INTERNATIONAL STANDARD

ISO/IEC 15476-3

First edition
2006-01-01

Information technology — CDIF semantic metamodel —

Part 3: Data definitions

Technologies de l'information — Métamodèle sémantique CDIF — Partie 3: Definition de données

Reference number
ISO/IEC 15476-3:2006(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Table of Illustrations

Table of Tables

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 15476-3 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Software and system engineering*.

ISO/IEC 15476 consists of the following parts, under the general title *Information technology — CDIF semantic metamodel*:

— *Part 1: Foundation*

— *Part 2: Common*

— *Part 3: Data definitions*

— *Part 4: Data models*

— *Part 5: Data flow models*

— *Part 6: State/event models*

Introduction

This International Standard will assist the vendors and users of modelling tools and meta-data repositories in developing mechanisms for interchanging information. This International Standard specifies an element of a family of related standards. When used together, these International Standards specify a mechanism for transferring information between tools.

ISO/IEC 15474-1:2002**,** *Information technology - CDIF framework - Part 1: Overview* and ISO/IEC 15474-2**,** *Information technology - CDIF framework - Part 2: Modelling and extensibility* should be read first when initially exploring CDIF. The first explains the overall CDIF architecture and how the family of standards fits together. The second explains the scope, and modelling approach in CDIF. The CDIF meta-metamodel and extensibility mechanism are also defined in that document.

This International Standard explains the data definitions subject area of the CDIF semantic metamodel, which defines the primitive data types and the objects which are used for structured data. The CDIF semantic metamodel is used to ensure that the information transferred by tools communicating using CDIF is expressed with an agreed meaning.

This International Standard has been developed with the wide support and participation of vendors, users, academia and government involved in or familiar with the CASE industry, its products and the general requirements associated with interchanging information between these products.

This document is organized into the following Clauses:

— Clause 1 to 5 are prescribed ISO/IEC Clauses.

— Clause 6: Subject area overview:

  This Clause gives an overview of the coverage of this subject area.

— Clause 7: Subject area summary:

  This Clause gives an overview of the content of this subject area.

— Clause 8: Subject area specification:

  This Clause gives the formal specification of all the objects defined in the subject area, and the formal reference to those used, but not defined in the subject area.

This document is intended to be used by anyone wishing to understand and/or use CDIF. This document provides a definition of a single subject area of the CDIF semantic metamodel. It is suitable for:

— Those evaluating CDIF;

— Those who wish to understand the principles and concepts of a CDIF transfer; and

— Those developing importers and exporters.

This document, ISO/IEC 15474-1:2002*, Information technology - CDIF framework - Part 1: Overview,* and the framework document ISO/IEC 15474-2:2002*, Information technology - CDIF framework - Part 2: Modelling and extensibility,* should be read first when initially exploring CDIF and before attempting to read other documents in the CDIF family of International Standards.

While there are no specific prerequisites for reading this document, it will be helpful for the reader to have familiarity with the following:

— Entity-Relationship-Attribute modelling;

— Modelling (CASE) tools;

— Information repositories;

— Data dictionaries; and

— Multiple meta-layer modelling.

Information technology — CDIF semantic metamodel —

Part 3:
Data definitions

1 Scope

The CDIF family of International Standards is primarily designed to be used as a description of a mechanism for transferring information between modelling tools. It facilitates a successful transfer when the authors of the importing and exporting tools have nothing in common except an agreement to conform to CDIF. The language that is defined for the transfer format also has applicability as a general language for import/export from repositories. The CDIF semantic metamodel defined for CASE also has applicability as the basis of standard definitions for use in repositories.

The International Standards which form the complete family of CDIF standards are documented in ISO/IEC 15474-1:2002, *Information technology — CDIF framework — Part 1: Overview*. These standards cover the overall framework, the transfer format and the CDIF semantic metamodel.

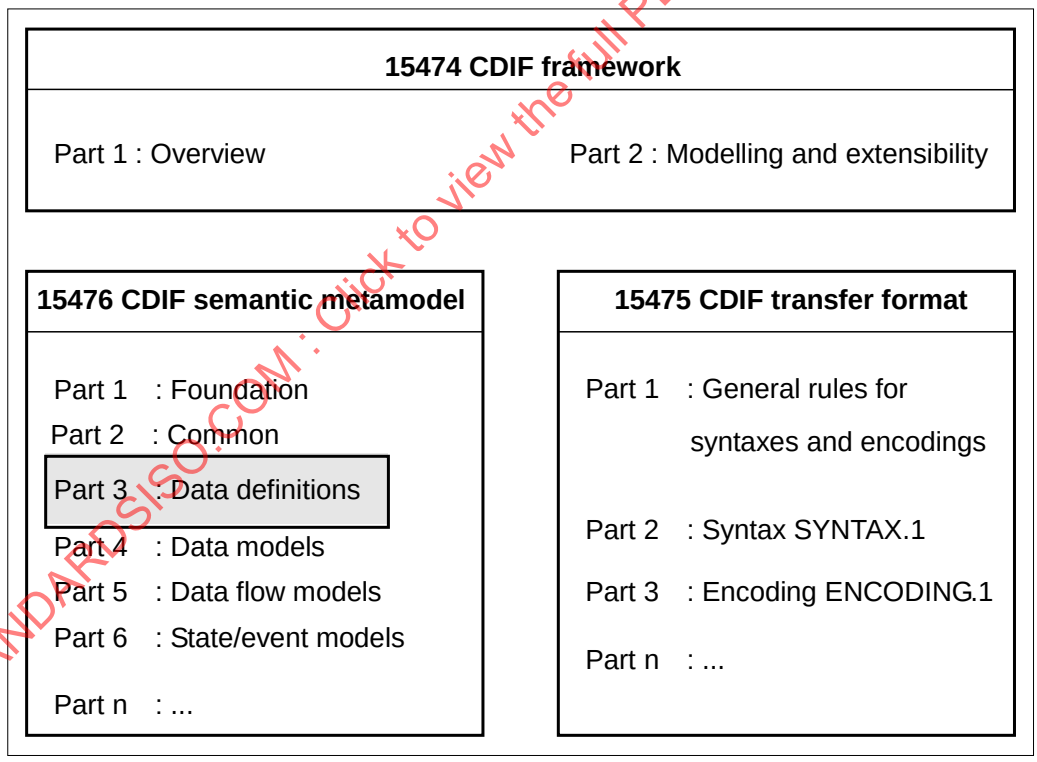

Figure 1 – CDIF family of International Standards

The diagram in Figure 1 depicts the various International Standards that comprise the CDIF family of standards. The shaded box depicts this International Standard and its position in the CDIF family of standards.

This International Standard defines the Data Definition Subject Area of the CDIF semantic metamodel. This subject area contains meta-objects that are used as the basis of the data components of other subject area standards, and also meta-relationships and meta-attributes that are applicable to all data-related meta-objects.

2 Conformance

2.1 General

A product is fully standards conformant to a CDIF subject area standard if and only if it is input-conformant, output-conformant and round-trip conformant to each and every *MetaEntity*, *MetaRelationship*, *MetaAttribute*, and *AttributableMetaObject* which is defined and/or used in that standard, and it is also CDIF architecture conformant. A product may be partially input-conformant, and/or partially output-conformant, and/or partially round-trip conformant to a CDIF subject area standard.

2.2 Input conformance

Input conformance for a specific *MetaEntity*, *MetaRelationship*, *MetaAttribute*, or *AttributableMetaObject* (short: *CollectableMetaObject*) is determined by applying the following test:

A set of meta-data containing all meanings and structures standardized by a CDIF subject area is imported by the product under test. Then the meta-data which has arrived in the product is examined. The following options exist for the relation between the input (CDIF) meta-data and the imported (product) meta-data:

For a specific *CollectableMetaObject*:

1 The product is input conformant if each instance of the specific *CollectableMetaObject* has arrived in the product without change of meaning or structure. If the *CollectableMetaObject* is a meta-entity or meta-relationship, its structural relationships to other *CollectableMetaObjects* have been preserved. If the *CollectableMetaObject* is a meta-attribute, the value of the meta-attribute has been preserved.

2 The product is input morphing conformant if each instance of the specific *CollectableMetaObject* has arrived in the product, but with some changes in meaning or structure. If the *CollectableMetaObject* is a meta-attribute, the value(s) for some instances of the meta-attribute have changed.

3 The product is not input conformant for that *CollectableMetaObject* if neither of the previous tests is satisfied.

2.3 Output conformance

Output conformance for a specific *CollectableMetaObject* is determined by applying the following test:

For the product being tested, a set of meta-data that includes all possible meanings and structures representable in that product is exported. Then the meta-data that has been exported is examined. The following options exist for the relation between the product's meta-data and the exported (CDIF) meta-data:

For a specific *CollectableMetaObject*:

1 The product is output conformant if all of the meaning and structure for the specific *CollectableMetaObject* has been represented as meta-data in the product and has been exported as one or more instances of that *CollectableMetaObject*. If the *CollectableMetaObject* is a meta-attribute, the correct value of the meta-attribute has been exported.

2 The product is output morphing conformant if each instance of meta-data in the product that has the same meaning and structure as the *CollectableMetaObject* has been exported, but some instances have been exported as a different *CollectableMetaObject* or some of the meaning and structure has been changed.

3 If the product does not represent the meaning and structure associated with the *CollectableMetaObject*, output conformance for that *CollectableMetaObject* is not applicable to the product.

4 In all other cases, the product is not output conformant for that *CollectableMetaObject*.

2.4 Round-trip conformance

Round-trip conformance for a specific *CollectableMetaObject* is determined by applying the following test:

A set of meta-data containing all meanings and structures standardized by a CDIF subject area is imported by the product under test. Then the meta-data is exported again. The following options exist for the relation between the input meta-data and the output meta-data:

For a specific *CollectableMetaObject*:

1 The product is round-trip conformant if the meaning and structure of each instance of the *CollectableMetaObject* is preserved without changes during the round-trip. For a vendor to claim round-trip conformance, it is also necessary for the tool to be able to perform create, read, update, and delete operations on the imported (product) meta-data corresponding to the instances of the *CollectableMetaObject*.

2 The product is round-trip morphing conformant if each instance of the input *CollectableMetaObject* is preserved, but with some changes in meaning and/or structure. If the *CollectableMetaObject* is a meta-entity or meta-relationship, some of its instances' structural relationships to other *CollectableMetaObjects* have changed, or some instances have been transformed into other *CollectableMetaObjects*, or instances of other *CollectableMetaObjects* have been transformed into instances of the *CollectableMetaObject.* If the *CollectableMetaObject* is a meta-attribute, the values of some instances of the meta-attribute have changed or the domain of the meta-attribute has changed.

3 In all other cases, the product is not round-trip conformant for that CollectableMetaObject.

3 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 31-1:1992 *Quantities and units – Part 1: Space and time*

ISO 4217:2001, *Codes for the representation of currencies and funds*

ISO/IEC 9945-1:1996 *Information technology – Portable Operating System Interface (POSIX) –Part 1: System Application Program Interface (API) [C Language]*

ISO/IEC 13238-1, *Information technology - Data management export/import - Part 1: Standardization framework.*

ISO/IEC 15474-1, *Information technology — CDIF framework — Part 1: Overview*

ISO/IEC 15474-2, *Information technology — CDIF framework — Part 2: Modelling and extensibility*

ISO/IEC 15476-1, *Information technology — CDIF semantic metamodel — Part 1: Foundation*

ISO/IEC 15476-2, *Information technology — CDIF semantic metamodel — Part 2: Common*

ISO/IEC 15476-4 *Information technology — CDIF semantic metamodel — Part 4: Data models*

CHARACTER SETS, IANA, available at <http://www.iana.org/assignments/character-sets>

4 Terms and definitions

For the purposes of this document, the following definitions apply. Unless otherwise noted, the definitions are specific to this International Standard.

4.1 From other International Standards

4.1.1 ISO/IEC 15474-1

This part of ISO/IEC 15476 makes use of the following terms defined in ISO/IEC 15474-1:

CDIF
CDIF family of standards
CDIF semantic metamodel
CDIF meta-metamodel
CDIF transfer
Instance
Meta-attribute
Meta-entity
Metamodel
Meta-object
Meta-relationship
Model
Subject area
Transfer
Transfer format

4.1.2 ISO/IEC 13238-1

This part of ISO/IEC 15476 makes use of the following terms from ISO/IEC 13238-1:

Exporter
Importer

4.1.3 For this International Standard

For the purpose of this part of ISO/IEC 15476 new terms are defined when introduced. Double quotes are used to introduce new terms (e.g., "model layer")

5 Symbols (and abbreviated terms)

5.1 Naming, diagramming and definition conventions

Conventions for naming, diagramming, describing and defining meta-objects can be found in Clause 7 of the framework document (ISO/IEC 15474-2:2002*, Information technology - CDIF framework - Part 2: Modelling and extensibility*).

5.2 Abbreviations

The following abbreviation is used in this International Standard:

CDIF CASE Data Interchange Format (originally)

6 Data definition subject area overview

6.1 Introduction

The Data Definition Subject Area provides support for describing data objects and provides a data typing scheme. It allows for simple and complex structures, array and pointer qualification and domain specification. The Data Definition Subject Area diagrams are shown in Figure 9 through Figure 14.

6.2 Data Typing

A data type is described by one or more Attributes, and each of these Attributes may have a defined data type, represented by the meta-entity DataType. This Subject Area does not cover the internal representation of information within the basic data types; it only covers the concept of the type of information represented.

Figure 3 shows a fragment of the meta-model illustrating how data typing is represented. Figure 4 is an instance diagram of the model elements in Figure 3. Note that in the instance diagrams, values of meta-attributes (usually Name) are only given where they are required.

The data type of each Attribute is defined by relating it to the meta-entity DataType and its subtypes. These are described in subclauses 6.8 to 6.11. In the instance diagrams, the most appropriate subtype of DataType is used.

The meta-attribute Name of the meta-entity Attribute allows a local name to be defined for the attribute. If no value is given, then it is assumed that the name is the same as that of the underlying DataType. A default value for the Attribute can be defined using the meta-attribute DefaultValue, as can the optionality, using the meta-attribute IsOptional (The meta-attributes, DefaultValue and IsOptional, are defined in ISO/IEC 15476-4 *Data models*).

6.3 The General Structuring Mechanism

6.3.1 Introduction

Support for the decomposition and structuring of objects into other objects, and the reuse of definitions, is provided by a general structuring mechanism. This mechanism is defined in ISO/IEC 15476-2:2002*, Information Technology - CDIF Semantic Metamodel - Part 2: Common*, and is also used in other CDIF subject areas.

In this subject area DataTypes may be structured. Structuring of DataTypes is performed using the general structuring mechanism because its definition may be used for several objects in different contexts.

6.3.2 Meta-entities and Meta-relationships

The general structuring mechanism used in this subject area makes use of the following meta-entities:

- ComponentObject
- DefinitionObject

and the following meta-relationships:

- DefinitionObject.Contains.ComponentObject
- ComponentObject.References.DefinitionObject.

