Note.	
USA-13	3.11		ed	The Note divides the definition.	Put the note text back into the definition	
USA-14	3.13, Clause 8		ed	"declaration" is not a verb. This usage appears only at the beginning of clause 8	Delete 3.13. Revise "specification of the means of datatype declaration" in Clause 8 (p.29) to read "specification of the means of declaring datatypes".	
USA-15	3.26		ed	"generator declaration" is never used in the text in this sense	Delete 3.26, but merge the definitive text with 3.27.	
USA-16	3.28, 3.47, 6.9		ed	The term "normative datatype" is not defined in clause 3, and it is used in a narrower sense than one might expect. The definition of "provision" is the ISO definition, but it is used in this standard in a narrower sense. The definition of "instruction" is essentially the first definition in the New Oxford Dictionary of English. Because the terms "normative" and "provision" may be used in this standard (it is itself a "normative document") with the ISO meanings, defining narrower meanings for	Define "normative datatype" in clause 3. Prefer a less confusing term. The narrower sense of "provision" used in this standard should replace the definition in 3.47. Prefer a less confusing term. If "instruction" is used in a narrower sense in specifying features of datatypes, that narrower definition should appear in 3.28. Otherwise delete 3.28.	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

Template for comments and secretariat observations

Date: [2003-12-10](#)

Document: [US Comments on CD 11404](#)

1	2	(3)	4	5	(6)	(7)
MB ¹	Clause No./ Subclause No./ Annex (e.g. 3.1)	Paragraph/ Figure/Table/ Note (e.g. Table 1)	Type of com- ment ²	Comment (justification for change) by the MB	Proposed change by the MB	Secretariat observations on each comment submitted
				those terms is undesirable. The IDN is not bound by the practices of an existing language in this area. Could we find a less confusing terminology?		
USA-17	3.51 and 3.54		ed	"subject to" is a curious term.	In 3.51, replace "subject to" by "consistent with". In 3.54, replace "not subject to" by "not completely consistent with"	
USA-18	4.4 (and 7.6)		te	The content of 4.4 appears to be entirely covered by 4.1, unless for some reason "program text" is not (part of) a directly conforming entity. Program-text is just the highest-level syntactic object. Is a program-text intended to be a "normative document" in the sense of 3.37?	State in 4.4 that program-text is a special case of directly conforming entity and clarify the relationship to clause 4.1.	
USA-19	6.1	para 2, 2nd sentence	ed	"The term GPD datatype refers to ..." should read: The term <i>GPD datatype</i> refers to datatypes generated or defined by means specified in this International Standard.		
USA-20	6.2 and 6.6, 6.7		ed	"A distinct value may belong to the value spaces of many datatypes, such as subtypes of those datatype (see 8.2)." The second occurrence of "datatype" should be plural. But subtypes are a special case, in that they are defined to preserve the characterizing operations.	Replace with: "A distinct value may belong to the value space of more than one datatype, so long as it properly supports the properties and characterizing operations of each of them (see 6.6)."	
USA-21	6.2		ed	There is some confusion about the level of abstraction in ISO 11404 and its relationship to ISO/IEC 11179 Metadata. Provide a Note describing the relationships of value space and datatype to "value domain" in ISO/IEC 11179.	After the first paragraph of 6.2 add a Note, such as: "Note -- This International Standard defines the concept "value space", which is just a set of values. It extends that notion to "datatype" by adding computational properties supported by characterizing operations. ISO/IEC 11179 [full title] introduces the concept "value domain". A "value domain" is a set of (value, meaning) pairs, each pair consisting of a value and its conceptual interpretation.. That is, ISO/IEC 11179 extends the notion value space to "value domain" by adding its	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

Template for comments and secretariat observations

Date: [2003-12-10](#)

Document: [US Comments on CD 11404](#)

