INTERNATIONAL STANDARD

ISO 12006-3

First edition
2007-04-15

Building construction — Organization of information about construction works —

Part 3: Framework for object-oriented information

Construction immobilière — Organisation de l'information des travaux de construction —

Partie 3: Schéma pour l'information basée sur l'objet

Reference number
ISO 12006-3:2007(E)

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

COPYRIGHT PROTECTED DOCUMENT

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

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

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

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

Attention is drawn to the possibility that some of the elements of this International Standard ISO 12006-3 may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO 12006-3 was prepared by Technical Committee ISO/TC 59, *Building construction*, Subcommittee SC 13, *Organization of information about construction works*.

This first edition of ISO 12006-3 cancels and replaces ISO/PAS 12006-3:2001.

ISO 12006 consists of the following parts, under the general title *Building construction — Organization of information about construction works*:

— *Part 2: Framework for classification of information*

— *Part 3: Framework for object-oriented information*

Introduction

The main part of ISO 12006-3 consists of the specification of a taxonomy model, which provides the ability to define concepts by means of properties, to group concepts, and to define relationships between concepts. Objects, collections and relationships are the basic entities of the model. The set of properties associated with an object provide the formal definition of the object as well as its typical behaviour. Properties have values, optionally expressed in units.

The role that an object is intended to play can be designated through the model and this provides the capability to define the context within which the object is used. Each object may have multiple names and this allows for its expression in terms of synonyms or in multiple languages. The language name of each object must always be given in English (the default language). An object may also be named in terms of the language of the location in which it is determined or used. Objects may be related to formal classification systems through the provision of references.

The model has one root entity from which the following three subtype entities inherit: objects, collections and the relationships between them. The root entity provides the ability to assign any set of names, labels, descriptions and references, in any language, to its derived types, as well as identifiers and dates.

Objects are divided into subjects, activities, actors, units, values and measures with units and properties. Subjects and activities are the things and processes that are described. The others are description entities related to other objects and themselves through relationships.

Relationships provide an association mechanism between objects. Relationships are divided into association, collection, specialization, composition, involvement (acting upon), property assignment, sequencing and measure assignment.

Collections provide for all kinds of groupings of objects, including nested collections, by means of the collect relationship.

Properties are entities that provide the context for data stored as values. Properties are differentiated according to types of data containment: enumeration values, list values, bounded list values, bounded values, single values and table values.

The value content, associated with a property through a measure with a unit, will be stored in the value component, which is language-dependent. The latter entity models the way any name, description, value or reference is represented on a per language base.

The model described in this part of ISO 12006 is proposed as a bridge between classification systems as described in ISO 12006-2 [5], and product modelling as described in several publications [2], [3], [6], [7].

Building construction — Organization of information about construction works —

Part 3: Framework for object-oriented information

1 Scope

This part of ISO 12006 specifies a language-independent information model which can be used for the development of dictionaries used to store or provide information about construction works.

It enables classification systems, information models, object models and process models to be referenced from within a common framework.

2 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 10303-11, *Industrial automation systems and integration — Product data representation and exchange — Part 11: Description methods: The EXPRESS language reference manual*

ISO/IEC 10646, *Information technology — Universal Multiple-Octet Coded Character Set (UCS)*

3 Language encoding

All information that is specified as type “String”, or that resolves to type “String”, shall be able to be expressed using the UNICODE character set [8] as set out in ISO/IEC 10646, preferably using the UTF-8 encoding form, the UTF-8 encoding scheme and the “UCS Transformation Format 8” [4].

4 Specification

4.1 General

The model in this part of ISO 12006 is specified using the EXPRESS data definition language according to ISO 10303-11.

The model is described informally in 4.2, conforming to the EXPRESS-G notation.

The model is described formally in the EXPRESS language specification presented in 4.3 and as an EXPRESS long form specification in 4.4.

4.2 EXPRESS-G specification

The informal EXPRESS-G specification that uses the EXPRESS-G notation is given in six diagrams (Figures 1 through 6), where each diagram specifies a part of the model. All entities in these diagrams are specified formally in 4.3.

— Figure 1 shows the top level diagram with xtdRoot, its attributes and its derived types xtdObject, xtdRelationship and xtdCollection.

— Figure 2 shows xtdLanguageRepresentation with its derived types xtdName and xtdDescription that are attributed to xtdRoot and its subtypes.

— Figure 3 shows the relationship types derived from xtdRelationship that are used to establish possible relationships between xtdObjects, xtdCollections and xtdExternalDocuments or subtypes of these.

— Figure 4 shows the assignment of xtdProperties to xtdObjects through xtdRelAssignsProperties and xtdMeasureWithUnit with the assignment of values, through xtdRelAssignsMeasures.

— Figure 5 lists the basic types used in the model and their related EXPRESS types.

— Figure 6 shows details of xtdValue and xtdExternalDocument.