Figure 2 shows the part of meta-model for the general structuring mechanism related to meta-entity DataType.

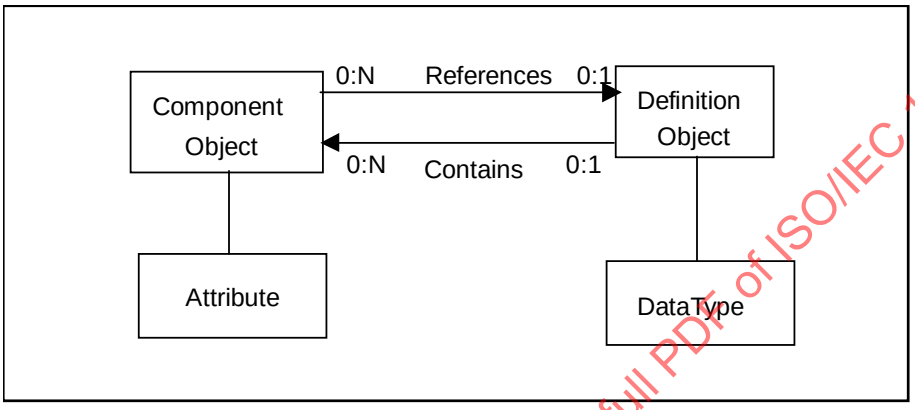

Figure 2 – Part of general Structuring Mechanism for data definitions subject area

Figure 3 and Figure 4 show how the general structuring mechanism represent a hierarchical data structure. The combination of the meta-relationship DefinitionObject.Contains.ComponentObject which is used for data structure and the meta-relationship ComponentObject.References.DefinitionObject which is used for data declaration is shown in Figure 3. The sample instance diagram is Figure 4. In this case, two data named **Customer** and **Order** are defined. Both data definition have fixed length sting type named **CustomerNumber.**

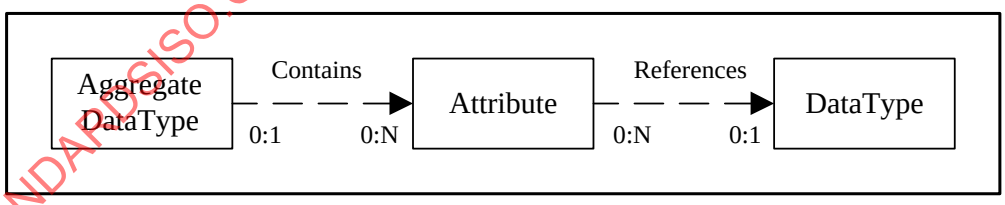

Figure 3 – Meta-model fragment for simple attribution with defined DataTypes

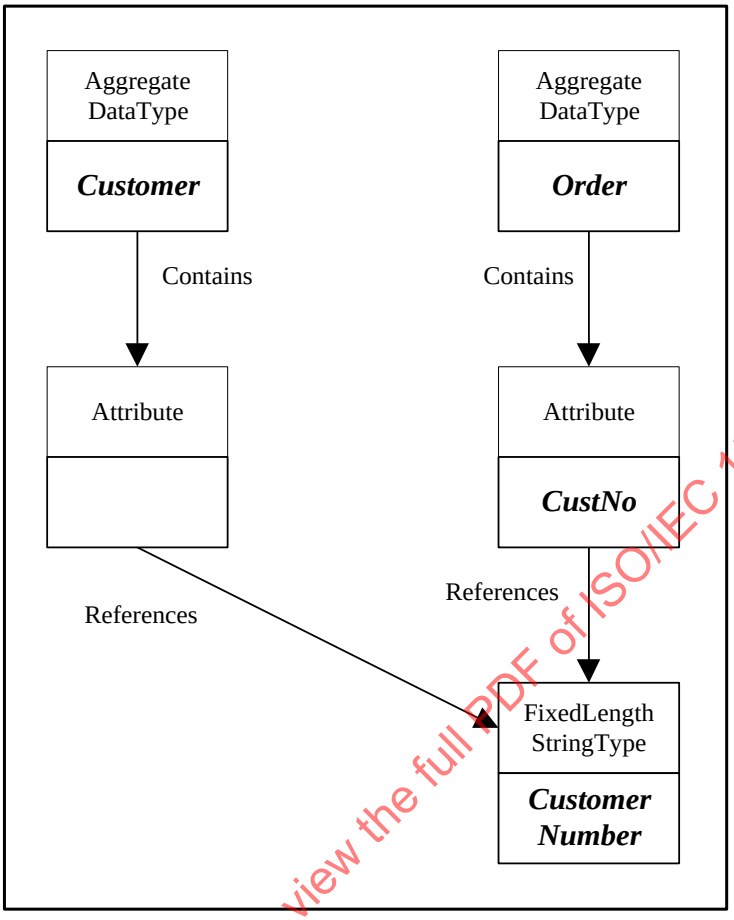

Figure 4 – Instance Diagram showing simple attribution with defined DataTypes

6.3.3 DefinitionObject and ComponentObject

DefinitionObject serves as an abstract supertype for all decomposition and structure definitions that may be reused. A *DefinitionObject* represents the definition of a *ComponentObject*, using meta-relationship *ComponentObject.References.DefinitionObject*. A *DefinitionObject* may contain *ComponentObjects*, using the *DefinitionObject.Contains.ComponentObject* meta-relationship. The *ComponentObjects* contained in a *DefinitionObject* represent the components of the definition.

Both *ComponentObject* and *DefinitionObject* represent abstract concepts and thus shall not be instantiated. Instead, appropriate subtypes, provided by this or other subject areas, are used for instantiating data definitions.

A subtype of *DefinitionObject* may be referenced by any number of *ComponentObjects* indicating that all those *ComponentObjects* share the same definition. For example, in Figure 5, *Attributes* ***a*** and ***b*** share the same definition, *AggregateDataType* ***MyStruct***.

Instances of the subtypes of *ComponentObject* are used to describe the structure of a deta described by a *DefinitionObject*. There may be any number instances of *ComponentObjects* per instance of *DefinitionObject*. In Figure 5, *Attributes* ***x*** and ***y*** comprise the structure of ***MyStruct***.

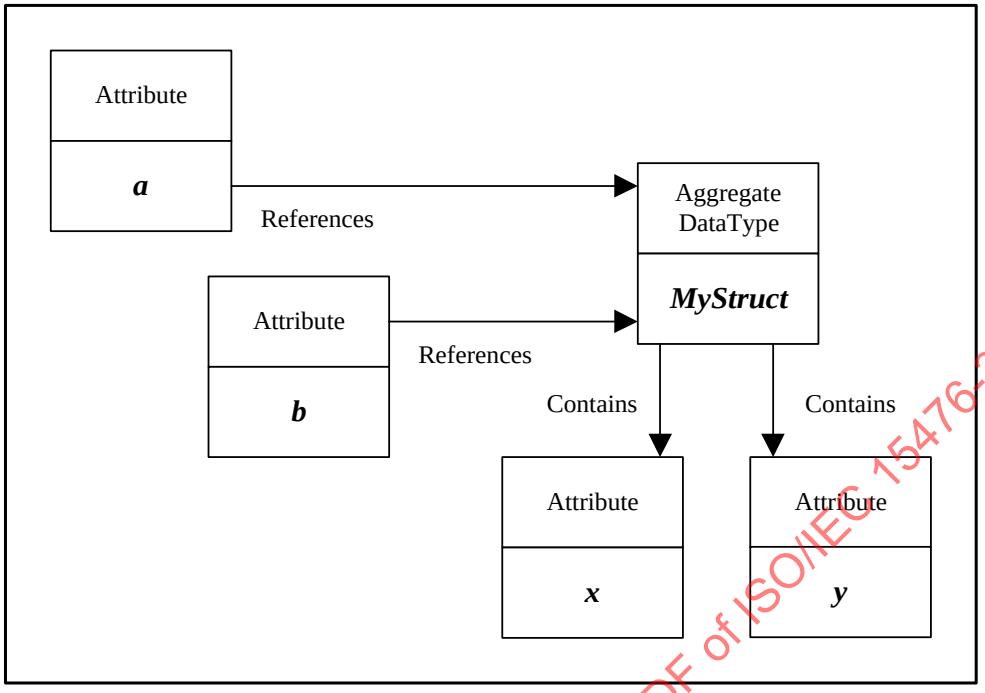

Figure 5 – Instance diagram of attributes sharing the same structured definition

For example, if a structure is built containing two components that share the same definition, the instance of structured *DefinitionObject* will contain two distinct instances of *ComponentObject* which refer to the same shared instance of the *DefinitionObject*.

The general structuring mechanism ensures that, for example, the instances of the components of one structured *Attribute* are different from those of another structured *Attribute* even if both *Attributes* share the same instance of a *DataType*.

6.4 Alternate Decompositions

This subject area supports alternate decompositions of *DataTypes* using the general structuring mechanism.

Decompositions may have one of two characteristics, represented by the meta-attribute *Operator* of the meta-entity *DefinitionObject*. *DataType* is a subtype of *DefinitionObject* and thus inherits the meta-attribute *Operator*.

If the instance value of this meta-attribute is ***AND***, this indicates that all instances of the all *ComponentObjects* related to the instance of *DefinitionObject* using the instance of meta-relationship *DefinitionObject.Contains.ComponentObject* are contained at the same time (i.e. inclusive).

For example, in Figure 6, the *AggregateDataType* instance ***Y1Def*** representing a data structure definition would set the value of *Operator* to ***AND*** because it contains all the attributes concurrently.

The value ***XOR*** of the *Operator* meta-attribute specifies that the *ComponentObject* instances contained in the *DefinitionObject* instance are exclusive alternates. This means that the object is decomposed into exactly one of the *ComponentObject* instances but which one is not specified. An alternate decomposition leaves it open as to which of the decompositions is to be used.

For example, in Figure 6, if an *Attribute* ***X*** has two decompositions into *Attributes* ***A***, ***B*** and ***C***, or into *Attributes* ***D*** and ***E***, a *AggregateDataType* ***XDef*** is created which is the definition of *Attribute* ***X***, using the instance of meta-relationship *ComponentObject.References.DefinitionObject*. *AggregateDataType* instance ***XDef*** has its meta-

attribute *Operator* set to ***XOR***. There are two *Attribute* instances ***Y1*** and ***Y2*** which are contained in ***XDef*** using meta-relationship *DefinitionObject.Contains.ComponentObject* instance, indicating that the *AggregateDataType* has two possible structures. The *Attribute* instances ***Y1*** and ***Y2*** have *AggregateDataType* instances ***Y1Def*** and ***Y2Def***, using *ComponentObject.References.DefinitionObject* instance. Those *AggregateDataTypes* instances have set their meta-attribute *Operator* to ***AND***, indicating that all their components are contained and they do not represent alternates. Note that if the *Operator* meta-attribute has a value of ***XOR***, then the contained *Attributes* instances do not actually exist as a part of the containing *Attribute's* structure, but are simply playing the role of a grouping mechanism. Those *AggregateDataType* instances ***Y1Def*** and ***Y2Def*** contain *Attribute* instances ***A***, ***B*** and ***C***, or ***D*** and ***E***, respectively.

For example, the "C" programming language supports the union concept. In this case the *AggregateDataType's* meta-attribute *Operator* would have the value ***XOR***.

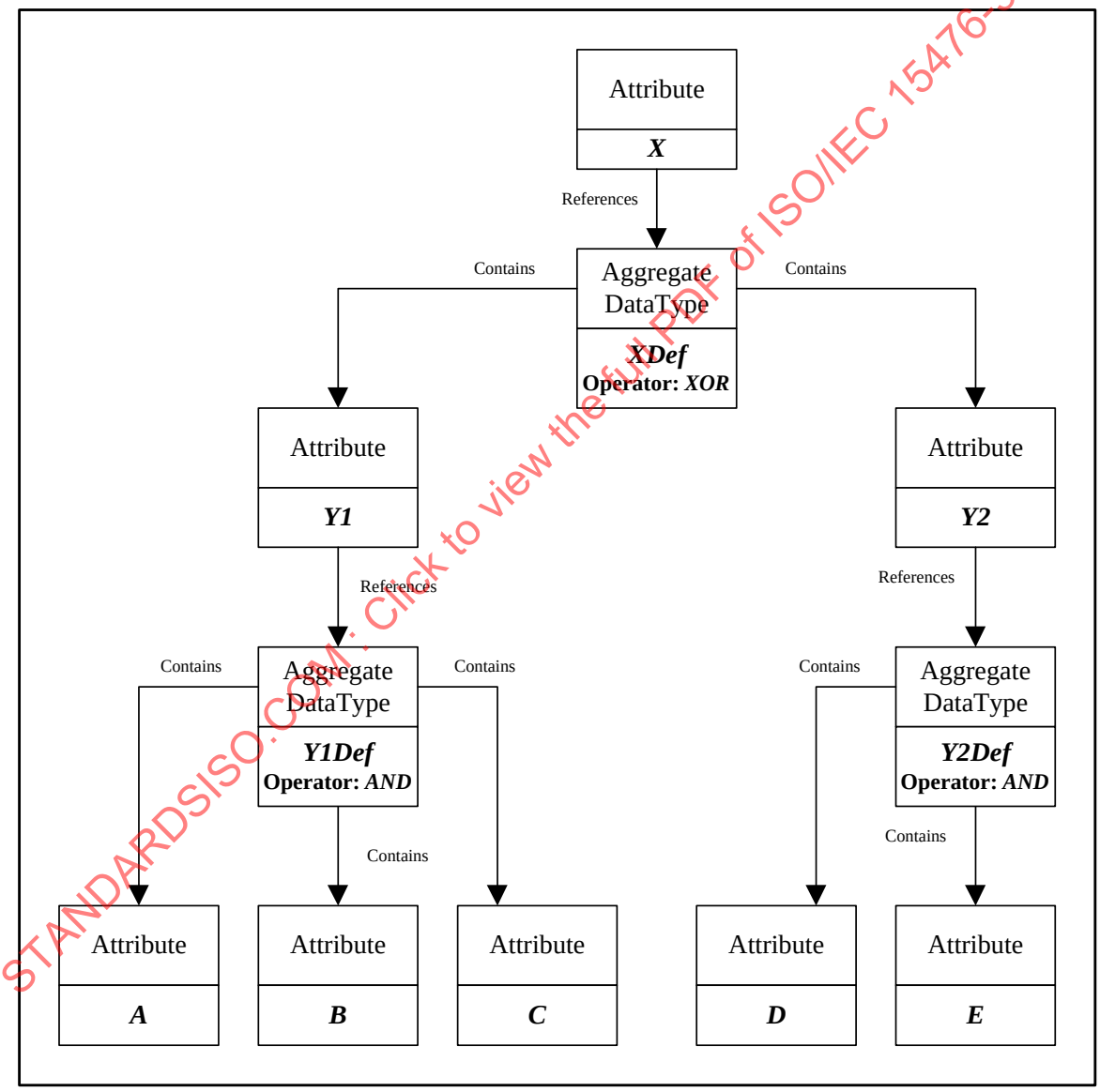

Figure 6 – Instance diagram of alternate Data Structures

The concepts of attributes, shared definitions and underlying types are illustrated in Figure 7.