1	2	(3)	4	5	(6)	(7)
MB ¹	Clause No./ Subclause No./ Annex (e.g. 3.1)	Paragraph/ Figure/Table/ Note (e.g. Table 1)	Type of comment ²	Comment (justification for change) by the MB	Proposed change by the MB	Secretariat observations on each comment submitted
					meaning to users and applications."	
USA-22	6.2	para ?	te	<p>" A value space contains <i>regular values</i> and may contain <i>sentinel values</i>. The properties and characterizing operations of a datatype shall apply to regular values. The properties and characterizing operations may apply to sentinel values."</p> <p>The words 'shall' and 'may' are inappropriate in defining terms. This paragraph should introduce the definitions of these terms that are given in clause 3.</p>	<p>Replace this paragraph with sentences that state the definitions in 3.51 and 3.54.</p>	
USA-23	6.4	para 3	te	<p>"Aggregates may be ... (bullet list)".</p> <p>Editorially, this text is out of place at this point – the concept it is trying to convey belongs in 6.8.</p> <p>But this text also needs to be simplified and clarified, consigning related concepts to Notes .</p> <p>The term "designator" is not defined at this point, it is not in clause 3, and, as used in 11404:1996 is a syntactic concept that has nothing to do with conceptual notions of data type. If "access method" is meant, that term should be used, and the reference to 6.8.6 should appear.</p> <p>The concept "known prior to use" is meaningless. The intent appears to be the "introspection" concept. "Introspection" is about examining metadata for an object in order to determine what its characterizing operations are, or what its datatype <i>is</i>. It is about (program/schema) declarations for the "datatype" of an object when that information is unknown, as distinct from being about the datatype of the object. And all of those concepts generalize to arbitrary datatypes, not just aggregates.</p> <p>See also USA -2</p>	<p>Delete this paragraph, the bullets, and the note, from 6.4.</p> <p>Add a new section to 6.8:</p> <p>6.8.8 Structured and unstructured</p> <p>Aggregate datatypes may be:</p> <ul style="list-style-type: none"> - conceptually structured, having both the component datatypes and the access method specified, or - conceptually semi-structured, have either the component datatypes or the access method specified, but not both, or - conceptually unstructured, having neither the component datatype nor the access method specified. 	
USA-24	6.4	Paragraph 3	te	<p>There is no syntactic feature or text in the subclauses of 8.4 that explicitly supports the semi-structured or unstructured property for aggregate type generators.</p>	<p>Add text to the appropriate subclause of 8.4 (if any) or 8.6 (if any) to refer to the semi-structured property explicitly and include the cross-reference</p>	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

Template for comments and secretariat observations

Date: [2003-12-10](#)

Document: [US Comments on CD 11404](#)

1	2	(3)	4	5	(6)	(7)
MB ¹	Clause No./ Subclause No./ Annex (e.g. 3.1)	Paragraph/ Figure/Table/ Note (e.g. Table 1)	Type of comment ²	Comment (justification for change) by the MB	Proposed change by the MB	Secretariat observations on each comment submitted
				How is this feature/property associated with a datatype? See also USA -2 and USA -4	to 6.4	
USA-25	6.8	last para	ed	the reference to 6.8.5 should be to 6.8.6.		
USA-26	6.9 (and 9.4)	Paragraphs 1, 2, 3	te	<p>"A datatype that includes provisions for a datatype is called a normative datatype."</p> <p>With the definition of "provision" given in paragraph 1 and the observations made in paragraph 2, it follows that every datatype defined in this standard is a normative datatype. Clause 8 states provisions for all of them.</p> <p>The critical idea here is almost captured in: "It is not possible to instantiate a normative datatype directly, but it is possible to instantiate an implementation (of the normative datatype) that conforms to the normative datatype."</p>	<p>Change the definition of normative datatype to read:</p> <p>"A <i>normative datatype</i> is a collection of specifications for datatype properties that may be simultaneously satisfied by more than one actual datatype. And there is a related concept of conformance to a normative datatype: A datatype <i>conforms to</i> a normative datatype if it satisfies all of the properties specified by the normative datatype.</p> <p>That is, a normative datatype does not have a specific value space, but it may specify properties that any conforming value space must have. Similarly, a normative datatype may specify operations that must be supported by a conforming datatype, without that set of operations itself being sufficient to characterize any one datatype.</p> <p>For example, the normative datatype Any can be satisfied by any GPD datatype, with any value space. The only requirement is that Equal is defined on the value space."</p>	
USA-27	6.9	All	te	<p>What properties of data types are being defined here?</p> <p>The text of this subclause does not define any datatype property.</p> <p>What is intended here is that some other kind of provision will be defined by a "type-attribute" beginning with the keyword "provision". And that kind of provision will not be ordering, or numeric, or access-method, etc., because</p>	<p>Replace this entire subclause with one or more subclauses that describe the datatype properties or the kinds of requirements a (syntactic) "provision" will state.</p> <p>In particular, see USA -2, USA-28 and USA -76</p>	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

Template for comments and secretariat observations

Date: [2003-12-10](#)

Document: [US Comments on CD 11404](#)