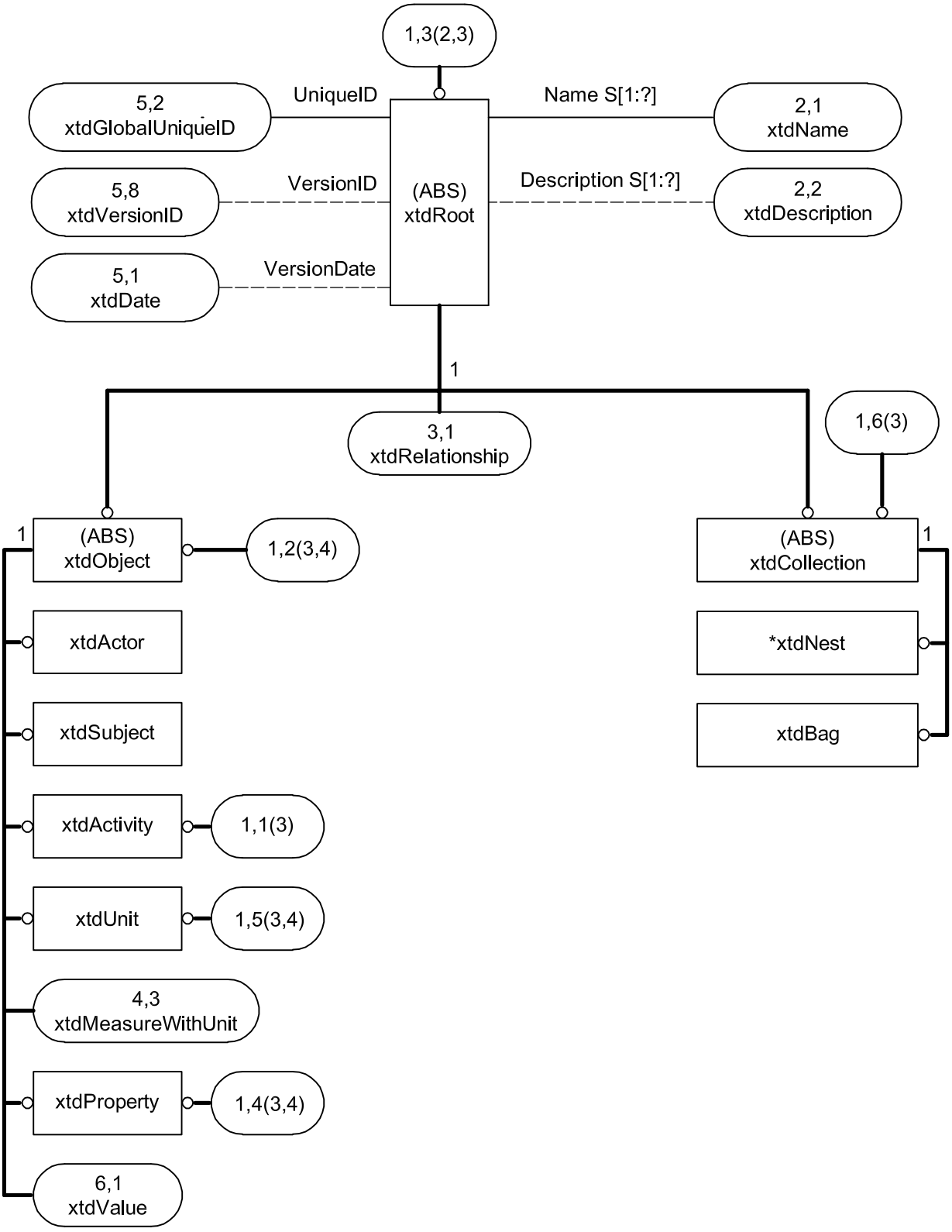

Figure 1 — EXPRESS-G diagram 1 — Top level with root concept

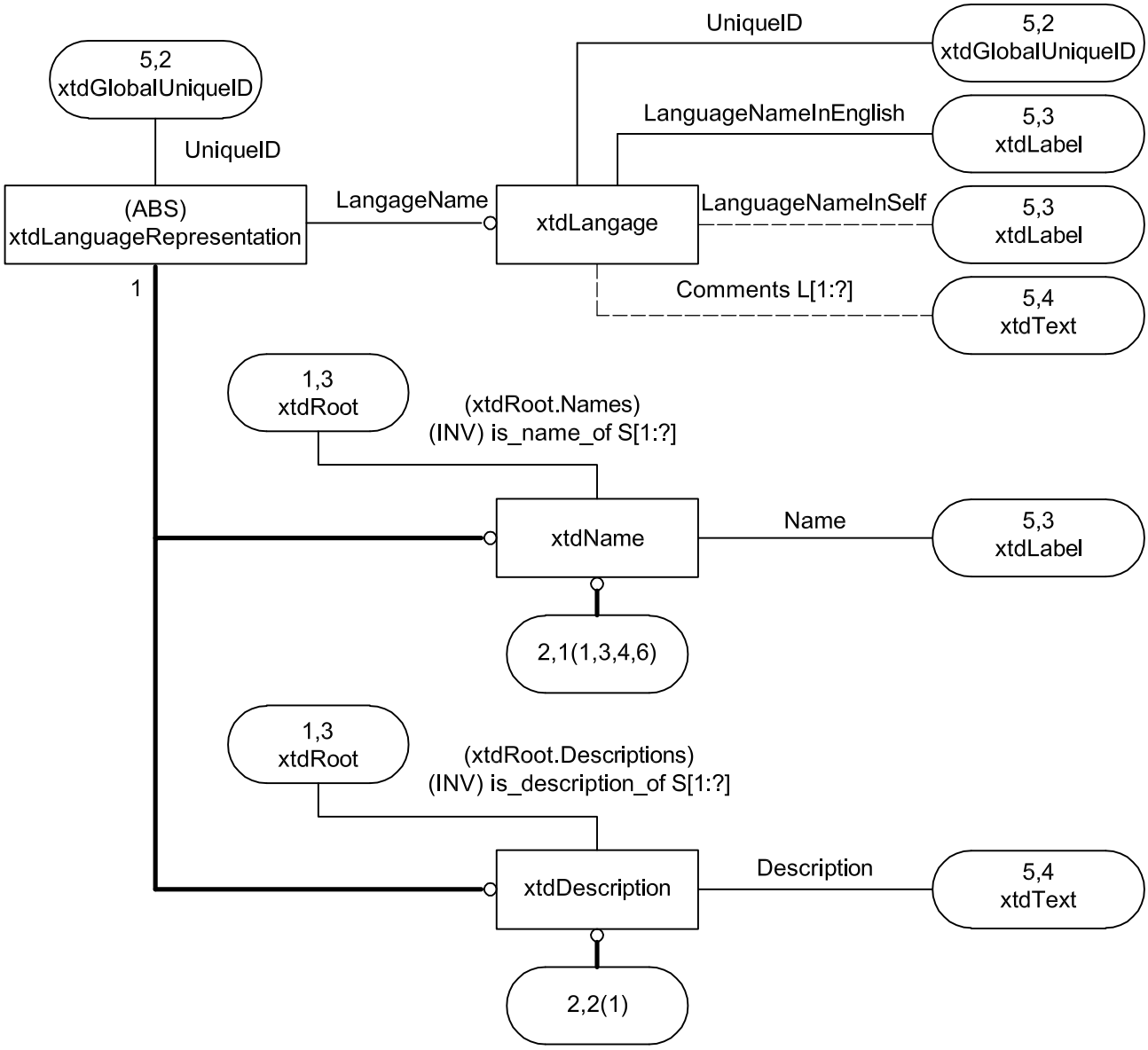

Figure 2 — EXPRESS-G diagram 2 — Language representation, names and descriptions

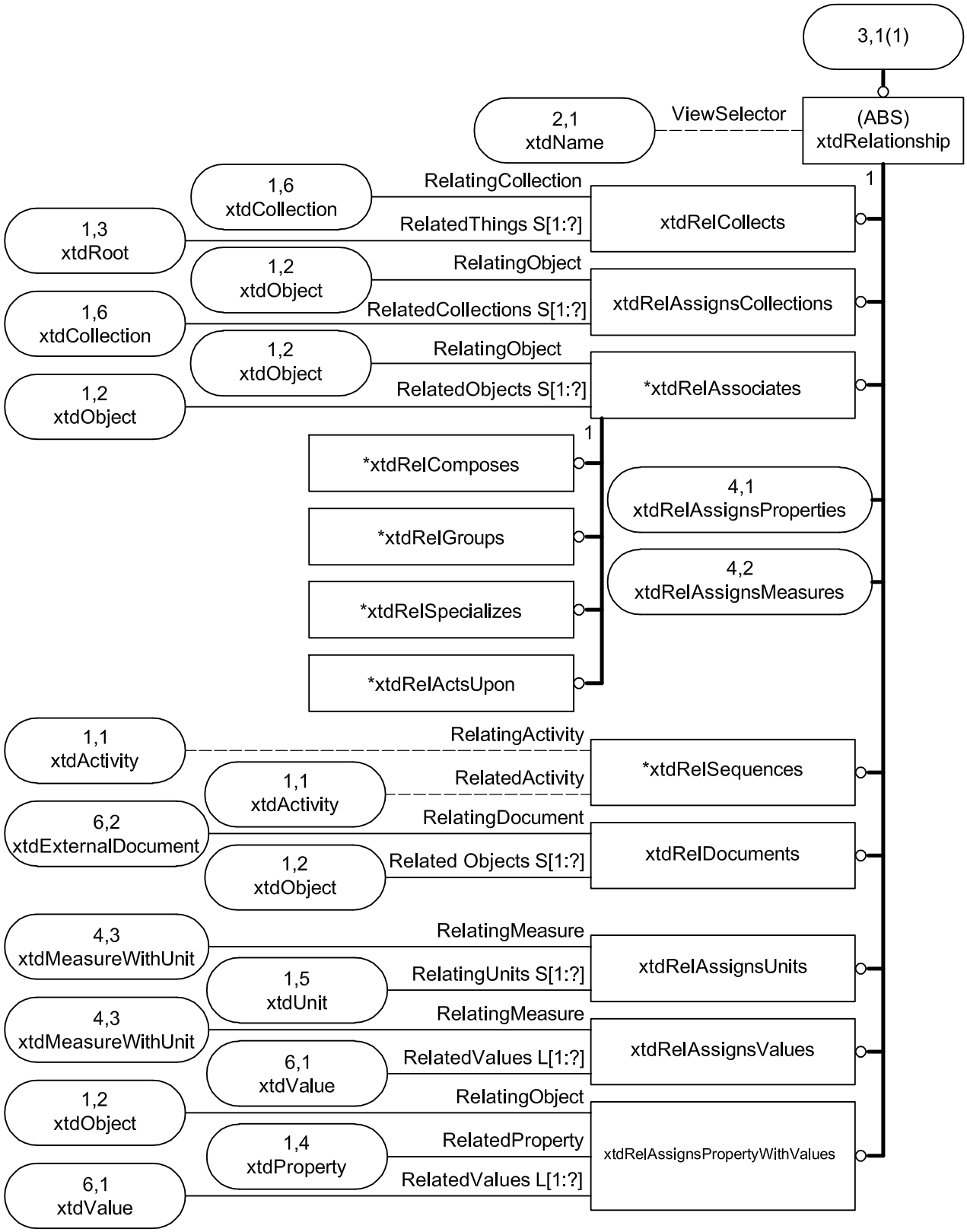

Figure 3 — EXPRESS-G diagram 3 — Relationships

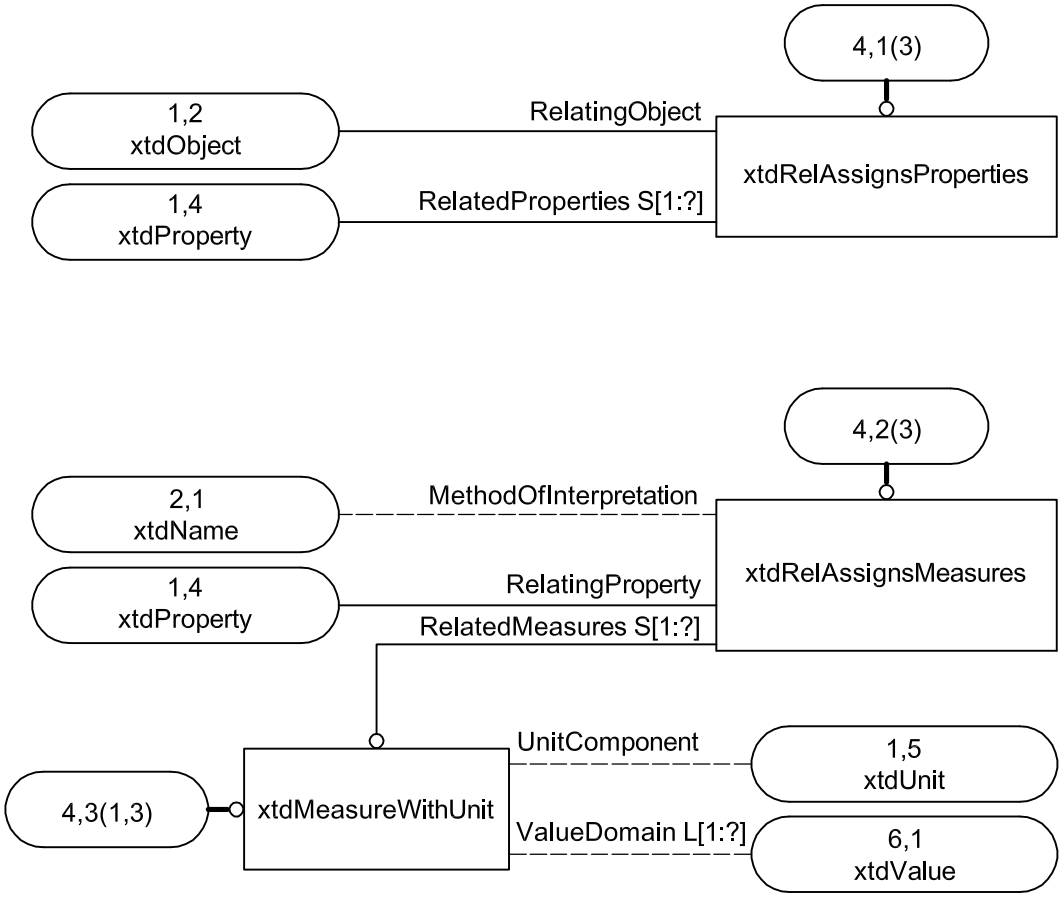

Figure 4 — EXPRESS-G diagram 4 — Assignment of properties and measures

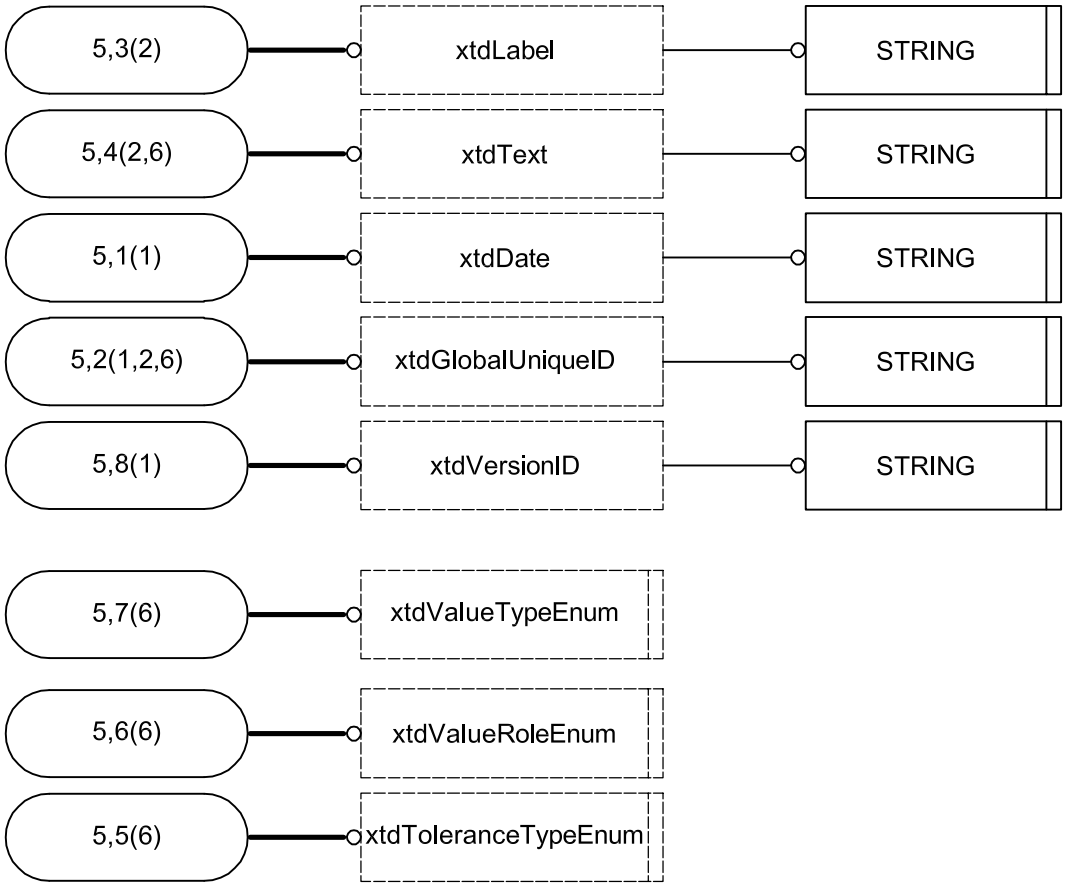

Figure 5 — EXPRESS-G diagram 5 — Basic types

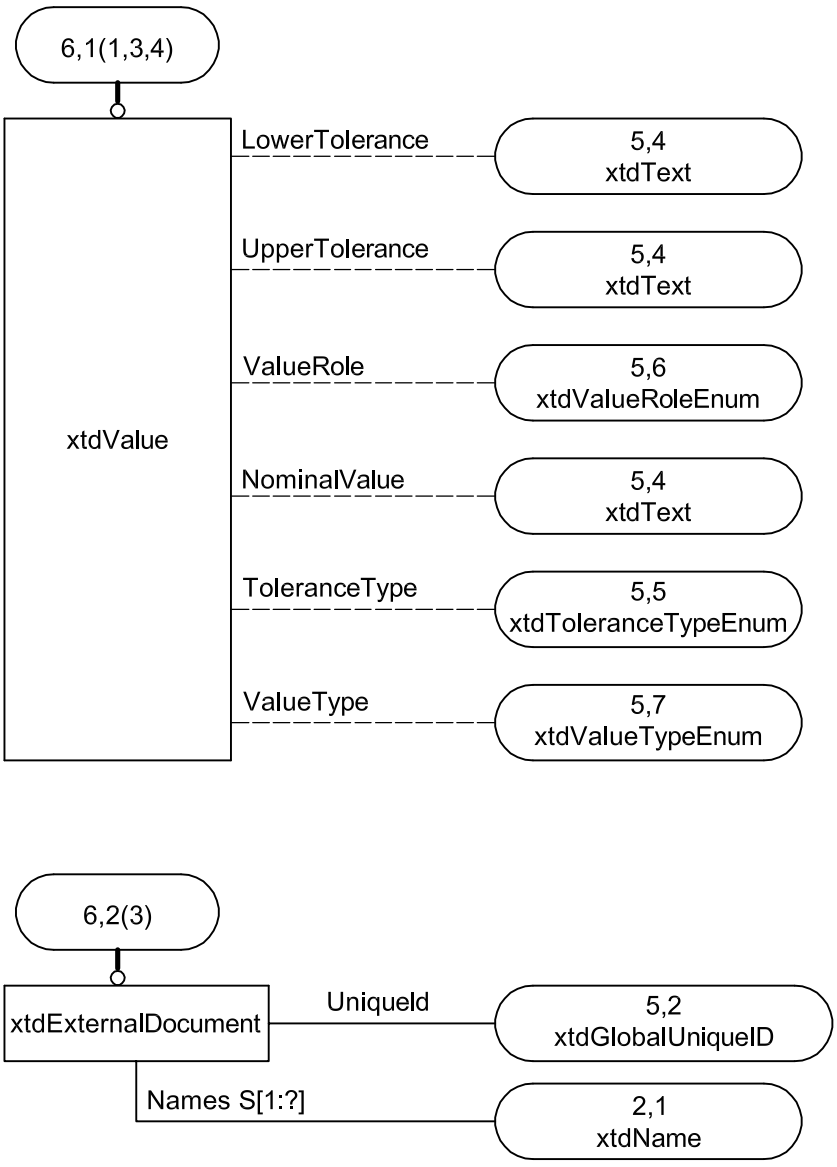

Figure 6 — EXPRESS-G diagram 6 — Values and external documents

4.3 EXPRESS specification

This formal specification is provided in the EXPRESS language.

Express Specification
(*
ISO_12006_3_VERSION_3

EXPRESS specification:

```
*)
SCHEMA ISO_12006_3_VERSION_3;
(*
```

4.3.1 xtdDate