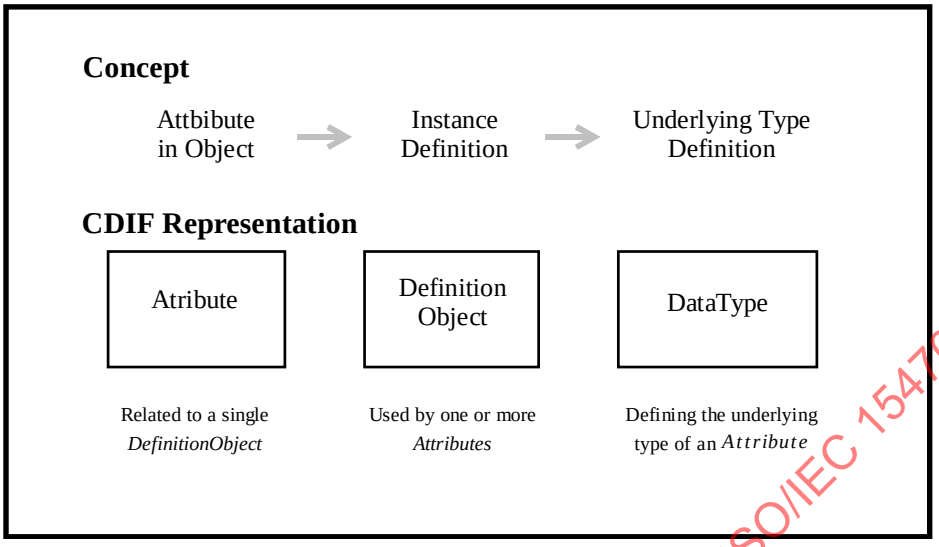

Figure 7 – Concepts of shared attribute values and definitions

6.5 Pointers and Arrays

A *DataType* can be qualified to indicate that, instead of a single occurrence of the data type itself, there is either an array of the data type or a pointer to the data type, or indeed any combination of these qualifiers. This is expressed using the meta-entities *QualifiedDataType* and *Qualifier* and its subtypes and the meta-relationship *QualifiedDataType.IsQualifiedBy.Qualifier* and *QualifiedDataType.IsQualificationOf.DataType*.

When a pointer is required, the *PointerQualifier* subtype is used. Likewise, when an array is required, the *ArrayQualifier* subtype is used. Arrays may be further specified as unbounded (via *UnboundedArrayQualifier*) or bounded (via *BoundedArrayQualifier*). The type of the *ArrayQualifier* may be expressed using the meta-relationship *ArrayQualifier.HasType.DataType*. Normally the data types will be *EnumerationType* or *IntegerType*.

More than one *Qualifier* can be defined for a single *DataType*. The meta-attribute *PrecedenceNumber* on the meta-entity *Qualifier* is used to determine the order of evaluation of the *Qualifiers*. For example a bounded two-dimensional array would have two instances of *BoundedArrayQualifier*, with the *BoundedArrayQualifier* instance expressing the first dimension having a *PrecedenceNumber* of ***1***, and the second *BoundedArrayQualifier* instance having a value of ***2***. In the case of unbounded one dimension array of pointers, it have ***UnboundedArrayQualifier*** with a *PrecedenceNumber* of ***1*** and a *PointerQualifier* with a *PrecedenceNumber* of ***2***.

6.6 Data Types

Simple data types are represented by *BasicDataType*. Since many data types are in common use, a number of predefined data types are provided as subtypes of *BasicDataType*. These pre-defined data types are shown in Figure 9 through Figure14. They may be used directly or subtyped further, using extensibility.

6.7 Constraining Data Type and Attribute Values

ValueDomains and *ValueDomainGroups* are used to constrain the values a data type can take; the domain is the valid set of values for a data type. A *ValueDomain* can specify a single value or set of values (via *ValueDomainEnumeration*), a range of values (via *ValueDomainRange*), a procedure used to generate the valid values (via *ValueDomainProcedure*), or the format for the value (via *ValueDomainRule*).

Examples of:

A set of values for ***DaysOfTheWeek*** is (***Sunday***, ***Monday***, ***Tuesday*** ...).

A range is (***0..100***).

A procedure is: ***do I=1 to 30, if I modulo 3 equals 1 then return I, end***.

A format rule is ***AANNN***.

The domain of a *DataType* should be described precisely by one of the subtypes of *ValueDomain*. However, when only a textual description of the domain is required, the meta-entity *ValueDomain* itself should be used.

The domain of an *AggregateDataType* describes the interaction of the individual *ValueDomains* of the *DataTypes* in the structure. Thus, the domain of an *AggregateDataType* should be described by a *ValueDomainGroup* containing only *ValueDomainProcedures* or *ValueDomain* meta-entities.

When the domain of a data type cannot be expressed by a single domain, the *ValueDomainGroup* is used to compose complex domain expressions. *ValueDomainGroups* can connect using the *Operator* meta-attribute values ***AND***, ***OR***, ***NOT***, or ***XOR,*** *ValueDomains* and *ValueDomainGroups* to form complex domain expressions.

An example of the use of *ValueDomainGroup* to build up a complex domain specification is shown on Figure 8.

The values defined by a domain must be congruent with the only one underlying data type.

For refining the domain of an *Attribute*, the instance of meta-relatiohship *DataType*.*TakesValuesFrom*.*ValueDomain* should be declared to the *DataType* instance for the *Attribute*. In this case, the restricted domain takes precedence over the domain that related to its DataType.

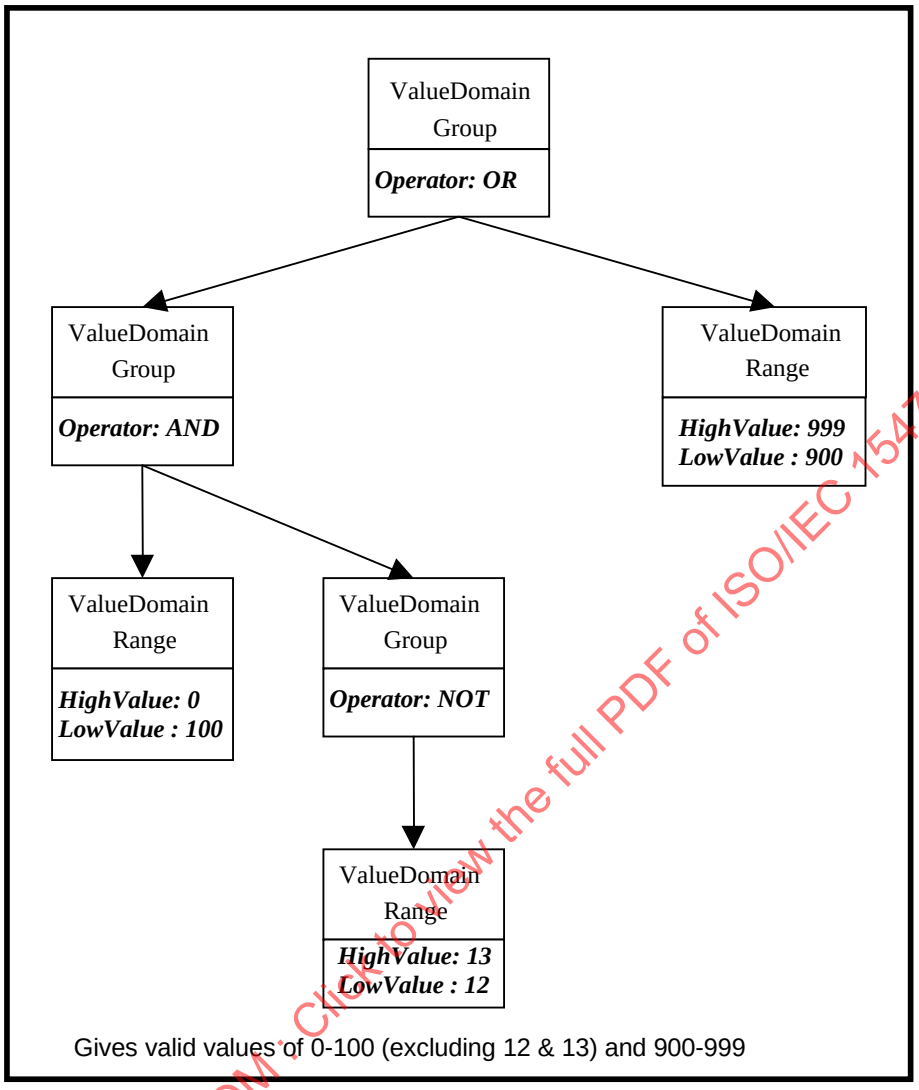

Figure 8 – Usage of ValueDomainGroup and ValueDomain

6.8 Units for Numeric DataTypes

The units of measurement for numeric data types can be expressed using the meta-entity *Unit*, related to the *NumericType* using the meta-relationship *NumericType.IsMeasuredIn.Unit*. The unit is expressed by defining the exponent for each of the SI concepts defined as a positive or negative integer.

NOTE: SI units are International System of Units defined in ISO 31-1:1992 *Quantities and units.* SI base units are meter, kilogram, second, ampere, kelvin, mole and candela.

If the unit is not actually expressed using the SI units, then the meta-attribute *IsSI* should be set to ***'FALSE'*** to indicate this; nevertheless the other meta-attributes should be set to indicate the concepts expressed by the unit. For example, acceleration with units of ms^{-2} ($Meter^{+1} * Second^{-2}$) should be expressed with the meta-attribute *ExponentForMeter* which express a meter unit and the meta-attribute *ExponentForSecond* which express a second unit set as follows for SI Units:

ExponentForMeter ***1***

ExponentForSecond ***-2***

IsSI ***TRUE***

If the acceleration is not expressed in SI units, then the meta-attribute *IsSI* sets to ***'FALSE'*** and the corresponding meta-attributes sets to the same values which expressed in SI units. For example ft per second per second, then the meta-attribute *IsSI* sets to ***'FALSE'*** , the meta-attribute *ExponentForMeter* sets to 1, the meta-attribute *ExponentForSecond* set to –2 and the meta-attributes *name* sets to feet/(second*second).

6.9 Void and Unknown Data Types

The *VoidType* is an explicit data type to be used in certain situations where the data type cannot otherwise be defined; an example of this is where the data type is a pointer to memory - the type that it points at is not defined, or may be of different types at different times, and therefore *VoidType* is used.

Where the data type of an attribute is not known, *VoidType* must not be used. Instead, no data type should be related to the attribute concerned; the cardinality of the *ComponentObject.References.DefinitionObject* has been defined to permit the absence of a data type where it is not known.

6.10 Computable Languages

The set of computable languages that can be supported is defined in Subclause 6.12 of ISO/IEC 15476-2:2002, *Information Technology - CDIF Semantic Metamodel - Part 2: Common*.

6.11 Formats for Computable Values

There are situations where it is necessary to declare a value that will serve as a default value or domain value for an attribute or data type. This subclause defines how such values are to be expressed. The value is held in a meta-attribute of type String, delimited by single quotes. The form to be used for the value depends on its type, and is defined in Table 1. The notation used for this is explained in

Table 2.

It is also permissible for values to be derived by execution of run-time functions. Where the value is derived by a function, such as the user-id of the user making the change, or Null, the value is expressed without quotes to indicate that it is a special function rather than an explicit value. Some of the possible values for such functions are listed in 3.

Table 1 – Formats for Computable Values

DATA TYPE	Format to be used for Value	Example
Binary	as a hexadecimal string. If the resulting pattern results in too many binary digits for the underlying type or attribute, excess trailing digits are ignored.	***'FF'***
Boolean	**TRUE** or **FALSE.**	***'TRUE'***

Complex	as two Real Numbers, separated by a colon (**:**).	***'1.125E3:0.5'*** ***'20:45'***
Decimal	[-]N(x)[**.**N(x)].	***'12.25'***
Enumerated	X(x).	***'Green'***
Float	as for Real.	***'123.56E11'***
Integer	[-]N(x).	***'763'***
Money	[-]N(x)[**.**N(x)][AAA] where AAA is an optional 3-character Currency Symbol, as defined in ISO 4217 Currency Symbols.	***'100.00USD'***
Real	[-]N(x)[**.**N(x)][***E***[**-**]N(x)].	***'736.3145'***
String	X(x).	***'Pair of Handcuffs'***
Temporal	YYYY**/**MM**/**DD , YYYY**/**MM**/**DD**-**HH**:**MM[**:**SS[**.**N(x)]] or HH**:**MM[**:**SS[**.**N(x)]] where : YYYY is the year MM is the numeric representation of the month as 2 digits in the range 01..12 DD is the numeric representation of the day as 2 digits in the range 01..31 HH is the hour shown as 2 digits in the range 00..23 MM is the minutes shown as 2 digits in the range 00..59 SS is the seconds shown as 2 digits in the range 00..59 N(x) are the fractional seconds to the degree of precision required.	***'1994/04/01-12:00'***

Table 2 – Notation Used for Computable Values

Symbol	Meaning
[]	Square brackets enclose optional elements in an expression. The portion of the expression within the brackets may be explicitly specified or may be omitted.
(x)	The (x) indicates that the element that it follows in a formula may be repeated any number of times (but note that optionality is covered by the square brackets convention). If it appears immediately after a closing square bracket "]", then it applies to the portion of the expression enclosed between that closing bracket and the corresponding opening bracket "[". If it appears after any other element, then it applies only to that element.
N	Any numeric character in the range 0..9.
A	Any alphabetic character.
X	Any printable character. (See *ISO/IEC 15475-3:2002, Information Technology - CDIF Transfer Format - Part 3: Encoding: ENCODING.1*)

Table 3 – Function Values for Computable Values

Value	Meaning
NULL	Null value.
USER	The identity of the current user.
CURRENT_DATE	The current date.
CURRENT_TIME	The current time.
CURRENT_TIMESTAMP	The current timestamp.