1	2	(3)	4	5	(6)	(7)
MB ¹	Clause No./ Subclause No./ Annex (e.g. 3.1)	Paragraph/ Figure/Table/ Note (e.g. Table 1)	Type of comment ²	Comment (justification for change) by the MB	Proposed change by the MB	Secretariat observations on each comment submitted
				provisions of that kind are discussed in 6.x above and have other syntax. But these additional "provisions" will also provide for specific properties of datatypes, values, and mappings. Those properties should be described here. Unfortunately, there is no normative text here that describes the kind(s) of properties that are intended.		
USA-28	6.9		te	<p>It appears, from the examples only, that one datatype property being identified is mandatory vs. optional features of datatypes and value spaces. From the examples, that property applies only to components of aggregate datatypes. Is this property the same one as the "semi-structured" property described in 6.4?</p> <p>If not, this property should be described here as a self-standing datatype property.</p>	<p>Add a subclause to 6.8:</p> <p>"6.x Mandatory and Optional components.</p> <p>The components of an aggregate datatype may not all be required to have values in a valid value of the datatype. That is, the actual value space of the datatype may include values for which some of the component values are unspecified.</p> <p>When a component of the datatype is required to have a valid value in order for the aggregate value to be a valid value of the datatype, the component is said to be a <i>mandatory</i> component.</p> <p>When a component of the datatype is not required to have a valid value in order for the aggregate value to be a valid value of the datatype, the component is said to be an <i>optional</i> component.</p> <p>Note -- this property applies to fields of records, members of classes, and elements of sequences, tables, and arrays.</p> <p>Examples ..."</p> <p>3. Add a subclause to 11.4 to define support for the mandatory/optional feature.</p> <p>4. If this is the same concept as "structured/semi-structured", merge this text with the text on that feature that is in 6.4.</p>	
USA-29	6.9		Ed	That "normative" properties are stated in "provision"	Move all discussion of the type-attribute	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

Template for comments and secretariat observations

Date: [2003-12-10](#)

Document: [US Comments on CD 11404](#)

1	2	(3)	4	5	(6)	(7)
MB ¹	Clause No./ Subclause No./ Annex (e.g. 3.1)	Paragraph/ Figure/Table/ Note (e.g. Table 1)	Type of comment ²	Comment (justification for change) by the MB	Proposed change by the MB	Secretariat observations on each comment submitted
				attributes is a syntactic specification of the feature and it and the syntax examples should be in clause 8.6 or 9.4.	"provision", the IsConforming operation, and the examples to 8.6 or 9.4.1.	
USA-30						
USA-31	Clause 7	1st para	Q	is the second sentence still true? What is the relationship between the IDN language defined in this version of ISO 11404 and that of ISO 13886?	Correct the second sentence to specify the true relationship to ISO 13886.	
USA-32	7.1	bullet 4	ed		Modify the text of bullet 4 " An added-character may be referenced by name, by 8-digit short UCS identifier, or by 4-digit short UCS identifier, as specified by ISO 10646-1." to read: "An added character may be denoted by one of the notations specified in ISO 10646-1 – name, 8-digit short UCS identifier, or 4-digit short UCS identifier – surrounded by escape characters."	
USA-33	7.6		Q	Does a program-text have any meaning other than being a syntactic collection? It appears to have no delimiters and no identifiers.		
USA-34	7.6		te	The meaning of the occurrence of a provision-statement in a program-text, outside of any other declaration, is completely undefined, and most of the applicability attributes could not possibly apply. Provision-statement here may have been confused with the normative declaration.	Either state what a provision-statement defines, e.g. "defaults" for other declarations, or delete it from the production for program-text.	
USA-35	8.1	Operators	ed	In clause 8.1, and throughout clause 8, no font distinction is made between the explicit operation names, formal parameters, data types, etc. and the parametric ones. For example, in 8.1.2, in the definition of Equal, "state-value-list" is parametric, all of the other elements of the signature are explicit. (11404:1996 apparently italicizes the parametric ones, but never documents that.) Also in the description of this notation in 8.1, the		

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

Template for comments and secretariat observations

Date: [2003-12-10](#) Document: [US Comments on CD 11404](#)

1	2	(3)	4	5	(6)	(7)
MB ¹	Clause No./ Subclause No./ Annex (e.g. 3.1)	Paragraph/ Figure/Table/ Note (e.g. Table 1)	Type of comment ²	Comment (justification for change) by the MB	Proposed change by the MB	Secretariat observations on each comment submitted
				"syntactic" font is used for the elements of this "form", but the syntactic font is (properly) not used in specifying the "form" itself. (This whole section is a presentation problem.)		
USA-36	8.1.2	Parametric Values	Q	<p>If a value of a State/Enumerated type appears as a value in the definition of another State/Enumerated type, is it the same value? Can it be the same value?</p> <p>In PCTE, it is the same value, and the State type is a space composed of self-standing values. But in some programming languages, the value is implicitly qualified by the type, so that no two such values are the same.</p> <p>If the value space of a state/enumerated datatype is specified as set of words in English, because the originators happen to be English speakers, do those words get locked in to the values of the type, or are they translated along with the text of the specification into other languages?</p> <p>One view: The names used in the IDN are appellatives for the distinguished values of state/enumerated datatypes, but only in the IDN language. The notations for values in the IDN in general are carefully stated to be there solely to satisfy the need for references to specific values in datatype definitions. So the names for values that appear in an IDN State type are not the values themselves; they are just names for the values. E.g., 'stop' as a value of State(stop, go) means "the value called 'stop' in this program-text", and any interpretation beyond that is out of scope for ISO 11404. What is required for any mapping/implementation is that there is a 1-to-1 mapping for each value, and the IDN value "names" facilitate that mapping.</p> <p>Further, like all other value designators, value names are always interpreted relative to a particular datatype, just as "1" can be used to refer to a value of the datatype defined</p>	<p>Clarify the scope of State and Enumerated values and value identifiers.</p> <p>The specification should make clear that:</p> <ul style="list-style-type: none"> - Enumerated types {short, medium, tall} and {light, medium, heavy} are distinct types of the family "enumerated", even though they have exactly the same number of elements, and the same characterizing operations: IsEqual and InOrder. - Enumerated types {short, medium, tall} and {short, moderate, medium, tall} are distinct types. <p>The specification should make clear whether "medium" in the above examples:</p> <ul style="list-style-type: none"> - denotes the same value - could denote the same value - could not denote the same value. 	