xtdDate is a defined data type of simple data type STRING that is used to contain an alphanumeric representation of a date. It is recommended that the date format “YYYY.MM.DD” should be used.

EXAMPLE The 31st day of May in the year 2000 should be written as “2000.05.31”.

EXPRESS specification:

```
*)
  TYPE xtdDate = STRING;
  END_TYPE;
(*
```

4.3.2 xtdGlobalUniqueID

xtdGlobalUniqueID is a defined data type of simple data type STRING that holds an identifier that is globally unique.

Typically, the identifier is generated using an algorithm such as that published by the Object Management Group which is based on the IP address of the computer that generates the identifier.

There are implementations of the above algorithm that can be used to create an identifier.

An identifier is a unique 128-bit number. It is desirable to compress the identifier size to reduce overhead. Implementers shall compress the identifier down to a 22-character string using a compression algorithm with 64 characters for the base. The encoding of the base 64-character set is shown below:

```
            1         2         3         4         5         6
 0123456789012345678901234567890123456789012345678901234567890123
"0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz_$";
```

EXAMPLE The 22-character string “93f09e4A_899402a9$D013” is a globally unique identifier.

Every concept within a library can only have one globally unique identifier. Globally unique identifiers of similar concepts in different, separately developed libraries may differ. This part of ISO 12006 does not give a mechanism that automatically evaluates similar concepts with different globally unique identifiers across libraries. Evaluation of concepts between libraries to ascertain equivalence must be done separately.

NOTE The algorithm is explained at http://www.opengroup.org/dce/info/draft-leach-uuids-guids-01.txt.

EXPRESS specification:

```
*)
   TYPE xtdGlobalUniqueID = STRING;
   END_TYPE;
(*
```

4.3.3 xtdLabel

xtdLabel is a defined data type of simple data type STRING that is a set of alphanumeric UNICODE characters used to nominate an object.

EXPRESS specification:

```
*)
   TYPE xtdLabel = STRING;
   END_TYPE;
(*
```

4.3.4 xtdText

xtdText is a defined data type of simple data type STRING that is a set of alphanumeric UNICODE characters used to annotate an object.

EXPRESS specification:

```
*)
   TYPE xtdText = STRING;
   END_TYPE;
(*
```

4.3.5 xtdVersionID

xtdVersionID is a defined data type of simple data type STRING that is a set of alphanumeric UNICODE characters used to identify the version of an object.