6.12 Diagrams

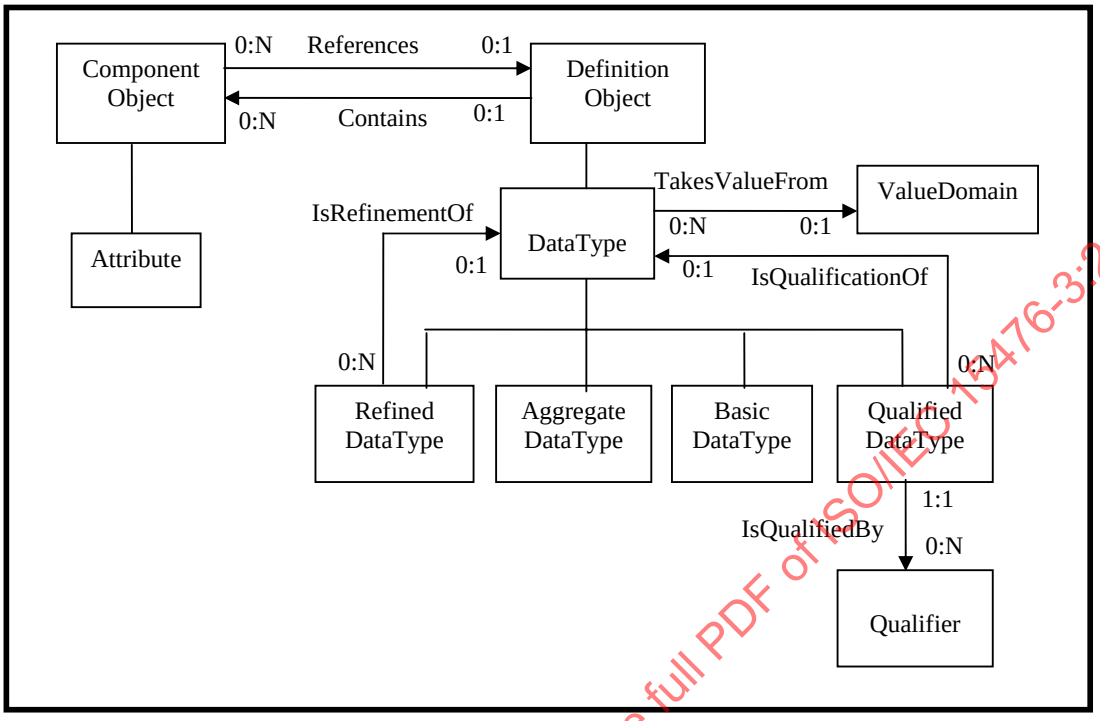

Figure 9 – Data Definition Subject Area - Main Diagram

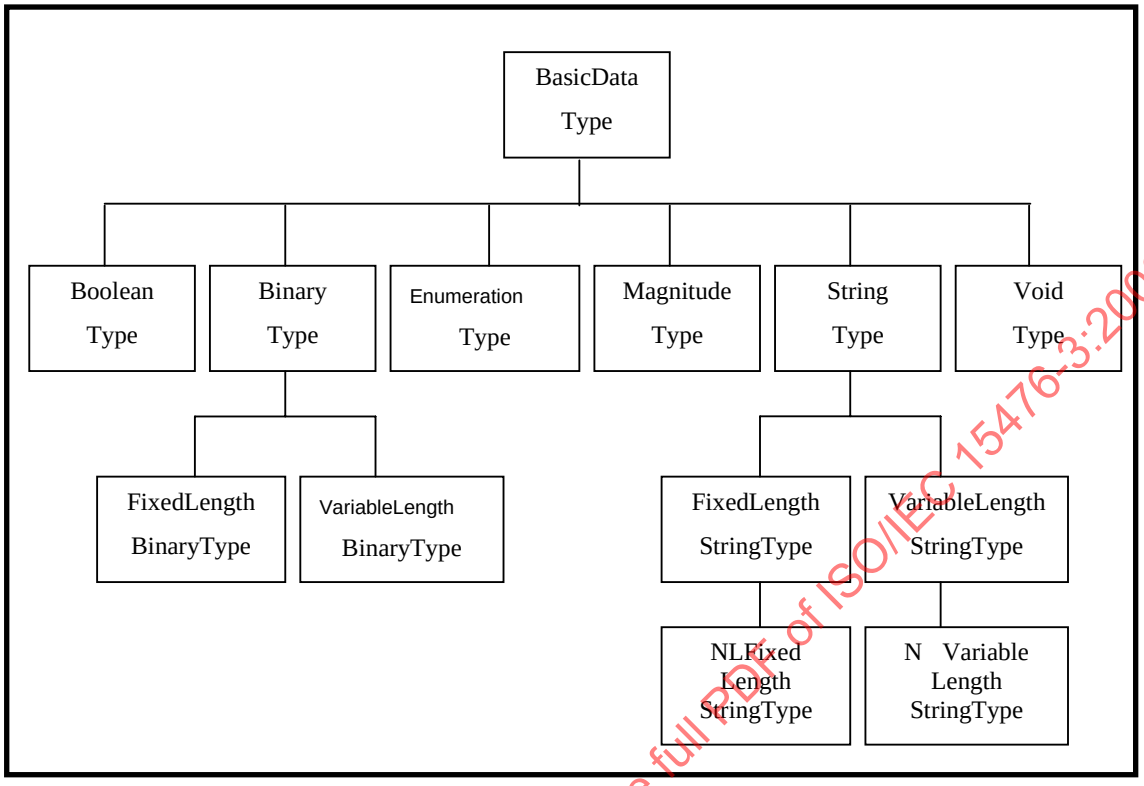

Figure 10 – BasicDataType Subtypes

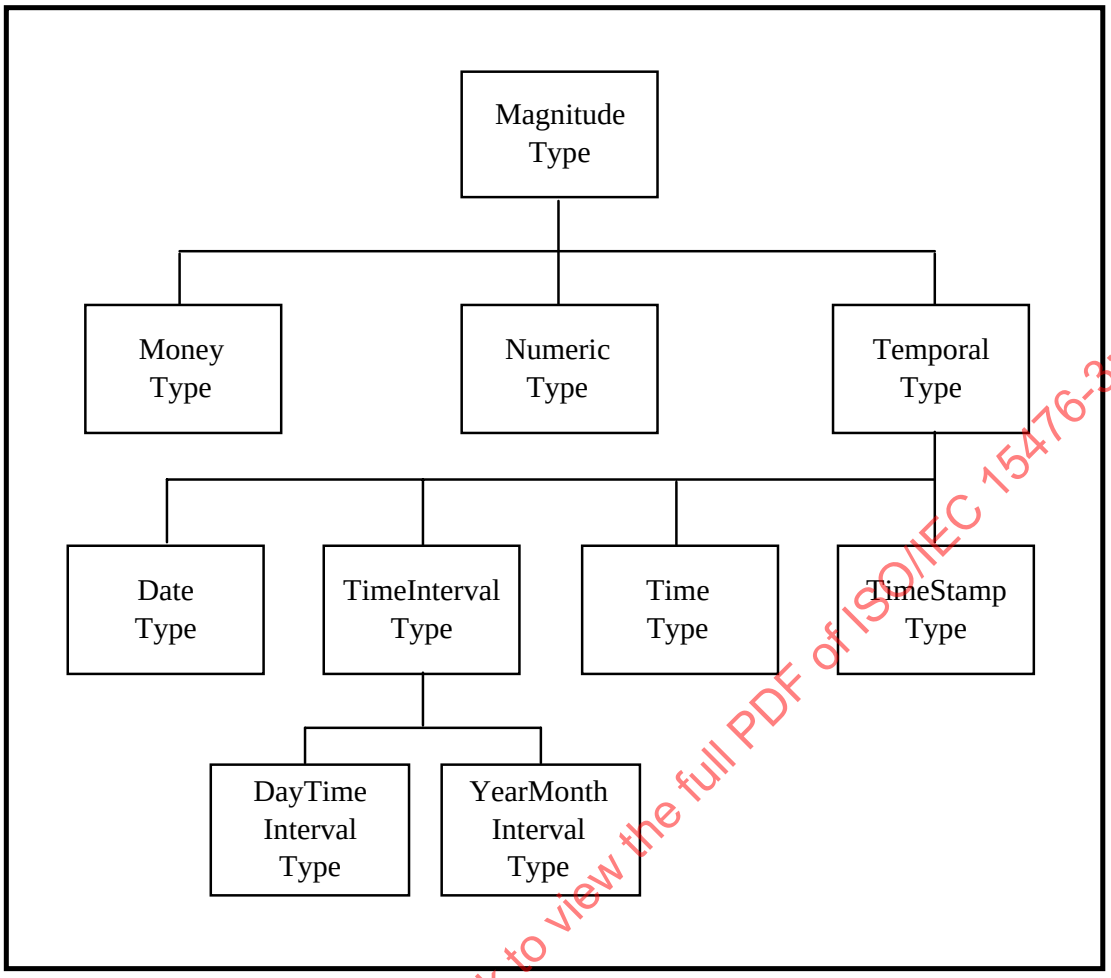

Figure 11 – MagnitudeType Subtypes

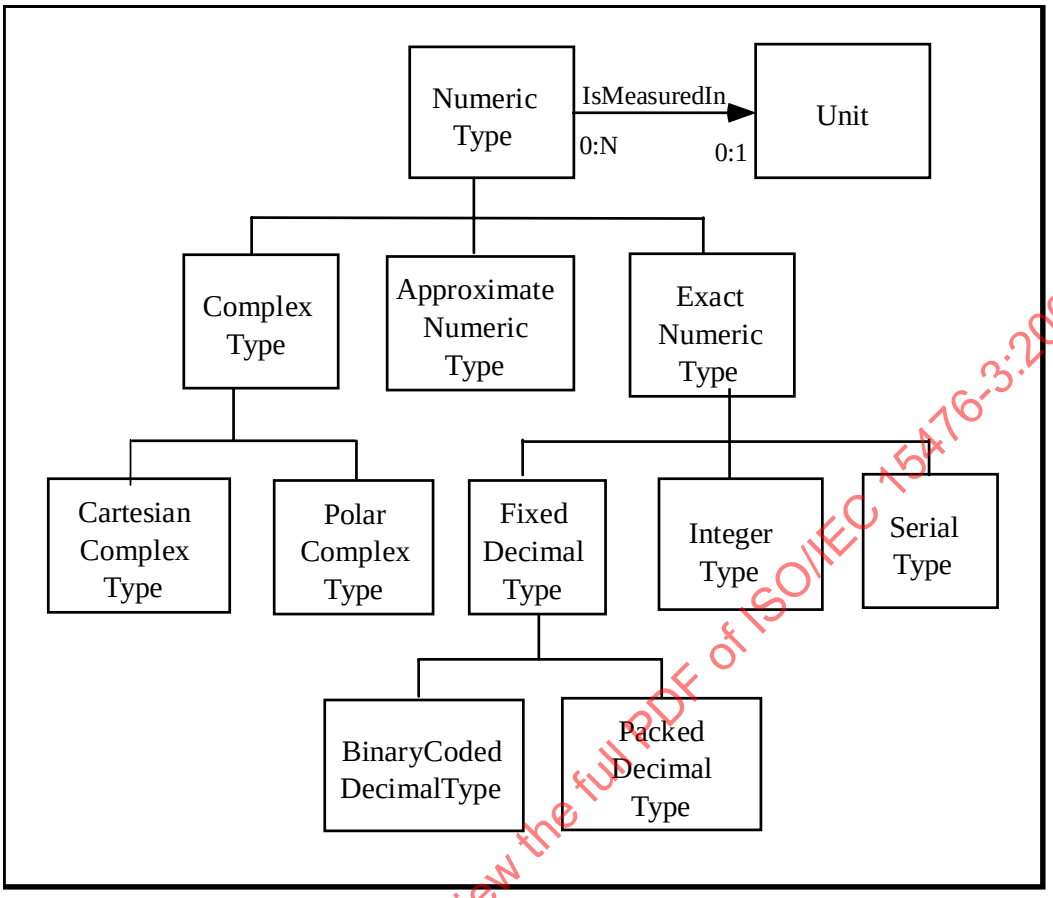

Figure 12 – NumericType Subtypes

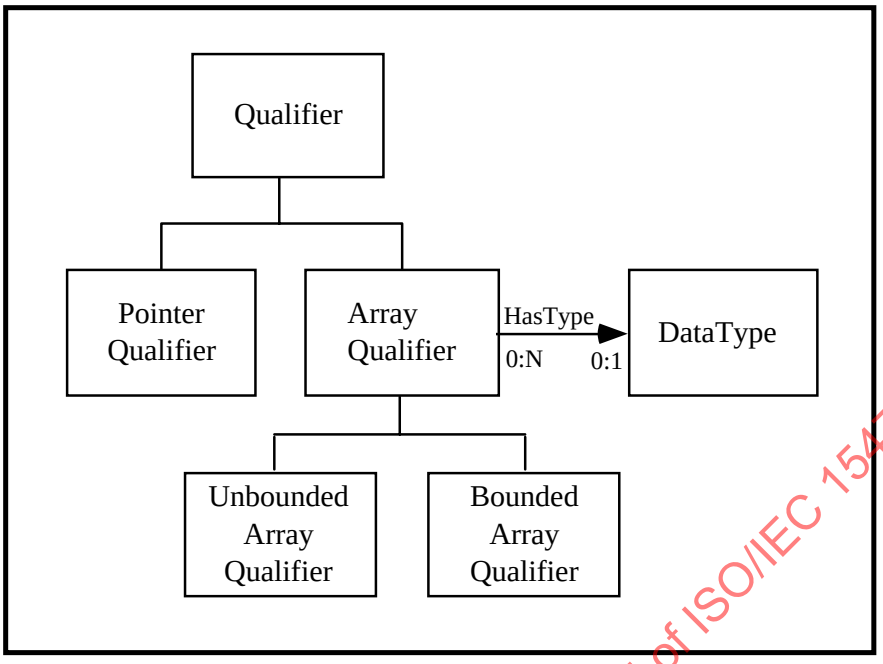

Figure 13 – Subtypes and meta-relationships for Qualifier

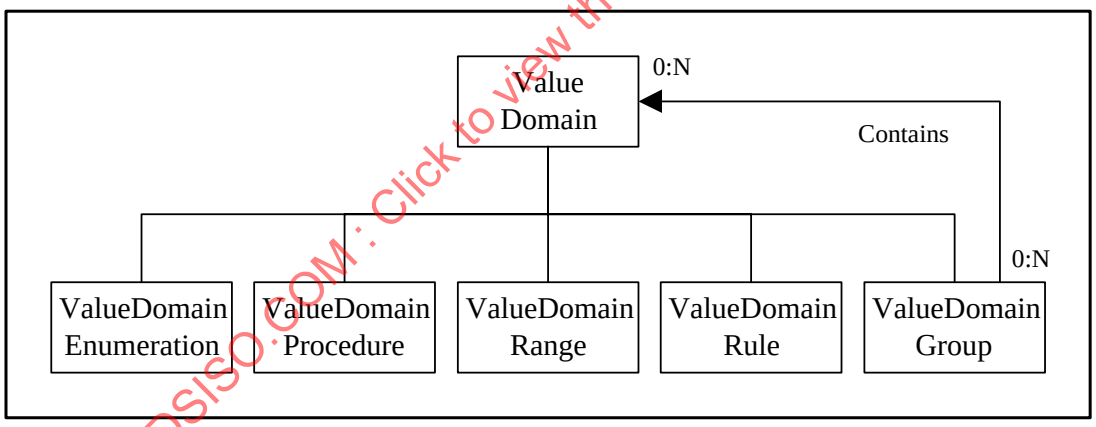

Figure 14 – ValueDomainGroup and related meta-entities

7 Data definitions subject area summary

7.1 AttributableMetaObject classification hierarchy

metaObject name	subjectArea name
RootEntity	Foundation
SemanticInformationObject	Common
ComponentObject	Common
Attribute	Data models
DefinitionObject	Common
DataType	Data definitions
AggregateDataType	Data definitions
BasicDataType	Data definitions
BinaryType	Data definitions
FixedLengthBinaryType	Data definitions
VariableLengthBinaryType	Data definitions
BooleanType	Data definitions
EnumerationType	Data definitions
MagnitudeType	Data definitions
MoneyType	Data definitions
NumericType	Data definitions
ApproximateNumericType	Data definitions
ComplexType	Data definitions
CartesianComplexType	Data definitions
PolarComplexType	Data definitions
ExactNumericType	Data definitions
IntegerType	Data definitions
FixedDecimalType	Data definitions
BinaryCodedDecimalType	Data definitions
PackedDecimalType	Data definitions
SerialType	Data definitions
TemporalType	Data definitions
DateType	Data definitions
TimeIntervalType	Data definitions
DayTimeIntervalType	Data definitions
YearMonthIntervalType	Data definitions
TimeStampType	Data definitions
TimeType	Data definitions
StringType	Data definitions
FixedLengthStringType	Data definitions
NLFixedLengthStringType	Data definitions
VariableLengthStringType	Data definitions
NLVariableLengthStringType	Data definitions
VoidType	Data definitions