1 MB = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 Type of comment: ge = general te = technical ed = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

Template for comments and secretariat observations

Date: [2003-12-10](#)

Document: [US Comments on CD 11404](#)

1	2	(3)	4	5	(6)	(7)
MB ¹	Clause No./ Subclause No./ Annex (e.g. 3.1)	Paragraph/ Figure/Table/ Note (e.g. Table 1)	Type of com- ment ²	Comment (justification for change) by the MB	Proposed change by the MB	Secretariat observations on each comment submitted
				by: type newint = new integer but that value is not the same as the Integer "1" (per 9.1.2)		
USA-37	8.1.2	syntax	te	<p>There is no production for URI-text and no text that explains the usage of the "import" form. It is not clear at all what the "identifier" might refer to.</p> <p>URI should be a defined data type in clause 10, and should be referred to here.</p> <p>Also, the term "import" is misleading here, in that what must really occur is an interpretation of some reference codeset, in many cases defined by an ISO or national standard.</p> <p>If the URI may refer to an active resource, such as a metadata repository, in which the set of codes and their interpretations may change over time, that is not significantly different from the notion document-that-may-be-revised, with the caveat that the codeset is deemed to be fixed for some length of time approximately equal to the lifetime of (a version of) the GPD datatype. If, however, the intent is that the value-set of the data type is dynamic in nature, that notion is not a State datatype, in the sense of ISO 11404. It is rather a "new characterstring" datatype, with the interpretation of "current value space maintained by named authority".</p> <p>Finally, it is appropriate to allow the source reference to be an ASN.1 object-identifier, since all ISO standards have such an identifier.</p>	<p>Change the last two productions to:</p> <pre>value-domain-source = "source", list-source ; list-source = xxx-identifier URI-value objectidentifier-value ; URI-value = string-literal ;</pre> <p>And modify the Values paragraph to read:</p> <p>When the state-value-list form of state-values is used, <existing text>.</p> <p>When the value-domain-source form is used, the value set shall be exactly the set of code values specified in the document identified by the list-source value. When the list-source is a URI-value, it shall denote a valid value of the URI datatype, as defined in 10.x.y. When the list-source is an objectidentifier-value, it shall denote a valid value of the objectidentifier datatype, as defined in 10.1.10. In either case, the list-source value shall identify a document that explicitly defines a set of code values and their denotations. When the list-source is a xxx-identifier, it shall identify a xxx ???.</p> <p>(It may also be appropriate to add the "If the URI..." paragraph in the comment as a Note.)</p>	
USA-38	8.1.2	syntax	ed	The term "value-domain" is 11179, whereas the term "value-space" is used 11404. (Properly they should probably both be "value set", in that both "space" and "domain" have inappropriate connotations with respect to formal usage.)	Change value-domain-source to value-space-source in all occurrences (including the above).	
USA-39	8.1.3	syntax	te	The non-terminal "URI-to-value-domain" is not defined,	Either:	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

Template for comments and secretariat observations

Date: [2003-12-10](#) Document: [US Comments on CD 11404](#)

1	2	(3)	4	5	(6)	(7)
MB ¹	Clause No./ Subclause No./ Annex (e.g. 3.1)	Paragraph/ Figure/Table/ Note (e.g. Table 1)	Type of comment ²	Comment (justification for change) by the MB	Proposed change by the MB	Secretariat observations on each comment submitted
				nor is it supported by text. Moreover, the definition of "ordering" in an external value domain is extremely difficult to determine.	Strike the UR-to-value-domain alternative. or: Make it analogous in syntax and explanation to 8.1.2 (see USA-37), and include the requirement that the referenced document explicitly defines an ordering of the codes in the codeset.	
USA-40	8.1.3		ed		Add a Note at the end of 8.1.3 as follows: Note: The ordering on enumeration types imposed by programming languages is a convenience that allows programs to reference all the values via for-loops and enables the compiler to use integer encodings to simplify implementation. Properly, the Enumeration type should be chosen over the State type only when the ordering has semantic value. However, it may be necessary to declare the datatype of an object to be an Enumerated GPD when the purpose is to ensure the correct interpretation of an integer-based implementation.	
USA-41	8.1.10	1st para	ed	"which are known to certain applications to some finite precision and must be distinguishable to at least that precision in those applications" Why is it useful to talk about applications here? Does this imply that the same datatype may not be meaningful to some other application?	Prefer: "which are expressed to some finite precision and must be distinguishable to at least that precision"	
USA-42	8.1.10	all	te	The definition of Real and its parameters should be aligned with ISO 10967-1. ISO 10967-1 specifies 5 parameters for a floating-point data type. While all interesting floating-point types are bounded, however, that is not a requirement for Real itself, and can be captured with Range. The problem lies with the expression of values near zero: How does Real characterize the smallest value that can	Clarify the relationship to ISO 10967-1.	