EXAMPLE “1A”, “12”, “1.1” are types of versioning.

EXPRESS specification:

```
*)
   TYPE xtdVersionID = STRING;
   END_TYPE;
(*
```

4.3.6 xtdToleranceTypeEnum

xtdToleranceTypeEnum is an enumeration type giving the range of possible types of tolerances which can be applied to a value. Allowed selections are:

Realvalue Defined type of equal type as the value that expresses the tolerance as a real value.

Percentage Defined type that expresses the tolerance as a percentage of the value.

EXPRESS specification:

```
*)
   TYPE xtdToleranceTypeEnum = ENUMERATION OF
      (REALVALUE,
      PERCENTAGE);
   END_TYPE;
(*
```

4.3.7 xtdValueRoleEnum

xtdValueRoleEnum is an enumeration type giving the range of possible types of interpretation which can be applied to a value. Allowed selections are:

Nominal Defined type of denominating the value of type nominal.

Maximum Defined type of denominating the value as a maximum value.

Minimum Defined type of denominating the value as a minimum value.

EXPRESS specification:

```
*)
   TYPE xtdValueRoleEnum = ENUMERATION OF
      (NOMINAL,
      MAXIMUM,
      MINIMUM);
   END_TYPE;
(*
```

4.3.8 xtdValueTypeEnum

xtdValueTypeEnum is an enumeration type giving the range of possible simple value types from which a selection can be made. Allowed selections are:

xtdString Defined type of simple type STRING for descriptive purposes.

xtdNumber Defined type of simple type STRING.

xtdInteger Defined type of simple type INTEGER.

xtdReal Defined type of simple type REAL.

xtdBoolean Defined type of simple type BOOLEAN.

xtdLogical Defined type of simple type LOGICAL.

EXPRESS specification:

```
*)
   TYPE xtdValueTypeEnum = ENUMERATION OF
      (XTDSTRING,
      XTDNUMBER,
      XTDINTEGER,
      XTDREAL,
      XTDBOOLEAN,
      XTDLOGICAL);
   END_TYPE;
(*
```

4.3.9 xtdActivity

xtdActivity is a specialization of xtdObject that represents an activity or process that modifies an xtdSubject.

EXAMPLE "bricklaying" represents the activity or process of bringing together bricks and mortar to create a wall or other element of construction.

EXPRESS specification:

```
*)
  ENTITY xtdActivity
    SUBTYPE OF(xtdObject);
  END_ENTITY;
(*
```

4.3.10 xtdActor

xtdActor is a specialization of xtdObject that represents an agent that acts upon an xtdSubject within an xtdActivity.

An actor can be a person, profession, organizational unit, equipment.

EXAMPLE 1 “carpenter” is an xtdActor.

EXAMPLE 2 “ISO” is an xtdActor.

EXAMPLE 3 “crane” may be used as an xtdActor.

EXPRESS specification:

```
*)
  ENTITY xtdActor
    SUBTYPE OF(xtdObject);
  END_ENTITY;
(*
```

4.3.11 xtdBag

xtdBag is a specialization of xtdCollection that has no constraints on the things that can be members of the collection.

A collection of actors and activities and subjects may be collected together within an instance of xtdBag.

EXPRESS specification:

```
*)
  ENTITY xtdBag
    SUBTYPE OF(xtdCollection);
  END_ENTITY;
(*
```

4.3.12 xtdCollection

xtdCollection represents a group or collection of things. An xtdCollection can be one of xtdNest or xtdBag.

EXPRESS specification:

```
*)
  ENTITY xtdCollection
    ABSTRACT SUPERTYPE OF (ONEOF(xtdNest, xtdBag))
    SUBTYPE OF(xtdRoot);
  END_ENTITY;
(*
```

4.3.13 xtdDescription

xtdDescription holds the description or definition of a thing by means of text.

EXAMPLE "A door leaf is the part of a door that opens to allow people or goods to pass" is the definitive description in English for the object "door leaf".

EXPRESS specification:

```
*)
  ENTITY xtdDescription
    SUBTYPE OF(xtdLanguageRepresentation);
    Description                :xtdText;
  INVERSE
    is_description_of          :SET [1:?] OF xtdRoot FOR Descriptions;
  END_ENTITY;
(*
```

Attribute definition:

Description
an instance of xtdText that holds the description or definition

4.3.14 xtdExternalDocument

xtdExternalDocument is used to represent external documents, books or other written information.

EXAMPLE 1 "ISO 31" is a document.

EXAMPLE 2 "NS 3420" is a document.

EXPRESS specification:

```
*)
  ENTITY xtdExternalDocument;
    UniqueID                   :xtdGlobalUniqueID;
    Names                      :SET [1:?] OF xtdName;
  END_ENTITY;
(*
```

Attribute definitions:

UniqueID
globally unique identifier for the external document

Names
set of language-dependent names of the external document

4.3.15 xtdLanguage

xtdLanguage represents the language in which names, descriptions, values and references are expressed including its designation in itself and in English.

EXPRESS specification:

```
*)
   ENTITY xtdLanguage;
     LanguageNameInEnglish      :xtdLabel;
     LanguageNameInSelf         :OPTIONAL xtdLabel;
     Comments                   :OPTIONAL LIST [1:?] OF xtdText;
     UniqueID                   :xtdGlobalUniqueID;
   END_ENTITY;
(*
```

Attribute definitions:

LanguageNameInEnglish
name of the language in which the identifying descriptive attribute is expressed in the form in which the language is known in English

The name should be interpreted in a non-case sensitive way.

Examples of English language names, according to ISO 3166-1[1], are given below.

EXAMPLE 1 “English” is the ISO 3166-1 name for all types of English.