MetaObject name	SubjectArea name
RootEntity	Foundation
SemanticInformationObject	Common
DefinitionObject	Common
DataType	Data definitions
QualifiedDataType	Data definitions
RefinedDataType	Data definitions
Qualifier	Data definitions
ArrayQualifier	Data definitions
BoundedArrayQualifier	Data definitions
UnboundedArrayQualifier	Data definitions
PointerQualifier	Data definitions
Unit	Data definitions
ValueDomain	Data definitions
ValueDomainEnumeration	Data definitions
ValueDomainGroup	Data definitions
ValueDomainProcedure	Data definitions
ValueDomainRange	Data definitions
ValueDomainRule	Data definitions
RootEntity.IsRelatedTo.RootEntity	Foundation
ArrayQualifier.HasType.DataType	Data definitions
DataType.TakesValueFrom.ValueDomain	Data definitions
NumericType.IsMeasuredIn.Unit	Data definitions
QualifiedDataType.IsQualificationOf.DataType	Data definitions
QualifiedDataType.IsQualifiedBy.Qualifier	Data definitions
RefinedDataType.IsRefinementOf.DataType	Data definitions
ValueDomainGroup.Contains.ValueDomain	Data definitions

7.2 MetaEntity summary

The inherited meta-attributes: CDIFIdentifier, DateCreated, DateUpdated, TimeCreated, TimeUpdated, which defined in RootEntity are not described at the following EetaEntity summary.

AggregateDataType	
BriefDescription	*Optional*
FormatStringLanguage	*Optional*
FormatStringValue	*Optional*
FullDescription	*Optional*
Name	*Optional*
Operator	*Optional*
SpecificationLanguage	*Optional*
SpecificationText	*Optional*
Usage	*Optional*

ApproximateNumericType	
BriefDescription	*Optional*
FormatStringLanguage	*Optional*
FormatStringValue	*Optional*
FullDescription	*Optional*
Name	*Optional*
Operator	*Optional*
SpecificationLanguage	*Optional*
SpecificationText	*Optional*
Usage	*Optional*
Precision	Optional
Scale	Optional

ArrayQualifier	
BriefDescription	*Optional*
FullDescription	*Optional*
PrecedenceNumber	*Mandatory*

BasicDataType	
BriefDescription	*Optional*
FormatStringLanguage	*Optional*
FormatStringValue	*Optional*
FullDescription	*Optional*
Name	*Optional*
Operator	*Optional*
SpecificationLanguage	*Optional*
SpecificationText	*Optional*
Usage	*Optional*

BinaryCodedDecimalType

BriefDescription	*Optional*
FormatStringLanguage	*Optional*
FormatStringValue	*Optional*
FullDescription	*Optional*
Name	*Optional*
Operator	*Optional*
Precision	*Optional*
Scale	*Optional*
SignedFlag	*Optional*
SpecificationLanguage	*Optional*
SpecificationText	*Optional*
Usage	*Optional*

BinaryType

BriefDescription	*Optional*
FormatStringLanguage	*Optional*
FormatStringValue	*Optional*
FullDescription	*Optional*
Name	*Optional*
Operator	*Optional*
SpecificationLanguage	*Optional*
SpecificationText	*Optional*
Usage	*Optional*

BooleanType

BriefDescription	*Optional*
FormatStringLanguage	*Optional*
FormatStringValue	*Optional*
FullDescription	*Optional*
Name	*Optional*
Operator	*Optional*
SpecificationLanguage	*Optional*
SpecificationText	*Optional*
Usage	*Optional*

BoundedArrayQualifier

BriefDescription	*Optional*
FullDescription	*Optional*
PrecedenceNumber	*Mandatory*
MaxSubscript	Optional
MinSubscript	Mandatory

CartesianComplexType	
BriefDescription	*Optional*
FormatStringLanguage	*Optional*
FormatStringValue	*Optional*
FullDescription	*Optional*
Name	*Optional*
Operator	*Optional*
SpecificationLanguage	*Optional*
SpecificationText	*Optional*
Usage	*Optional*
Precision	Optional
Scale	Optional

ComplexType	
BriefDescription	*Optional*
FormatStringLanguage	*Optional*
FormatStringValue	*Optional*
FullDescription	*Optional*
Name	*Optional*
Operator	*Optional*
SpecificationLanguage	*Optional*
SpecificationText	*Optional*
Usage	*Optional*

DataType	
BriefDescription	*Optional*
FullDescription	*Optional*
Name	*Optional*
Operator	*Optional*
SpecificationLanguage	*Optional*
SpecificationText	*Optional*
FormatStringLanguage	Optional
FormatStringValue	Optional
Usage	Optional

DateType	
BriefDescription	*Optional*
FormatStringLanguage	*Optional*
FormatStringValue	*Optional*
FullDescription	*Optional*
Name	*Optional*
Operator	*Optional*
SpecificationLanguage	*Optional*
SpecificationText	*Optional*
Usage	*Optional*

DayTimeIntervalType	
BriefDescription	*Optional*
FormatStringLanguage	*Optional*
FormatStringValue	*Optional*
FullDescription	*Optional*
Name	*Optional*
Operator	*Optional*
SpecificationLanguage	*Optional*
SpecificationText	*Optional*
Usage	*Optional*

EnumerationType	
BriefDescription	*Optional*
FormatStringLanguage	*Optional*
FormatStringValue	*Optional*
FullDescription	*Optional*
Name	*Optional*
Operator	*Optional*
SpecificationLanguage	*Optional*
SpecificationText	*Optional*
Usage	*Optional*

ExactNumericType	
BriefDescription	*Optional*
FormatStringLanguage	*Optional*
FormatStringValue	*Optional*
FullDescription	*Optional*
Name	*Optional*
Operator	*Optional*
SpecificationLanguage	*Optional*
SpecificationText	*Optional*
Usage	*Optional*
SignedFlag	Optional

FixedDecimalType

BriefDescription	*Optional*
FormatStringLanguage	*Optional*
FormatStringValue	*Optional*
FullDescription	*Optional*
Name	*Optional*
Operator	*Optional*
SignedFlag	*Optional*
SpecificationLanguage	*Optional*
SpecificationText	*Optional*
Usage	*Optional*
Precision	Optional
Scale	Optional

FixedLengthBinaryType

BriefDescription	*Optional*
FormatStringLanguage	*Optional*
FormatStringValue	*Optional*
FullDescription	*Optional*
Name	*Optional*
Operator	*Optional*
SpecificationLanguage	*Optional*
SpecificationText	*Optional*
Usage	*Optional*
Length	Optional
LengthMultiplier	Optional

FixedLengthStringType

BriefDescription	*Optional*
CharacterSet	*Optional*
FormatStringLanguage	*Optional*
FormatStringValue	*Optional*
FullDescription	*Optional*
Name	*Optional*
Operator	*Optional*
SpecificationLanguage	*Optional*
SpecificationText	*Optional*
StringEncoding	*Optional*
Usage	*Optional*
Length	Optional
LengthMultiplier	Optional

IntegerType	
BriefDescription	*Optional*
FormatStringLanguage	*Optional*
FormatStringValue	*Optional*
FullDescription	*Optional*
Name	*Optional*
Operator	*Optional*
SignedFlag	*Optional*
SpecificationLanguage	*Optional*
SpecificationText	*Optional*
Usage	*Optional*

MagnitudeType	
BriefDescription	*Optional*
FormatStringLanguage	*Optional*
FormatStringValue	*Optional*
FullDescription	*Optional*
Name	*Optional*
Operator	*Optional*
SpecificationLanguage	*Optional*
SpecificationText	*Optional*
Usage	*Optional*

MoneyType	
BriefDescription	*Optional*
FormatStringLanguage	*Optional*
FormatStringValue	*Optional*
FullDescription	*Optional*
Name	*Optional*
Operator	*Optional*
SpecificationLanguage	*Optional*
SpecificationText	*Optional*
Usage	*Optional*
Currency	Optional
Precision	Optional
Scale	Optional

NLFixedLengthStringType	
BriefDescription	*Optional*
CharacterSet	*Optional*
FormatStringLanguage	*Optional*
FormatStringValue	*Optional*
FullDescription	*Optional*
Length	*Optional*

LengthMultiplier	*Optional*
Name	*Optional*
Operator	*Optional*
SpecificationLanguage	*Optional*
SpecificationText	*Optional*
StringEncoding	*Optional*
Usage	*Optional*
BitsPerCharacter	Optional

NLVariableLengthStringType

BriefDescription	*Optional*
CharacterSet	*Optional*
FormatStringLanguage	*Optional*
FormatStringValue	*Optional*
FullDescription	*Optional*
LengthMultiplier	*Optional*
MaxLength	*Optional*
Name	*Optional*
Operator	*Optional*
SpecificationLanguage	*Optional*
SpecificationText	*Optional*
StringEncoding	*Optional*
Usage	*Optional*
BitsPerCharacter	Optional

NumericType

BriefDescription	*Optional*
FormatStringLanguage	*Optional*
FormatStringValue	*Optional*
FullDescription	*Optional*
Name	*Optional*
Operator	*Optional*
SpecificationLanguage	*Optional*
SpecificationText	*Optional*
Usage	*Optional*

PackedDecimalType	
BriefDescription	*Optional*
FormatStringLanguage	*Optional*
FormatStringValue	*Optional*
FullDescription	*Optional*
Name	*Optional*
Operator	*Optional*
Precision	*Optional*
Scale	*Optional*
SignedFlag	*Optional*
SpecificationLanguage	*Optional*
SpecificationText	*Optional*
Usage	*Optional*

PointerQualifier	
BriefDescription	*Optional*
FullDescription	*Optional*
PrecedenceNumber	*Mandatory*

PolarComplexType	
BriefDescription	*Optional*
FormatStringLanguage	*Optional*
FormatStringValue	*Optional*
FullDescription	*Optional*
Name	*Optional*
Operator	*Optional*
SpecificationLanguage	*Optional*
SpecificationText	*Optional*
Usage	*Optional*
DistancePrecision	Optional
DistanceScale	Optional
GradientPrecision	Optional
GradientScale	Optional

QualifiedDataType	
BriefDescription	*Optional*
FormatStringLanguage	*Optional*
FormatStringValue	*Optional*
FullDescription	*Optional*
Name	*Optional*
Operator	*Optional*
SpecificationLanguage	*Optional*
SpecificationText	*Optional*
Usage	*Optional*

Qualifier	
BriefDescription	*Optional*
FullDescription	*Optional*
PrecedenceNumber	Mandatory

RefinedDataType	
BriefDescription	*Optional*
FormatStringLanguage	*Optional*
FormatStringValue	*Optional*
FullDescription	*Optional*
Name	*Optional*
Operator	*Optional*
SpecificationLanguage	*Optional*
SpecificationText	*Optional*
Usage	*Optional*

SerialType	
BriefDescription	*Optional*
FormatStringLanguage	*Optional*
FormatStringValue	*Optional*
FullDescription	*Optional*
Name	*Optional*
Operator	*Optional*
SignedFlag	*Optional*
SpecificationLanguage	*Optional*
SpecificationText	*Optional*
Usage	*Optional*
Cycle	Optional
Interval	Optional
StartingValue	Optional

StringType	
BriefDescription	*Optional*
FormatStringLanguage	*Optional*
FormatStringValue	*Optional*
FullDescription	*Optional*
Name	*Optional*
Operator	*Optional*
SpecificationLanguage	*Optional*
SpecificationText	*Optional*
Usage	*Optional*
CharacterSet	Optional
StringEncoding	Optional

TemporalType	
BriefDescription	*Optional*
FormatStringLanguage	*Optional*
FormatStringValue	*Optional*
FullDescription	*Optional*
Name	*Optional*
Operator	*Optional*
SpecificationLanguage	*Optional*
SpecificationText	*Optional*
Usage	*Optional*

TimeIntervalType	
BriefDescription	*Optional*
FormatStringLanguage	*Optional*
FormatStringValue	*Optional*
FullDescription	*Optional*
Name	*Optional*
Operator	*Optional*
SpecificationLanguage	*Optional*
SpecificationText	*Optional*
Usage	*Optional*

TimeStampType	
BriefDescription	*Optional*
FormatStringLanguage	*Optional*
FormatStringValue	*Optional*
FullDescription	*Optional*
Name	*Optional*
Operator	*Optional*
SpecificationLanguage	*Optional*
SpecificationText	*Optional*
Usage	*Optional*
IsLocal	Optional
TimeZoneHours	Optional
TimeZoneMinutes	Optional

TimeType	
BriefDescription	*Optional*
FormatStringLanguage	*Optional*
FormatStringValue	*Optional*
FullDescription	*Optional*
Name	*Optional*
Operator	*Optional*
SpecificationLanguage	*Optional*
SpecificationText	*Optional*
Usage	*Optional*
IsLocal	Optional
TimeZoneHours	Optional
TimeZoneMinutes	Optional

UnboundedArrayQualifier	
BriefDescription	*Optional*
FullDescription	*Optional*
PrecedenceNumber	*Mandatory*

Unit	
BriefDescription	*Optional*
FullDescription	*Optional*
ExponentForAmpere	Optional
ExponentForCandela	Optional
ExponentForKelvin	Optional
ExponentForKilogram	Optional
ExponentForMeter	Optional
ExponentForMole	Optional
ExponentForSecond	Optional
IsSI	Optional
Name	Optional

ValueDomain	
BriefDescription	*Optional*
FullDescription	*Optional*
Name	Optional

ValueDomainEnumeration	
BriefDescription	*Optional*
FullDescription	*Optional*
Name	*Optional*
Value	Mandatory

ValueDomainGroup	
BriefDescription	*Optional*
FullDescription	*Optional*
Name	*Optional*
Operator	Mandatory

ValueDomainProcedure	
BriefDescription	*Optional*
FullDescription	*Optional*
Name	*Optional*
ProcedureName	Optional
SpecificationLanguage	Optional
SpecificationText	Optional