1 MB = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 Type of comment: ge = general te = technical ed = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

Template for comments and secretariat observations

Date: [2003-12-10](#)

Document: [US Comments on CD 11404](#)

1	2	(3)	4	5	(6)	(7)
MB ¹	Clause No./ Subclause No./ Annex (e.g. 3.1)	Paragraph/ Figure/Table/ Note (e.g. Table 1)	Type of com- ment ²	Comment (justification for change) by the MB	Proposed change by the MB	Secretariat observations on each comment submitted
				be distinguished from zero, and how does that relate to the 10967-1 characterization?		
USA-43	8.1.11	all	te	This definition and its parameters should be aligned with ISO 10967-3. The definition given appears to appeal to a polar model of complex.	Align with ISO 10967-3	
USA-44	8.2.6	syntax	ed	The production for extended-type allows any number of occurrences of "sentinel" before the single extended value list. A single optional occurrence of sentinel should precede either the list or the individual values.	Correct the production. If necessary reword the semantics under "values", and add a reference to 6.2.	
USA-45	8.2.6		te	There is a contradiction with regard to value spaces. On the one hand, each datatype is defined in terms of a value space. On the other hand, we are told that the values in the datatype definition are "regular" values, and that sentinel values are also part of the value space.	Reconcile this contradiction by stating: The value space of a datatype is the set of values specified in the definition of the datatype. Sentinel values are values that can occur wherever values of the value space can occur; they can be distinguished by isEqual from values in the value space. Sentinel values must be specified explicitly even for a datatype that is defined axiomatically. NOTE – It follows that {short, medium, tall} and {short, medium, tall; sentinels = Unknown, Unspecified} are two distinct datatypes with the same value space.	
USA-46	8.2.7	all	te	8.2 refers to an override-type, but this clause does not define override-type; it defines override-qualifier. Moreover, override qualifies a "class member", not a data type. And in that role, it performs a function akin to the explicit subtype.	Delete the reference to override-type in the production for subtype in 8.2. Delete 8.2.7 and move its syntax and semantic content to 8.4.2.	
USA-47	8.3.1	Value	ed	"the conceptual form (tag-value, alternative-value), where tag-value ..." does not refer to syntactic objects and should not use the syntactic markup.	change the font, and perhaps, remove the hyphens, to conform to the text conventions stated in 5.2.	
USA-48	8.3.1	choice-type	ed		In the production for choice-type, insert a comma after "of".	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

Template for comments and secretariat observations

Date: [2003-12-10](#)

Document: [US Comments on CD 11404](#)

1	2	(3)	4	5	(6)	(7)
MB ¹	Clause No./ Subclause No./ Annex (e.g. 3.1)	Paragraph/ Figure/Table/ Note (e.g. Table 1)	Type of com- ment ²	Comment (justification for change) by the MB	Proposed change by the MB	Secretariat observations on each comment submitted
USA-49	8.3.1	choice-value	ed		In the production for choice-value, strike the comma preceding the semicolon.	
USA-50	8.3.2	Values	ed	"the single value null" should read "the single value 'null'" (with no font change). In the paragraph following the value syntax, "null" should be bold italic Courier without quotation marks – it refers to the syntactic object.		
USA-51	8.3.3	procedure-declaration	te	<p>" A <i>procedure-declaration</i> declares the <i>procedure-identifier</i> to refer to a (specific) value of the <i>procedure</i> datatype ..."</p> <p>This was acceptable for FORTRAN and C, but not for Java and C++. The draft does not properly distinguish between a signature and a procedure value.</p> <p>The "procedure value" conceptualizes the behavior of the body of the procedure; the procedure-type, as defined here, characterizes some aspects of a procedure, namely the allowable values of the actual parameters. Adding a (syntactic) name to the signature (i.e. to the procedure-type) also associates those parameters with explicit roles, and identifies a nominal functionality. But the signature only characterizes the value to that extent; it doesn't necessarily identify a single value, in the sense of a single body. In the particular case of an operation on a class, the name identifies a "member" of the class whose value is a procedure value of the procedure-type/functionality given in the signature, but *not always the same procedure value/body*.</p> <p>If it is the intent of 11404 that a name identifies a single functionality, but not necessarily a single body, then we have to define procedure-value that way. If a name identifies a single body, then we have to create a work-around in defining the syntax and semantics for operations on a class.</p> <p>It is also necessary to address, but possibly only in a</p>	<p>Clarify the model of procedure-type, signature, procedure-declaration, procedure-value that is intended in 11404, and modify 8.3.3 and 8.4.2 accordingly.</p> <p>It is possible that 99% of the text should stand, and all that needs to be said is that a "procedure value" means a conceptual functionality, and not a specific body.</p>	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