EXAMPLE 2 “German” is the name in English of the language spoken in Germany and surrounding countries.

LanguageNameInSelf
name of the language in which the identifying descriptive attribute is expressed in the form in which the language is known in itself

EXAMPLE 1 “Deutsch” is the name by which the German language is known to German language speakers.

EXAMPLE 2 “Norsk nynorsk” is the name of a particular form of the Norwegian language.

EXAMPLE 3 “British English” is the name of the form of the English language which is characterized by the spelling and intonation used within the British Isles and certain other parts of the world.

Comments
optional list of comments for the language used

UniqueID
globally unique identifier for the language

4.3.16 xtdLanguageRepresentation

xtdLanguageRepresentation is a representation of the language used to specify a name, a description, a value or a reference.

EXPRESS specification:

```
*)
   ENTITY xtdLanguageRepresentation
     ABSTRACT SUPERTYPE OF (ONEOF(xtdName, xtdDescription));
     LanguageName               :xtdLanguage;
     UniqueID                   :xtdGlobalUniqueID;
   END_ENTITY;
(*
```

Attribute definitions:

LanguageName
name of the language used for the xtdLanguageRepresentation

UniqueID
globally unique identifier for the language representation

4.3.17 xtdMeasureWithUnit

xtdMeasureWithUnit is a specialization of xtdObject that makes an xtdProperty measurable by associating a unit to a value.

EXAMPLE 1 "1m" is an instance of xtdMeasureWithUnit with a unit component "m" and a value component "1".

EXAMPLE 2 "AB88" is an instance of xtdMeasureWithUnit with a value component "AB88" and no unit component.

EXPRESS specification:

```
*)
  ENTITY xtdMeasureWithUnit
    SUBTYPE OF(xtdObject);
    UnitComponent              :OPTIONAL xtdUnit;
    ValueDomain                :OPTIONAL LIST [1:?] OF xtdValue;
  END_ENTITY;
(*
```

Attribute definitions:

UnitComponent
unit in which the value is expressed

ValueDomain
value domain for the measure for the given unit

4.3.18 xtdName

xtdName is a language-dependent name given to an object. An object may have several names in the same language and names do not need to be unique for a single object.

EXAMPLE 1 "beam" and "truss" are both English names for the same object.

EXAMPLE 2 "bjelke" is the Norwegian name for the English "beam".

EXPRESS specification:

```
*)
  ENTITY xtdName
    SUBTYPE OF(xtdLanguageRepresentation);
    Name                       :xtdLabel;
  INVERSE
    is_name_of                 :SET [1:?] OF xtdRoot FOR Names;
  END_ENTITY;
(*
```

Attribute definition:

Name
label holding the name of the object

4.3.19 xtdNest

xtdNest is a specialization of xtdCollection that only allows things of the same type to be members of the collection.

EXAMPLE "Window properties" is an xtdNest whose members are only instances of xtdProperty.

EXPRESS specification:

```
*)
  ENTITY xtdNest
    SUBTYPE OF(xtdCollection);
  WHERE
    WR1       :SIZEOF(QUERY(Result <* SELF\xtdRelCollects.RelatedThings |
               NOT(TYPEOF(SELF\xtdRelCollects.RelatingCollection) =
               TYPEOF(Result)))) = 0 ;
  END_ENTITY;
(*
```

Formal Propositions:

WR1 The type of the Collection shall always be the same as the type of each thing, i.e. the Collection and all things that are members of it are of the same type.

4.3.20 xtdObject

xtdObject is a specialization of xtdRoot that is the abstract entity from which xtdProperty, xtdMeasureWithUnit, xtdActor, xtdUnit, xtdSubject, xtdValue and xtdActivity are derived.

EXPRESS specification:

```
*)
  ENTITY xtdObject
    ABSTRACT SUPERTYPE OF (ONEOF(xtdSubject, xtdActivity, xtdUnit,
                      xtdProperty, xtdMeasureWithUnit, xtdActor, xtdValue))
    SUBTYPE OF(xtdRoot);
  END_ENTITY;
(*
```

4.3.21 xtdProperty

xtdProperty is a specialization of xtdObject that is used to qualify or quantify an xtdObject.

EXAMPLE 1 "Width" is an xtdProperty.

EXAMPLE 2 "Door width" is an xtdProperty used to qualify the "Width" of the xtdSubject "door".

EXAMPLE 3 "Heat transfer" is an xtdProperty.

EXAMPLE 4 "Colour" is an xtdProperty.

EXAMPLE 5 "Duration" is an xtdProperty.

EXAMPLE 6 "Comfort" is an xtdProperty.

EXPRESS specification:

```
*)
  ENTITY xtdProperty
    SUBTYPE OF(xtdObject);
  END_ENTITY;
(*
```

4.3.22 xtdRelActsUpon

xtdRelActsUpon is a specialization of xtdRelAssociates representing one xtdObject acting upon one or more other xtdObjects.

EXAMPLE 1 "A column supports a beam".

EXAMPLE 2 "A bricklayer lays bricks".

EXPRESS specification:

```
*)
  ENTITY xtdRelActsUpon
    SUBTYPE OF(xtdRelAssociates);
  WHERE
    WR1      :SIZEOF(QUERY(Result <* RelatedObjects | RelatingObject:=:
              Result)) = 0;
  END_ENTITY;
(*
```

Formal Propositions:

WR1 The instance to which the attribute relating object points shall not be contained in the set of related objects.

4.3.23 xtdRelAssignsCollections

xtdRelAssignsCollections is a specialization of xtdRelationship that represents the assignment of one or more instances of xtdCollection to an xtdObject.