ValueDomainRange	
BriefDescription	*Optional*
FullDescription	*Optional*
Name	*Optional*
HighValue	Optional
HighValueIncluded	Optional
LowValue	Optional
LowValueIncluded	Optional

ValueDomainRule	
BriefDescription	*Optional*
FullDescription	*Optional*
Name	*Optional*
SpecificationLanguage	Optional
SpecificationString	Optional

VariableLengthBinaryType	
BriefDescription	*Optional*
FormatStringLanguage	*Optional*
FormatStringValue	*Optional*
FullDescription	*Optional*
Name	*Optional*
Operator	*Optional*
SpecificationLanguage	*Optional*
SpecificationText	*Optional*
Usage	*Optional*
LengthMultiplier	Optional
MaxLength	Optional

VariableLengthStringType	
BriefDescription	*Optional*
CharacterSet	*Optional*
FormatStringLanguage	*Optional*
FormatStringValue	*Optional*
FullDescription	*Optional*
Name	*Optional*
Operator	*Optional*
SpecificationLanguage	*Optional*
SpecificationText	*Optional*
StringEncoding	*Optional*
Usage	*Optional*
LengthMultiplier	Optional
MaxLength	Optional

VoidType	
BriefDescription	*Optional*
FormatStringLanguage	*Optional*
FormatStringValue	*Optional*
FullDescription	*Optional*
Name	*Optional*
Operator	*Optional*
SpecificationLanguage	*Optional*
SpecificationText	*Optional*
Usage	*Optional*

YearMonthIntervalType	
BriefDescription	*Optional*
FormatStringLanguage	*Optional*
FormatStringValue	*Optional*
FullDescription	*Optional*
Name	*Optional*
Operator	*Optional*
SpecificationLanguage	*Optional*
SpecificationText	*Optional*
Usage	*Optional*

7.3 MetaRelationship summary

ArrayQualifier.HasType.DataType	
CDIFIdentifier	*Mandatory*
DateCreated	*Optional*
DateUpdated	*Optional*
TimeCreated	*Optional*
TimeUpdated	*Optional*

DataType.TakesValueFrom.ValueDomainGroup	
CDIFIdentifier	*Mandatory*
DateCreated	*Optional*
DateUpdated	*Optional*
TimeCreated	*Optional*
TimeUpdated	*Optional*

NumericType.IsMeasuredIn.Unit	
CDIFIdentifier	*Mandatory*
DateCreated	*Optional*
DateUpdated	*Optional*
TimeCreated	*Optional*
TimeUpdated	*Optional*

QualifiedDataType.IsQualificationOf.DataType	
CDIFIdentifier	*Mandatory*
DateCreated	*Optional*
DateUpdated	*Optional*
TimeCreated	*Optional*
TimeUpdated	*Optional*

QualifiedDataType.IsQualifiedBy.Qualifier	
CDIFIdentifier	*Mandatory*
DateCreated	*Optional*
DateUpdated	*Optional*
TimeCreated	*Optional*
TimeUpdated	*Optional*

RefinedDataType.IsRefinementOf.DataType	
CDIFIdentifier	*Mandatory*
DateCreated	*Optional*
DateUpdated	*Optional*
TimeCreated	*Optional*
TimeUpdated	*Optional*

ValueDomainGroup.Contains.ValueDomain	
CDIFIdentifier	*Mandatory*
DateCreated	*Optional*
DateUpdated	*Optional*
TimeCreated	*Optional*
TimeUpdated	*Optional*

8 Data definitions subject area specification

8.1 Introduction

This clause provides the full definition of each object used in the Data definitions subject area of the CDIF semantic metamodel.

8.1.1 Subject area definition

SUBJECT AREA DEFINITION	
NAME	**Data Definitions**
VERSIONNUMBER	15476-3:2003
CDIFMETAIDENTIFIER	763000
DESCRIPTION	The Data definitions subject area is intended to cover aspects that are basic data type and structuring data to many CASE tools.
USAGE	It provides the objects that are used as to allow the definition of data structures and elements that are used to define the data content of objects defined in other subject areas.
ALIASES	
CONSTRAINTS	

8.2 Meta-entity definitions

8.2.1 AggregateDataType

META-ENTITY DEFINITION	
NAME	AggregateDataType
CDIFMETAIDENTIFIER	763001
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This is a data type thst is either structured or decomposed. *AggregateDataTypes* contain component *Attributes*.
USAGE	
ALIASES	
CONSTRAINTS	
TYPE	Kernel
ISABSTRACT	FALSE
LOCAL SUBTYPES	
LOCAL METARELATIONSHIPS	
LOCAL METAATTRIBUTES	

8.2.2 ApproximateNumericType

META-ENTITY DEFINITION	
NAME	ApproximateNumericType
CDIFMETAIDENTIFIER	763002
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This subtype defines numeric types whose values are inexact, or approximate.
USAGE	
ALIASES	Real
CONSTRAINTS	
TYPE	Kernel
ISABSTRACT	FALSE
LOCAL SUBTYPES	
LOCAL METARELATIONSHIPS	
LOCAL METAATTRIBUTES	Precision Scale

META-ATTRIBUTE DEFINITION	META-ATTRIBUTE OF ApproximateNumericType
NAME	Precision
CDIFMETAIDENTIFIER	763003
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This defines the maximum number of digits that can be supplied for the type, indicating the precision to which it can be specified.
USAGE	
ALIASES	
CONSTRAINTS	
DATA TYPE	Integer
DOMAIN	
LENGTH	
ISOPTIONAL	TRUE

META-ATTRIBUTE DEFINITION	META-ATTRIBUTE OF ApproximateNumericType
NAME	Scale
CDIFMETAIDENTIFIER	763004
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This defines the implicit position of the decimal point. This may be negative to indicate that the decimal point is implicitly to the right of the last digit, for example where the values stored are all thousands, the value of Scale would be -3; therefore a value of 7 would be interpreted as 7000.
USAGE	
ALIASES	
CONSTRAINTS	
DATA TYPE	Integer
DOMAIN	
LENGTH	
ISOPTIONAL	TRUE

8.2.3 ArrayQualifier

META-ENTITY DEFINITION	
NAME	ArrayQualifier
CDIFMETAIDENTIFIER	763005
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This provides a mechanism to describe the dimensions of an array.
USAGE	This object should not be instantiated unless it is not known whether a dimension is bounded or unbounded; where this is known, the appropriate subtype should be used.
ALIASES	
CONSTRAINTS	
TYPE	Characteristic
ISABSTRACT	FALSE
LOCAL SUBTYPES	UnboundedArrayQualifier BoundedArrayQualifier
LOCAL METARELATIONSHIPS	ArrayQualifier.HasType.DataType
LOCAL METAATTRIBUTES	

8.2.4 Attribute

META-ENTITY REFERENCE	
NAME	Attribute
CDIFMETAIDENTIFIER	17
SUBJECTAREANAME	Data models
SUBJECTAREAVERSION	15476-3:2003
LOCAL SUBTYPES	
LOCAL METARELATIONSHIPS	
LOCAL METAATTRIBUTES	

NOTE: The meta-attributes of the meta-entity Attibute, DefaultValue and IsOptional, are not defined in this subject area. They are defined in *Data models*.

8.2.5 BasicDataType

META-ENTITY DEFINITION	
NAME	BasicDataType
CDIFMETAIDENTIFIER	763006
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-4:2003
DESCRIPTION	This exists to define an elemental (primitive) type. This is a type that cannot be decomposed into further structural components.
USAGE	
ALIASES	PrimitiveType AtomicDataType
CONSTRAINTS	
TYPE	Kernel
ISABSTRACT	FALSE
LOCAL SUBTYPES	BinaryType BooleanType EnumerationType MagnitudeType StringType VoidType
LOCAL METARELATIONSHIPS	
LOCAL METAATTRIBUTES	

8.2.6 BinaryCodedDecimalType

META-ENTITY DEFINITION	
NAME	BinaryCodedDecimalType
CDIFMETAIDENTIFIER	763007
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This is the base data type for binary coded decimal data types.
USAGE	
ALIASES	
CONSTRAINTS	
TYPE	Kernel
ISABSTRACT	FALSE
LOCAL SUBTYPES	
LOCAL METARELATIONSHIPS	
LOCAL METAATTRIBUTES	

8.2.7 BinaryType

META-ENTITY DEFINITION	
NAME	BinaryType
CDIFMETAIDENTIFIER	763011
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This describes a binary data type
USAGE	This type should not be instantiated; one of the subtypes should be used where possible.
ALIASES	
CONSTRAINTS	
TYPE	Kernel
ISABSTRACT	FALSE
LOCAL SUBTYPES	FixedLengthBinaryType VariableLengthBinaryType
LOCAL METARELATIONSHIPS	
LOCAL METAATTRIBUTES	

8.2.8 BooleanType

META-ENTITY DEFINITION	
NAME	BooleanType
CDIFMETAIDENTIFIER	763008
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This defines a Boolean data type.
USAGE	
ALIASES	
CONSTRAINTS	
TYPE	Kernel
ISABSTRACT	FALSE
LOCAL SUBTYPES	
LOCAL METARELATIONSHIPS	
LOCAL METAATTRIBUTES	

8.2.9 BoundedArrayQualifier

META-ENTITY DEFINITION	
NAME	BoundedArrayQualifier
CDIFMETAIDENTIFIER	763009
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This subtype of *ArrayQualifier* is used where a dimension of an array is bounded.
USAGE	This subtype is used where it is known that the dimension of the array is bounded, even if the actual values of the bounds are not known.
ALIASES	
CONSTRAINTS	
TYPE	Characteristic
ISABSTRACT	FALSE
LOCAL SUBTYPES	
LOCAL METARELATIONSHIPS	ArrayQualifier.HasType.DataType
LOCAL METAATTRIBUTES	MaxSubscript MinSubscript

META-ATTRIBUTE DEFINITION	**META-ATTRIBUTE OF** BoundedArrayQualifier
NAME	MaxSubscript
CDIFMETAIDENTIFIER	763010
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This describes the maximum subscript of an array.
USAGE	
ALIASES	
CONSTRAINTS	The value should be equal or greater than the value of the meta-attribute *MinSubscript,* if both are present.
DATA TYPE	Integer
DOMAIN	
LENGTH	
ISOPTIONAL	TRUE

META-ATTRIBUTE DEFINITION	META-ATTRIBUTE OF BoundedArrayQualifier
NAME	**MinSubscript**
CDIFMETAIDENTIFIER	763011
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This describes the minimum subscript of an array.
USAGE	An array of anything in C would cause this value to be 0. In languages like Pascal, the minimum and maximum bounds can be specified.
ALIASES	
CONSTRAINTS	
DATA TYPE	Integer
DOMAIN	
LENGTH	
ISOPTIONAL	FALSE

8.2.10 CartesianComplexType

META-ENTITY DEFINITION	
NAME	CartesianComplexType
CDIFMETAIDENTIFIER	763012
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This subtype defines a type for expressing complex numbers in Cartesian form.
USAGE	
ALIASES	
CONSTRAINTS	
TYPE	Kernel
ISABSTRACT	FALSE
LOCAL SUBTYPES	
LOCAL METARELATIONSHIPS	
LOCAL METAATTRIBUTES	Precision Scale

META-ATTRIBUTE DEFINITION	META-ATTRIBUTE OF CartesianComplexType
NAME	Precision
CDIFMETAIDENTIFIER	763013
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This defines the maximum number of digits that can be supplied for the type, indicating the precision to which each part can be specified. The value of *Precision* applies to both the real and the imaginary parts.
USAGE	
ALIASES	
CONSTRAINTS	
DATA TYPE	Integer
DOMAIN	
LENGTH	
ISOPTIONAL	TRUE

META-ATTRIBUTE DEFINITION	META-ATTRIBUTE OF CartesianComplexType
NAME	Scale
CDIFMETAIDENTIFIER	763014
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This defines the implicit position of the decimal point. This may be negative to indicate that the decimal point is implicitly to the right of the last digit, for example where the values stored are all thousands, the value of *Scale* would be -3; therefore a value of 7 would be interpreted as 7000. It applies to each of the real and the imaginary parts.
USAGE	
ALIASES	
CONSTRAINTS	
DATA TYPE	Integer
DOMAIN	
LENGTH	
ISOPTIONAL	TRUE

8.2.11 ComplexType

META-ENTITY DEFINITION	
NAME	ComplexType
CDIFMETAIDENTIFIER	763015
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This subtype defines a type for representing complex numbers.
USAGE	
ALIASES	
CONSTRAINTS	This subtype shall only be used when the form of expression of a complex type is not known. Where the type is known to be either polar or cartesian, the appropriate subtype shall be used.
TYPE	Kernel
ISABSTRACT	TRUE
LOCAL SUBTYPES	CartesianComplexType PolarComplexType
LOCAL METARELATIONSHIPS	
LOCAL METAATTRIBUTES	

8.2.12 DataType

META-ENTITY DEFINITION	
NAME	DataType
CDIFMETAIDENTIFIER	763016
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This holds the definition, structure, or decomposition of an *Attribute*.
USAGE	
ALIASES	
CONSTRAINTS	
TYPE	Kernel
ISABSTRACT	TRUE
LOCAL SUBTYPES	AggregateDataType BasicDataType QualifiedDataType RefinedDataType
LOCAL METARELATIONSHIPS	ArrayQualifier.HasType.DataType DataType.TakesValueFrom.ValueDomainGroup QualifiedDataType.IsQualificationOf.DataType RefinedDataType.IsRefinementOf.DataType
LOCAL METAATTRIBUTES	FormatStringLanguage FormatStringValue Usage

META-ATTRIBUTE DEFINITION	META-ATTRIBUTE OF DataType
NAME	FormatStringLanguage
CDIFMETAIDENTIFIER	763017
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This defines the language used for the meta-attribute *FormatStringValue.*
USAGE	
ALIASES	
CONSTRAINTS	This shall be supplied when a value is supplied for the meta-attribute *FormatStringValue*.
DATA TYPE	Enumerated
DOMAIN	As specified in Subclause 6.12 of ISO/IEC 15476-2:2002, *Information Technology - CDIF Semantic Metamodel - Part 2: Common*.
LENGTH	
ISOPTIONAL	TRUE

META-ATTRIBUTE DEFINITION	META-ATTRIBUTE OF DataType
NAME	FormatStringValue
CDIFMETAIDENTIFIER	763018
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This is a string giving format information for the data type.
USAGE	For example, using COBOL, Z,ZZN would indicate a 1-4 digit number, with leading zeros suppressed, and a comma separator for the thousands.
ALIASES	
CONSTRAINTS	
DATA TYPE	String
DOMAIN	
LENGTH	1024
ISOPTIONAL	TRUE

META-ATTRIBUTE DEFINITION	META-ATTRIBUTE OF DataType
NAME	Usage
CDIFMETAIDENTIFIER	763019
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This describes how the *DataType* is used.
USAGE	
ALIASES	
CONSTRAINTS	
DATA TYPE	Text
DOMAIN	
LENGTH	
ISOPTIONAL	TRUE