Template for comments and secretariat observations

Date: [2003-12-10](#)

Document: [US Comments on CD 11404](#)

1	2	(3)	4	5	(6)	(7)
MB ¹	Clause No./ Subclause No./ Annex (e.g. 3.1)	Paragraph/ Figure/Table/ Note (e.g. Table 1)	Type of com- ment ²	Comment (justification for change) by the MB	Proposed change by the MB	Secretariat observations on each comment submitted
				Note, the "polymorphism" notions of some OOPLs. Some of those notions are of the "conformable datatypes" kind (see USA-2). Others are about syntactic concerns in procedure signatures. But still others are about the relationship between signatures and procedure values.		
USA-52	8.3.3	Values	te	Note 5 and the last sentence of the paragraph before Value syntax touch on the issue of side-effects. It is clear that the given mathematical model of procedure as a mapping really doesn't address side-effects. ISO 13886 does discuss side-effects and gives a more elaborate model of procedure.	The text on Values should explicitly say that the model it presents assumes there is no persistent state and no side-effects, and refer to ISO 13886 for a proper model.	
USA-53	8.4.2		ed	The term "member" of a "class" has the wrong intuitive interpretations by speakers of English. It arises from the fact that C described the "fields" of a structured-type (a "record") as "members", and C++ extended the structured-type notion to get "class". But the intuitive interpretation of the "members" of a class is the values in its value space. The more common word in the modeling world is "attributes", which also more correctly describes the relationship of the members/fields to the instances/values of the class datatype.	Replace all occurrences of the word "member" in this context with "attribute".	
USA-54	8.4.4	Components	te	"The <i>element-type</i> shall designate an exact datatype..." This requirement arises from the model of Bag as a mapping from the element-type into Integer. And it disallows Bag of Real and the like, which prevents Bag from being a "generic" or "unstructured" aggregator. If we retain this definition of Bag, and this constraint, then we need to add an aggregate generator that does not constrain the element-type.	Discard the model of bag as a mapping to Integer. Allow the base type to be any datatype. Keep the characterizing operations except for Delete, but define them in terms of Serialize and then Sequence.	
USA-55	8.6	Syntax	te	There is no defining production for actual-param-list.		
USA-56	8.6.1	Examples	te	In each of the examples, the first normative declaration does not conform to the syntax given in 9.4, or anything like it. This appears to be a new kind of declaration that	Define the declaration syntax used in the examples in 9.x (see also USA -66)	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

Template for comments and secretariat observations

Date: [2003-12-10](#)

Document: [US Comments on CD 11404](#)

1	2	(3)	4	5	(6)	(7)
MB ¹	Clause No./ Subclause No./ Annex (e.g. 3.1)	Paragraph/ Figure/Table/ Note (e.g. Table 1)	Type of comment ²	Comment (justification for change) by the MB	Proposed change by the MB	Secretariat observations on each comment submitted
				like it. This appears to be a new kind of declaration that is not documented in clause 9 and should begin with a different keyword.		
USA-57	8.6.1	Example 1	te	What is the value space of a record datatype, all of whose fields are optional? Is it the same as the value space of the record datatype, designated: <pre>record(field1 : optional(characterstring(Latin-1)), field2 : optional(real(10,7))) ?</pre>	Explain the relationship of optional fields as specified via 8.6 to the Optional generator in 10.2.4, at least in a Note. The interpretation of a 'void' field in 10.2.4 is stated to be that that field "has no value". It is not clear that that is the intended interpretation of the provision in Example 1.	
USA-58	8.6.1.1	values	te	obligation=unspecified means the provision can specify nothing of "normative" value.	Delete "unspecified".	
USA-59	8.6.1.1	values	te	obligation=default means the "provision" specifies a default value for something. This seems to say that the <something> is syntactically optional, but that some value is always present. Syntactic options should be specified in syntax. "default" may be a useful part of procedure signatures, and might properly have syntax (and explanation) in 8.3.3.	Delete "default" in 8.6.1.1. Make it a modifier for procedure parameters in 8.3.3.	
USA-60	8.6.1.2	values	te	Target=runtime type and target=runtime access deal with concepts that relate to mappings. They seem to add requirements to the support requirements for the type.	Explain the relationship of these values to mappings and the internal-type concept in Clause 11. If necessary, add supporting text to clause 11.	
USA-61	8.6.1.2	values	te	Target=value refers to: all values? some value? a particular value? Is this further specified by scope?	Clarify	
USA-62	8.6.1.3		te	How is "what is affected by the provision" (scope) different from "what is intended to satisfy the provision" (target)? It seems that scope merely refines "target". Or perhaps the definitions are exactly reversed.	Clarify	
USA-63	8.6.1.4	values	te	What does it mean for an identifier to be "undefined"? I would think that an identifier is "undefined" if it appears where a defined/declared identifier is syntactically	Clarify	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