8.2.13 DateType

META-ENTITY DEFINITION	
NAME	DateType
CDIFMETAIDENTIFIER	763020
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This defines a type used to represent a date.
USAGE	
ALIASES	
CONSTRAINTS	
TYPE	Kernel
ISABSTRACT	FALSE
LOCAL SUBTYPES	
LOCAL METARELATIONSHIPS	
LOCAL METAATTRIBUTES	

8.2.14 DayTimeIntervalType

META-ENTITY DEFINITION	
NAME	DayTimeIntervalType
CDIFMETAIDENTIFIER	763021
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This defines a type representing a time interval measuring days and time within days.
USAGE	
ALIASES	
CONSTRAINTS	
TYPE	Kernel
ISABSTRACT	FALSE
LOCAL SUBTYPES	
LOCAL METARELATIONSHIPS	
LOCAL METAATTRIBUTES	

8.2.15 DefinitionObject

META-ENTITY REFERENCE	
NAME	DefinitionObject
CDIFMETAIDENTIFIER	8002
SUBJECTAREANAME	Common
SUBJECTAREAVERSION	15476-2:2002
LOCAL SUBTYPES	DataType
LOCAL METARELATIONSHIPS	
LOCAL METAATTRIBUTES	

8.2.16 EnumerationType

META-ENTITY DEFINITION	
NAME	EnumerationType
CDIFMETAIDENTIFIER	763022
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This data type has a fixed set of distinct logical values.
USAGE	This type is used for such sets as **Days of the Week** or **Colors of the Rainbow**.
ALIASES	
CONSTRAINTS	
TYPE	Kernel
ISABSTRACT	FALSE
LOCAL SUBTYPES	
LOCAL METARELATIONSHIPS	
LOCAL METAATTRIBUTES	

8.2.17 ExactNumericType

META-ENTITY DEFINITION	
NAME	ExactNumericType
CDIFMETAIDENTIFIER	763023
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This subtype defines numeric types that are exact.
USAGE	
ALIASES	
CONSTRAINTS	
TYPE	Kernel
ISABSTRACT	TRUE
LOCAL SUBTYPES	FixedDecimalType IntegerType SerialType
LOCAL METARELATIONSHIPS	
LOCAL METAATTRIBUTES	SingedFlag

META-ATTRIBUTE DEFINITION	META-ATTRIBUTE OF IntegerType
NAME	SignedFlag
CDIFMETAIDENTIFIER	763024
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This indicates whether the *IntegerType* can hold negative values. If it is false, then only positive values can be held by the data type.
USAGE	
ALIASES	
CONSTRAINTS	
DATA TYPE	Boolean
DOMAIN	
LENGTH	
ISOPTIONAL	TRUE

8.2.18 FixedDecimalType

META-ENTITY DEFINITION	
NAME	FixedDecimalType
CDIFMETAIDENTIFIER	763025
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This describes a data type with a fixed position decimal, with the number of digits of precision on both sides of the decimal point specified.
USAGE	
ALIASES	
CONSTRAINTS	
TYPE	Kernel
ISABSTRACT	FALSE
LOCAL SUBTYPES	BinaryCodedDecimalType PackedDecimalType
LOCAL METARELATIONSHIPS	
LOCAL METAATTRIBUTES	Precision Scale

META-ATTRIBUTE DEFINITION	META-ATTRIBUTE OF FixedDecimalType
NAME	Precision
CDIFMETAIDENTIFIER	763026
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This defines the maximum number of digits that can be supplied for the type, indicating the precision to which it can be specified.
USAGE	
ALIASES	
CONSTRAINTS	
DATA TYPE	Integer
DOMAIN	Positive
LENGTH	
ISOPTIONAL	TRUE

META-ATTRIBUTE DEFINITION	META-ATTRIBUTE OF FixedDecimalType
NAME	Scale
CDIFMETAIDENTIFIER	763027
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This defines the implicit position of the decimal point. This may be negative to indicate that the decimal point is implicitly to the right of the last digit, for example where the values stored are all thousands, the value of *Scale* would be -3; therefore a value of 7 would be interpreted as 7000.
USAGE	
ALIASES	
CONSTRAINTS	
DATA TYPE	Integer
DOMAIN	
LENGTH	
ISOPTIONAL	TRUE

8.2.19 FixedLengthBinaryType

META-ENTITY DEFINITION	
NAME	FixedLengthBinaryType
CDIFMETAIDENTIFIER	763028
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This describes a data type of fixed length binary, of the length specified.
USAGE	
ALIASES	
CONSTRAINTS	
TYPE	Kernel
ISABSTRACT	FALSE
LOCAL SUBTYPES	
LOCAL METARELATIONSHIPS	
LOCAL METAATTRIBUTES	Length LengthMultiplier

META-ATTRIBUTE DEFINITION	META-ATTRIBUTE OF FixedLengthBinaryType
NAME	Length
CDIFMETAIDENTIFIER	763029
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This defines the length of the *FixedLengthBinaryType*.
USAGE	
ALIASES	
CONSTRAINTS	If a value is specified for *Length*, the multiplier used shall be contained in *LengthMultiplier*.
DATA TYPE	Integer
DOMAIN	Positive Integer
LENGTH	
ISOPTIONAL	TRUE

META-ATTRIBUTE DEFINITION	META-ATTRIBUTE OF FixedLengthBinaryType
NAME	LengthMultiplier
CDIFMETAIDENTIFIER	763030
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This defines the multiplier units used for the length defined in *Length*.
USAGE	
ALIASES	
CONSTRAINTS	A value shall only be supplied for *LengthMultipiler* when a value is supplied for *Length*.
DATA TYPE	Enumerated
DOMAIN	Bit Byte (8-bit) Kilobyte (1,024 bytes) Megabyte (1,048,576 bytes) Gigabyte (1,073,741,824 bytes)
LENGTH	
ISOPTIONAL	TRUE

8.2.20 FixedLengthStringType

META-ENTITY DEFINITION	
NAME	FixedLengthStringType
CDIFMETAIDENTIFIER	763031
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This describes a string data type with a fixed storage capacity specified by the meta-attribute *Length*. This differs from *VariableLengthStringType* in that the string must contain exactly the number of characters required to fill the allocated storage space.
USAGE	In a DBMS, this may map to the type 'char'; in C, it could map to a character array; in Pascal, it could map to an array of 'char'.
ALIASES	
CONSTRAINTS	
TYPE	Kernel
ISABSTRACT	FALSE
LOCAL SUBTYPES	NLFixedLengthStringType
LOCAL METARELATIONSHIPS	
LOCAL METAATTRIBUTES	Length LengthMultiplier

META-ATTRIBUTE DEFINITION	META-ATTRIBUTE OF FixedLengthStringType
NAME	Length
CDIFMETAIDENTIFIER	763032
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This defines the length in characters of the *FixedLengthStringType*, in units defined by *LengthMultiplier*.
USAGE	
ALIASES	StringSize, StringCapacity
CONSTRAINTS	If specified, the multiplier used shall be contained in *LengthMultiplier*.
DATA TYPE	Integer
DOMAIN	Positive Integer
LENGTH	
ISOPTIONAL	TRUE

META-ATTRIBUTE DEFINITION	META-ATTRIBUTE OF FixedLengthStringType
NAME	LengthMultiplier
CDIFMETAIDENTIFIER	763033
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This defines the multiplier used for the length defined in *Length*.
USAGE	
ALIASES	
CONSTRAINTS	A value shall only be supplied for *LengthMultiplier* when a value is supplied for *Length*.
DATA TYPE	Enumerated
DOMAIN	Char KiloChar (1,024) MegaChar (1,048,576) GigaChar (1,073,741,824)
LENGTH	
ISOPTIONAL	TRUE

8.2.21 IntegerType

META-ENTITY DEFINITION	
NAME	IntegerType
CDIFMETAIDENTIFIER	763034
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This describes an integral number data type that may be signed or unsigned. It may also have a maximum and minimum value specified through use of a domain.
USAGE	An instance of this data type equates to char, short, int or long in the C language. The domain can be used to determine which C type is intended to be represented.
ALIASES	Integer, Long, Short, Char.
CONSTRAINTS	
TYPE	Kernel
ISABSTRACT	FALSE
LOCAL SUBTYPES	
LOCAL METARELATIONSHIPS	
LOCAL METAATTRIBUTES	

8.2.22 MagnitudeType

META-ENTITY DEFINITION	
NAME	MagnitudeType
CDIFMETAIDENTIFIER	763035
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This describes a type that is used to define quantitative information, such as size, extent or significance.
USAGE	
ALIASES	
CONSTRAINTS	
TYPE	Kernel
ISABSTRACT	TRUE
LOCAL SUBTYPES	MoneyType NumericType TemporalType
LOCAL METARELATIONSHIPS	
LOCAL METAATTRIBUTES	

8.2.23 MoneyType

META-ENTITY DEFINITION	
NAME	MoneyType
CDIFMETAIDENTIFIER	763036
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This data type is used to store monetary values. The currency can be specified.
USAGE	
ALIASES	
CONSTRAINTS	
TYPE	Kernel
ISABSTRACT	FALSE
LOCAL SUBTYPES	
LOCAL METARELATIONSHIPS	
LOCAL METAATTRIBUTES	Currency Precision Scale

META-ATTRIBUTE DEFINITION	META-ATTRIBUTE OF MoneyType
NAME	Currency
CDIFMETAIDENTIFIER	763037
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This defines the currency for the *MoneyType*.
USAGE	The value could be “USD” or “GBP”
ALIASES	
CONSTRAINTS	The allowable values are the 3-character codes defined in ISO 4217.
DATA TYPE	String
DOMAIN	as defined by the constraint.
LENGTH	3
ISOPTIONAL	TRUE

META-ATTRIBUTE DEFINITION	META-ATTRIBUTE OF MoneyType
NAME	Precision
CDIFMETAIDENTIFIER	763038
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This defines the maximum number of digits that can be supplied for the type, indicating the precision to which it can be specified.
USAGE	
ALIASES	
CONSTRAINTS	
DATA TYPE	Integer
DOMAIN	
LENGTH	
ISOPTIONAL	TRUE

META-ATTRIBUTE DEFINITION	META-ATTRIBUTE OF MoneyType
NAME	Scale
CDIFMETAIDENTIFIER	763039
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This defines the implicit position of the decimal point. This may be negative to indicate that the decimal point is implicitly to the right of the last digit, for example where the values stored are all thousands, the value of *Scale* would be -3; therefore a value of 7 would be interpreted as 7000.
USAGE	
ALIASES	
CONSTRAINTS	
DATA TYPE	Integer
DOMAIN	
LENGTH	
ISOPTIONAL	TRUE

8.2.24 NLFixedLengthStringType

META-ENTITY DEFINITION	
NAME	NLFixedLengthStringType
CDIFMETAIDENTIFIER	763040
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This describes a string data type, capable of storing National Language character sets, with a fixed storage capacity specified by the inherited meta-attribute *Length*. This differs from *NLVariableLengthStringType* in that the string must contain exactly the number of characters required to fill the allocated storage space.
USAGE	This is used for strings using character sets where each character in the national character set used can take more than one 8-bit byte to represent it. The number of bits required for each character may be held in the meta-attribute *BitsPerCharacter*.
ALIASES	
CONSTRAINTS	
TYPE	Kernel
ISABSTRACT	FALSE
LOCAL SUBTYPES	
LOCAL METARELATIONSHIPS	
LOCAL METAATTRIBUTES	BitsPerCharacter

META-ATTRIBUTE DEFINITION	META-ATTRIBUTE OF NLFixedLengthStringType
NAME	BitsPerCharacter
CDIFMETAIDENTIFIER	763041
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This defines the number of bits required for each character of the *NLFixedLengthStringType*.
USAGE	
ALIASES	
CONSTRAINTS	
DATA TYPE	Integer
DOMAIN	Positive Integer
LENGTH	
ISOPTIONAL	TRUE

8.2.25 NLVariableLengthStringType

META-ENTITY DEFINITION	
NAME	NLVariableLengthStringType
CDIFMETAIDENTIFIER	763042
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This describes a string data type, capable of storing National Language character sets, with a variable capacity up to a maximum specified by the inherited meta-attribute *MaxLength*.
USAGE	This is used for strings using multi-byte character sets, where each character in the national character set used can take more than one 8-bit byte to represent it. The number of bits required for each character may be held in the meta-attribute *BitsPerCharacter*.
ALIASES	
CONSTRAINTS	
TYPE	Kernel
ISABSTRACT	FALSE
LOCAL SUBTYPES	
LOCAL METARELATIONSHIPS	
LOCAL METAATTRIBUTES	BitsPerCharacter

META-ATTRIBUTE DEFINITION	META-ATTRIBUTE OF NLVariableLengthStringType
NAME	BitsPerCharacter
CDIFMETAIDENTIFIER	763043
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This defines the number of bits required for each character of the *NLVariableLengthStringType*.
USAGE	
ALIASES	
CONSTRAINTS	
DATA TYPE	Integer
DOMAIN	Positive Integer
LENGTH	
ISOPTIONAL	TRUE

8.2.26 NumericType

META-ENTITY DEFINITION	
NAME	NumericType
CDIFMETAIDENTIFIER	763044
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This describes a type that is used to hold a number, the units of which may be specified though association to an instance of the meta-entity *Unit*.
USAGE	
ALIASES	
CONSTRAINTS	
TYPE	Kernel
ISABSTRACT	TRUE
LOCAL SUBTYPES	ApproximateNumericType ComplexType ExactNumericType
LOCAL METARELATIONSHIPS	NumericType.IsMeasuredIn.Unit
LOCAL METAATTRIBUTES	

8.2.27 PackedDecimalType

META-ENTITY DEFINITION	
NAME	PackedDecimalType
CDIFMETAIDENTIFIER	763045
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This defines the data type for packed decimal data.
USAGE	
ALIASES	
CONSTRAINTS	
TYPE	Kernel
ISABSTRACT	FALSE
LOCAL SUBTYPES	
LOCAL METARELATIONSHIPS	
LOCAL METAATTRIBUTES	

8.2.28 PointerQualifier

META-ENTITY DEFINITION	
NAME	PointerQualifier
CDIFMETAIDENTIFIER	763046
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This provides a mechanism to describe the occurrence of a pointer.
USAGE	
ALIASES	
CONSTRAINTS	
TYPE	Characteristic
ISABSTRACT	FALSE
LOCAL SUBTYPES	
LOCAL METARELATIONSHIPS	
LOCAL METAATTRIBUTES	

8.2.29 PolarComplexType

META-ENTITY DEFINITION	
NAME	PolarComplexType
CDIFMETAIDENTIFIER	763047
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This subtype defines a type for expressing complex numbers in polar form.
USAGE	
ALIASES	
CONSTRAINTS	
TYPE	Kernel
ISABSTRACT	FALSE
LOCAL SUBTYPES	
LOCAL METARELATIONSHIPS	
LOCAL METAATTRIBUTES	DistancePrecision DistanceScale GradientPrecision GradientScale