Template for comments and secretariat observations

Date: [2003-12-10](#)

Document: [US Comments on CD 11404](#)

1	2	(3)	4	5	(6)	(7)
MB ¹	Clause No./ Subclause No./ Annex (e.g. 3.1)	Paragraph/ Figure/Table/ Note (e.g. Table 1)	Type of com- ment ²	Comment (justification for change) by the MB	Proposed change by the MB	Secretariat observations on each comment submitted
				required. That should mean that the program-text is syntactically incomplete and therefore non-conforming.		
USA-64	8.6.1.4	syntax	te	selecting-expr, range-expr and value-expr are not defined	Complete the production set, or correct the spelling of the non-terminals	
USA-65	8.6.3	all	te	8.4.1 requires the field identifiers to be unique within the record definition. What is this specifying?	clarify or delete	
USA-66	9.4	all	te	The text of paragraph 1 says that this feature does a subset of what 9.1.2 does. What does it do that is different? The examples of "normative" in 8.6.1, however, don't appear to define datatypes. Rather they define "provisions". Since using the normative keyword instead of "type" in a type-declaration doesn't add any semantics (provisions can appear in a type-declaration, too), the normative declaration should be as it appears in 8.6.1.	Change the normative text to say that normative simply provides an identifier for a provision. Change the production to read: <pre>normative-datatype-declaration = "normative", identifier, "=", provision-statement ;</pre>	
USA-67	9.5	first para	te	This paragraph is taken verbatim from 8.5 and is out-of-place here. This section is really about the program-text concept and the construction of a program-text from direct and imported declarations.	Replace with: This section describes declarations that relate to construction of a program-text from other program-texts.	
USA-68	9.5.1	Components	te	A "Components" section is only appropriate in the definition of a datatype. The term "source" isn't defined and doesn't clearly refer to any syntactic object. How can a datatype (declaration) be obtained from a source that is a "type-identifier", i.e. a type-reference?	In the syntax, replace URI-or-type-identifier with source, and leave the production. Replace the Components tag and first sentence of the paragraph with: "The <i>source</i> value identifies a resource that contains a program-text, as specified in x.x. Each declaration in that program-text is included in the current program-text as if it appeared verbatim in the current program-text. Exceptions: "	
USA-69	9.5.2	Syntax	te	The intent of this feature is to support a blind "macro-	Add a "macro-definition" declaration, e.g.	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

Template for comments and secretariat observations

Date: [2003-12-10](#)

Document: [US Comments on CD 11404](#)

1	2	(3)	4	5	(6)	(7)
MB ¹	Clause No./ Subclause No./ Annex (e.g. 3.1)	Paragraph/ Figure/Table/ Note (e.g. Table 1)	Type of comment ²	Comment (justification for change) by the MB	Proposed change by the MB	Secretariat observations on each comment submitted
				<p>substitution" feature in a program-text. But:</p> <p>It should not be restricted to, or even related to, formal parameters of the data type declaration; and It should have a syntax that alerts the lexical analyzer, no matter where it occurs.</p> <p>Otherwise it must be called out in the syntactic productions for the elements of a datatype declaration. E.g. in the example given, it would have to be a valid expansion of the field-specifier non-terminal.</p>	<p>macro-definition = "macro", identifier, "(", [macro-param-list], ")", "{", text, ")"</p> <p>where the macro-param-list is a list of identifiers, and text is any character-string, possibly containing "parameter-references" of a form like the XML: "&name;"</p> <p>then the program-text can anywhere thereafter contain an occurrence of &macro-id(string-literal parameter values) and the effect of "eval" will take place at that point, substituting the "text" of the macro definition with the values of the parameters substituted verbatim for the references to them (no extra spaces or delimiters).</p>	
USA-70	9.5 and 7.6		ed	The program-text and lexical features should be together, probably in clause 7.	Move 9.5 to clause 7.	
USA-71	10.2.3	all	te	Cyclicenumerated has never been identified with any use.	Strike 10.2.3	
USA-72	10.1		te	Add a new defined datatype "URI", for use in 9.5.1 and 8.1.2 and for general use.	Add 10.1.x Uniform Resource Identifier (Definition to be provided, with normative reference to IETF RFC)	
USA-73	10.1		Minor	Add a new defined datatype family for float/double that explicitly adds the NaNs to Real(2,p) and constrains the range.	Add 10.x.x IECfloat Type IECfloat(p,c) = Real(2,p):range(-2 ^c ..2 ^c) plus(sentinel pinf, ninf, indeterminate, underflow) With appropriate definition and reference to IEC 60559.	
USA-74	10		te	<p>Add a new defined datatype "interface" which is derived from "class" by restricting all the "members" (attributes) to have procedure datatypes (i.e. to be methods).</p> <p>This corresponds to a feature in IDL, WSDL and UML that characterizes only the interactions exhibited by instances</p>	Add 10.x.x Interface (Definition to be provided.)	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