META-ATTRIBUTE DEFINITION	META-ATTRIBUTE OF PolarComplexType
NAME	DistancePrecision
CDIFMETAIDENTIFIER	763048
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This defines the maximum number of digits that can be supplied for the distance component of the type, indicating the precision to which it can be specified.
USAGE	
ALIASES	
CONSTRAINTS	
DATA TYPE	Integer
DOMAIN	
LENGTH	
ISOPTIONAL	TRUE

META-ATTRIBUTE DEFINITION	META-ATTRIBUTE OF PolarComplexType
NAME	DistanceScale
CDIFMETAIDENTIFIER	763049
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This defines the implicit position of the decimal point in the distance component of the type. This may be negative to indicate that the decimal point is implicitly to the right of the last digit, for example where the values stored are all thousands, the value of *Scale* would be -3; therefore a value of 7 would be interpreted as 7000.
USAGE	
ALIASES	
CONSTRAINTS	
DATA TYPE	Integer
DOMAIN	
LENGTH	
ISOPTIONAL	TRUE

META-ATTRIBUTE DEFINITION	META-ATTRIBUTE OF PolarComplexType
NAME	GradientPrecision
CDIFMETAIDENTIFIER	763050
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This defines the maximum number of digits that can be supplied for the gradient component of the type, indicating the precision to which it can be specified.
USAGE	
ALIASES	
CONSTRAINTS	
DATA TYPE	Integer
DOMAIN	
LENGTH	
ISOPTIONAL	TRUE

META-ATTRIBUTE DEFINITION	META-ATTRIBUTE OF PolarComplexType
NAME	GradientScale
CDIFMETAIDENTIFIER	763051
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This defines the implicit position of the decimal point in the gradient component of the type. This may be negative to indicate that the decimal point is implicitly to the right of the last digit, for example where the values stored are all thousands, the value of *Scale* would be -3; therefore a value of 7 would be interpreted as 7000.
USAGE	
ALIASES	
CONSTRAINTS	
DATA TYPE	Integer
DOMAIN	
LENGTH	
ISOPTIONAL	TRUE

8.2.30 QualifiedDataType

META-ENTITY DEFINITION	
NAME	QualifiedDataType
CDIFMETAIDENTIFIER	763052
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This is a pointer or array data type.
USAGE	
ALIASES	
CONSTRAINTS	
TYPE	Associative
ISABSTRACT	TRUE
LOCAL SUBTYPES	
LOCAL METARELATIONSHIPS	QualifiedDataType.IsQualificationOf.DataType
LOCAL METAATTRIBUTES	

8.2.31 Qualifier

META-ENTITY DEFINITION	
NAME	Qualifier
CDIFMETAIDENTIFIER	763053
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This provides a mechanism to describe common characteristics between *PointerQualifier* and *ArrayQualifier*.
USAGE	This meta-entity shall not be instantiated.
ALIASES	
CONSTRAINTS	
TYPE	Characteristic
ISABSTRACT	FALSE
LOCAL SUBTYPES	ArrayQualifier PointerQualifier
LOCAL METARELATIONSHIPS	
LOCAL METAATTRIBUTES	PrecedenceNumber

META-ATTRIBUTE DEFINITION	META-ATTRIBUTE OF Qualifier
NAME	PrecedenceNumber
CDIFMETAIDENTIFIER	763054
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This describes the precedence of the *Qualifier* within the set of qualifiers associated with the data type to which it is related. Since many languages and methods allow complex array and pointer groupings, this allows precedence to be specified in a syntax-free fashion.
USAGE	An array of pointers to integers would cause an instance of *ArrayQualifier* to have a precedence of 1 and an instance of *PointerQualifier* to have a precedence of 2.
ALIASES	
CONSTRAINTS	
DATA TYPE	Integer
DOMAIN	Positive Integer
LENGTH	
ISOPTIONAL	FALSE

8.2.32 RefinedDataType

META-ENTITY DEFINITION	
NAME	RefiledDataType
CDIFMETAIDENTIFIER	763055
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This is a data type that is a subtype of another data type or in some way a refinement of the another data type.
USAGE	
ALIASES	
CONSTRAINTS	
TYPE	Kernel
ISABSTRACT	FALSE
LOCAL SUBTYPES	
LOCAL METARELATIONSHIPS	
LOCAL METAATTRIBUTES	

8.2.33 SerialType

META-ENTITY DEFINITION	
NAME	SerialType
CDIFMETAIDENTIFIER	763056
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This defines a data type that is initialized to a given value and is then incremented (or optionally decremented) for every instance of the parent container created. The minimum and maximum values are defined using a domain.
USAGE	The data type Serial in SQL is an example of this data type.
ALIASES	
CONSTRAINTS	
TYPE	Kernel
ISABSTRACT	FALSE
LOCAL SUBTYPES	
LOCAL METARELATIONSHIPS	
LOCAL METAATTRIBUTES	Cycle Interval StartingValue

META-ATTRIBUTE DEFINITION	META-ATTRIBUTE OF SerialType
NAME	Cycle
CDIFMETAIDENTIFIER	763057
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	For ascending sequences, this specifies that the sequence will return to the minimum value after reaching the maximum value. For descending sequences, the sequence will return to the maximum value after reaching the minimum value. If set to ***FALSE***, this indicates that the value does not cycle.
USAGE	This is used where a bounding of the range of the sequence is desired, such as a product batch where the maximum number of objects in the process is known and each object is identified within the context of the batch. Each new batch would start the sequence anew.
ALIASES	
CONSTRAINTS	
DATA TYPE	Boolean
DOMAIN	
LENGTH	
ISOPTIONAL	TRUE

META-ATTRIBUTE DEFINITION	META-ATTRIBUTE OF SerialType
NAME	Interval
CDIFMETAIDENTIFIER	763058
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This defines the interval between the values of the type. If the value is negative, then the value is decremented.
USAGE	
ALIASES	
CONSTRAINTS	
DATA TYPE	Integer
DOMAIN	
LENGTH	
ISOPTIONAL	TRUE

META-ATTRIBUTE DEFINITION	META-ATTRIBUTE OF SerialType
NAME	StartingValue
CDIFMETAIDENTIFIER	763059
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This defines the initial number to be used for the serial data type.
USAGE	
ALIASES	
CONSTRAINTS	The value specified shall fall within the range defined by the domain for the type, if this is present.
DATA TYPE	Integer
DOMAIN	
LENGTH	
ISOPTIONAL	TRUE

8.2.34 StringType

META-ENTITY DEFINITION	
NAME	StringType
CDIFMETAIDENTIFIER	763060
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This describes a string data type.
USAGE	This type should not be instantiated; one of the subtypes should be used where possible.
ALIASES	
CONSTRAINTS	
TYPE	Kernel
ISABSTRACT	FALSE
LOCAL SUBTYPES	FixedLengthStringType VariableLengthStringType
LOCAL METARELATIONSHIPS	
LOCAL METAATTRIBUTES	CharacterSet

META-ATTRIBUTE DEFINITION	META-ATTRIBUTE OF StringType
NAME	CharacterSet
CDIFMETAIDENTIFIER	763061
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This describes the encoding used to represent the characters of the string.
USAGE	
ALIASES	
CONSTRAINTS	This meta-attribute is used to define what character encoding is used to represent the string. The values are described by reference to the relevant standard. The strings of character encoding name are found in the IANA site (http://www.iana.org/assignments/character-sets).
DATA TYPE	String
DOMAIN	
LENGTH	256
ISOPTIONAL	TRUE

8.2.35 TemporalType

META-ENTITY DEFINITION	
NAME	TemporalType
CDIFMETAIDENTIFIER	763062
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This describes a type that is used to define a temporal value.
USAGE	
ALIASES	
CONSTRAINTS	
TYPE	Kernel
ISABSTRACT	TRUE
LOCAL SUBTYPES	DateType TimeIntervalType TimeStampType TimeType
LOCAL METARELATIONSHIPS	
LOCAL METAATTRIBUTES	

8.2.36 TimeIntervalType

META-ENTITY DEFINITION	
NAME	TimeIntervalType
CDIFMETAIDENTIFIER	763063
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This describes a time interval.
USAGE	
ALIASES	
CONSTRAINTS	
TYPE	Kernel
ISABSTRACT	TRUE
LOCAL SUBTYPES	DayTimeIntervalType YearMonthIntervalType
LOCAL METARELATIONSHIPS	
LOCAL METAATTRIBUTES	

8.2.37 TimeStampType

META-ENTITY DEFINITION	
NAME	TimeStampType
CDIFMETAIDENTIFIER	763064
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This defines the timestamp data type.
USAGE	This is used to give a timestamp to an object for some operation, such as creation or updating.
ALIASES	
CONSTRAINTS	
TYPE	Kernel
ISABSTRACT	FALSE
LOCAL SUBTYPES	
LOCAL METARELATIONSHIPS	
LOCAL METAATTRIBUTES	IsLocal TimeZoneHours TimeZoneMinutes

META-ATTRIBUTE DEFINITION	META-ATTRIBUTE OF TimeStampType
NAME	IsLocal
CDIFMETAIDENTIFIER	763065
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This specifies that the time is expressed in local time, with no time zone specified.
USAGE	
ALIASES	
CONSTRAINTS	If this meta-attribute is set to **TRUE**, neither *TimeZoneHours* nor *TimeZoneMinutes* shall have values supplied.
DATA TYPE	Boolean
DOMAIN	
LENGTH	
ISOPTIONAL	TRUE

META-ATTRIBUTE DEFINITION	META-ATTRIBUTE OF TimeStampType
NAME	TimeZoneHours
CDIFMETAIDENTIFIER	763066
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This specifies the offset in hours from UTC, as defined in ISO 9945-1. If negative, the time zone shall be to the east of the Prime Meridian; otherwise it shall be to the west.
USAGE	For example, Toronto is UTC minus five hours.
ALIASES	
CONSTRAINTS	No value shall be supplied if *IsLocal* is set to **TRUE**. A value shall be supplied if *TimeZoneMinutes* is supplied.
DATA TYPE	Integer
DOMAIN	-23..+23
LENGTH	
ISOPTIONAL	TRUE

NOTE: UTC stands for Universal Coordinated Time (English translation of French term) as defined in ISO 9945-1.

META-ATTRIBUTE DEFINITION	META-ATTRIBUTE OF TimeStampType
NAME	TimeZoneMinutes
CDIFMETAIDENTIFIER	763067
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This specifies the offset in minutes from UTC.
USAGE	For example, Pakistan is UTC plus 5 hours and 45 minutes.
ALIASES	
CONSTRAINTS	No value shall be supplied if *IsLocal* is set to **TRUE**. A value shall be supplied if *TimeZoneHours* is supplied.
DATA TYPE	Integer
DOMAIN	0..59
LENGTH	
ISOPTIONAL	TRUE

8.2.38 TimeType

META-ENTITY DEFINITION	
NAME	TimeType
CDIFMETAIDENTIFIER	763068
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This describes the time data type.
USAGE	
ALIASES	
CONSTRAINTS	
TYPE	Kernel
ISABSTRACT	FALSE
LOCAL SUBTYPES	
LOCAL METARELATIONSHIPS	
LOCAL METAATTRIBUTES	IsLocal TimeZoneHours TimeZoneMinutes

META-ATTRIBUTE DEFINITION	META-ATTRIBUTE OF TimeType
NAME	IsLocal
CDIFMETAIDENTIFIER	763069
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This specifies that the time is expressed in local time, with no time zone specified.
USAGE	
ALIASES	
CONSTRAINTS	If this meta-attribute is set to **TRUE**, neither *TimeZoneHours* nor *TimeZoneMinutes* shall have values supplied.
DATA TYPE	Boolean
DOMAIN	
LENGTH	
ISOPTIONAL	TRUE

META-ATTRIBUTE DEFINITION	META-ATTRIBUTE OF TimeType
NAME	TimeZoneHours
CDIFMETAIDENTIFIER	763070
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This specifies the offset in hours from UTC, as defined in ISO 9945-1. If negative, the time zone shall be to the east of the Prime Meridian; otherwise it shall be to the west.
USAGE	For example, Toronto is UTC plus five hours.
ALIASES	
CONSTRAINTS	No value shall be supplied if *IsLocal* is set to **TRUE**. A value shall be supplied if *TimeZoneMinutes* is supplied.
DATA TYPE	Integer
DOMAIN	-23..+23
LENGTH	
ISOPTIONAL	TRUE

META-ATTRIBUTE DEFINITION	META-ATTRIBUTE OF TimeType
NAME	TimeZoneMinutes
CDIFMETAIDENTIFIER	763071
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This specifies the offset in minutes from UTC.
USAGE	For example, Pakistan is UTC minus 5 hours and 45 minutes.
ALIASES	
CONSTRAINTS	No value shall be supplied if *IsLocal* is set to **TRUE**. A value shall be supplied if *TimeZoneHours* is supplied.
DATA TYPE	Integer
DOMAIN	0..59
LENGTH	
ISOPTIONAL	TRUE

8.2.39 UnboundedArrayQualifier

META-ENTITY DEFINITION	
NAME	UnboundedArrayQualifier
CDIFMETAIDENTIFIER	763072
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This subtype of *ArrayQualifier* is used where a dimension of an array is unbounded.
USAGE	
ALIASES	
CONSTRAINTS	
TYPE	Characteristic
ISABSTRACT	FALSE
LOCAL SUBTYPES	
LOCAL METARELATIONSHIPS	
LOCAL METAATTRIBUTES	

8.2.40 Unit

META-ENTITY DEFINITION	
NAME	Unit
CDIFMETAIDENTIFIER	763073
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This describes the unit that applies to any related numeric type. It is defined in terms of coefficients of the SI basic units, and a flag to indicate if actual SI units are used. If SI units are not used, then the exponents indicate that the unit represents the **concept** that the SI unit represents; for example a temperature in Fahrenheit would have a coefficient for *ExponentForKelvin* of ***1***. The name of the unit can also be conveyed.
USAGE	For example, a pressure measured in kilograms per square meter would have an exponent for Kilogram of ***1*** and one of ***-2*** for Meter.
ALIASES	
CONSTRAINTS	
TYPE	Kernel
ISABSTRACT	FALSE
LOCAL SUBTYPES	
LOCAL METARELATIONSHIPS	
LOCAL METAATTRIBUTES	ExponentForAmpere ExponentForCandela ExponentForKelvin ExponentForKilogram ExponentForMeter ExponentForMole ExponentForSecond IsSI Name

META-ATTRIBUTE DEFINITION	META-ATTRIBUTE OF Unit
NAME	ExponentForAmpere
CDIFMETAIDENTIFIER	763074
SUBJECTAREANAME	Data definitions
SUBJECTAREAVERSION	15476-3:2003
DESCRIPTION	This contains the exponent for Amperes, which is a measure of current, if used to form the unit of measurement.
USAGE	
ALIASES	
CONSTRAINTS	
DATA TYPE	Integer
DOMAIN	
LENGTH	
ISOPTIONAL	TRUE