Template for comments and secretariat observations

Date: [2003-12-10](#)

Document: [US Comments on CD 11404](#)

1	2	(3)	4	5	(6)	(7)
MB ¹	Clause No./ Subclause No./ Annex (e.g. 3.1)	Paragraph/ Figure/Table/ Note (e.g. Table 1)	Type of com- ment ²	Comment (justification for change) by the MB	Proposed change by the MB	Secretariat observations on each comment submitted
				of the type – i.e. true encapsulation.		
USA-75	8.3		te	<p>The modeling language EXPRESS (and some others) defines a data type that is the "undiscriminated union" of the value sets of other data types (of similar kinds). That is, it should be possible to define a State type or Characterstring type or Class type whose value space is the union of the value spaces of other State types or Characterstring types or Class types. In each case, the semantic interpretation of the union is pure: all of the named types are implicitly or explicitly subtypes of some single super-type, so that all of the values are consistently interpreted with respect to the supertype value space.</p> <p>In the particular case of class values, it is assumed that there is a universal "object-identifier" data type to which all of the class values belong. In the case of state-types, the union itself indicates that all the value-identifiers belong to a single "namespace".</p> <p>This is different from the Choice generator, where the alternative values are heterogeneous. It is also different from the C "union", in which the values can be heterogeneous, but the discriminator is simply not declared.</p>	<p>Add a subclause to 8.3 (or 8.2?):</p> <p>8.3.x Union data types</p> <p>Description: Union generates a datatype called a <i>union</i> datatype, each of whose values is a single value from any of a set of alternative datatypes. The alternative datatypes of a union datatype are all subtypes of another datatype, called the base datatype.</p> <p>Syntax:</p> <pre>union-type = "union", "(" , subtype-list, ")", ["source", basetype] ; subtype-list = subtype, { ",", subtype } ; subtype = type-specifier ; basetype = type-specifier ;</pre> <p>Components: All of the datatypes denoted by subtypes in the subtype-list shall be of the same kind: if the base-type is specified, they shall all be subtypes of the base-type; if the basetype is not specified they shall all be state-types or defined-types whose definition is a state-type, or they shall all be class-types or defined-types whose definition is a class-type.</p> <p>Values: The value space of the union datatype is the union of the value spaces of the subtypes in the subtype list. Values are distinguished as they are distinguished in the basetype. If the subtypes are state-types, the values are distinguished only by their identifiers, and not by the subtype in which they were declared. If the subtypes are class-types, the values are distinguished in the undefined state datatype that underlies all class values.</p>	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.

Template for comments and secretariat observations

Date: [2003-12-10](#)

Document: [US Comments on CD 11404](#)

1	2	(3)	4	5	(6)	(7)
MB ¹	Clause No./ Subclause No./ Annex (e.g. 3.1)	Paragraph/ Figure/Table/ Note (e.g. Table 1)	Type of comment ²	Comment (justification for change) by the MB	Proposed change by the MB	Secretariat observations on each comment submitted
					Operations: The characterizing operations on the union datatype are those of the base-type.	
USA-76	8.3.1, 8.4, 8.6		te	<p>"Extensibility" is a needed property of record types or choice types, to support features of EXPRESS and XML schema, and to support mappings to several languages, e.g. to support discriminators and Optional fields.</p> <p>An arcane example in 8.6 suggests that it is supported, but it is nowhere documented normatively, and clause 8.6 provides no explicit syntax for declaring a datatype extensible or for declaring an extension of it.</p>	<p>a. Add a subclause to clause 6:</p> <p>"6.x Extensibility</p> <p>A generated datatype is said to be <i>extensible</i> if mappings of the datatype can additional elements, fields, or alternatives without altering the required behaviours of values of the datatype."</p> <p>b. Add syntax to 8.3.1 and to 8.4.x to provide syntax by which a datatype can be declared extensible.</p> <p>c. If appropriate, add syntax to clause 8.2 or 9 to declare a generated datatype to be an extension of another generated datatype.</p> <p>d. Define the requirements/permissions for supporting extensibility in Clause 11.</p>	
USA-77			te	Need to reference annotations for ASN.1 and provide and ASN.1 example in the annexes.	Provide an Annex containing an informative mapping between General-purpose Datatypes and ASN.1 datatypes. If necessary, define annotations specific to this mapping.	

1 **MB** = Member body (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

NOTE Columns 1, 2, 4, 5 are compulsory.