EXPRESS specification:

```
*)
  ENTITY xtdRelAssignsCollections
    SUBTYPE OF(xtdRelationship);
    RelatingObject             :xtdObject;
    RelatedCollections         :SET [1:?] OF xtdCollection;
  END_ENTITY;
(*
```

Attribute definitions:

RelatingObject
object to which the collections are assigned

RelatedCollections
set of collections that is assigned to an object

4.3.24 xtdRelAssignsMeasures

xtdRelAssignsMeasures is a specialization of xtdRelationship that represents the assignment of one or more instances of xtdMeasureWithUnit to an xtdProperty.

EXPRESS specification:

```
*)
   ENTITY xtdRelAssignsMeasures
     SUBTYPE OF(xtdRelationship);
     RelatingProperty           :xtdProperty;
     RelatedMeasures            :SET [1:?] OF xtdMeasureWithUnit;
     MethodOfInterpretation     :OPTIONAL xtdName;
   END_ENTITY;
(*
```

Attribute definitions:

RelatingProperty
property to which the measures with unit are assigned

RelatedMeasures
set of measures with unit that is assigned to a property

MethodOfInterpretation
optional explanation for the value component of a measure with unit

EXAMPLE "Upper bound" is the method of interpretation for a measure with unit representing the upper bound of a bounded value.

4.3.25 xtdRelAssignsProperties

xtdRelAssignsProperties is a specialization of xtdRelationship that represents the assignment of one or more instances of xtdProperty to an xtdObject.

EXAMPLE "door height" is an xtdProperty assigned to the xtdObject "door".

EXPRESS specification:

```
*)
   ENTITY xtdRelAssignsProperties
     SUBTYPE OF(xtdRelationship);
     RelatedProperties          :SET [1:?] OF xtdProperty;
     RelatingObject             :xtdObject;
   END_ENTITY;
(*
```

Attribute definitions:

RelatedProperties
set of properties that is assigned to an object

RelatingObject
object to which the properties are assigned

4.3.26 xtdRelAssignsPropertyWithValues

xtdRelAssignsPropertyWithValues is a specialization of xtdRelationship that is a ternary relationship representing the assignment of one or more related values from a complete set of values of a related property on to a relating object.

EXPRESS specification:

```
*)
  ENTITY xtdRelAssignsPropertyWithValues
    SUBTYPE OF(xtdRelationship);
    RelatedProperty           :xtdProperty;
    RelatingObject            :xtdObject;
    RelatedValues             :LIST [1:?] OF UNIQUE xtdValue;
  END_ENTITY;
(*
```

Attribute definitions:

RelatedProperty
property that contains the complete list of values in which RelatedValues is selected

RelatingObject
object to which the list of values is assigned

RelatedValues
list of values selected from the complete set of values of the related property

4.3.27 xtdRelAssignsUnits

xtdRelAssignsUnits is a specialization of xtdRelationship that represents the assignment of one or more instances of xtdUnit to an xtdMeasureWithUnit.

EXPRESS specification:

```
*)
  ENTITY xtdRelAssignsUnits
    SUBTYPE OF(xtdRelationship);
    RelatingMeasure           :xtdMeasureWithUnit;
    RelatedUnits              :SET [1:?] OF xtdUnit;
  END_ENTITY;
(*
```

Attribute definitions:

RelatingMeasure
measure to which the units are assigned

RelatedUnits
set of units that is assigned to a measure

4.3.28 xtdRelAssignsValues

xtdRelAssignsValues is a specialization of xtdRelationship that represents the assignment of one or more instances of xtdValue to an xtdMeasureWithUnit.

EXPRESS specification:

```
*)
  ENTITY xtdRelAssignsValues
    SUBTYPE OF(xtdRelationship);
    RelatingMeasure           :xtdMeasureWithUnit;
    RelatedValues             :LIST [1:?] OF UNIQUE xtdValue;
  END_ENTITY;
(*
```

Attribute definitions:

RelatingMeasure
measure to which the values are assigned

RelatedValues
list of values that is assigned to a measure

4.3.29 xtdRelAssociates

xtdRelAssociates is a specialization of xtdRelationship that represents the association of a relating xtdObject to a set of related instances of xtdObject.

EXPRESS specification:

```
*)
  ENTITY xtdRelAssociates
    SUPERTYPE OF (ONEOF(xtdRelComposes, xtdRelGroups, xtdRelSpecializes,
                        xtdRelActsUpon))
    SUBTYPE OF(xtdRelationship);
    RelatingObject             :xtdObject;
    RelatedObjects             :SET [1:?] OF xtdObject;
  WHERE
    WR1      :SIZEOF(QUERY(Result <* RelatedObjects | RelatingObject:=:
              Result)) = 0;
  END_ENTITY;
(*
```

Attribute definitions:

RelatingObject
object that is the target of the relationship

RelatedObjects
set of objects that is associated with a target object

Formal Propositions:

WR1 The instance to which the attribute relating object points shall not be contained in the set of related objects.

4.3.30 xtdRelCollects

xtdRelCollects is a specialization of xtdRelationship that represents the collecting of arbitrary entities into an xtdCollection. xtdCollection is instantiated as either an xtdNest or an xtdBag.

EXPRESS specification:

```
*)
  ENTITY xtdRelCollects
    SUBTYPE OF(xtdRelationship);
    RelatedThings              :SET [1:?] OF xtdRoot;
    RelatingCollection         :xtdCollection;
  END_ENTITY;
(*
```

Attribute definitions:

RelatedThings
set of things making up a collection

RelatingCollection
collection of things where each instance in the collection is an object or relationship

4.3.31 xtdRelComposes

xtdRelComposes is a specialization of xtdRelAssociates that represents an xtdObject as being composed of other instances of xtdObject. The composition relationship can be applied recursively so that a composed xtdObject can be a part of another composed xtdObject.

EXAMPLE A "door frame" and a "door leaf" are parts of the (whole) "door set".

EXPRESS specification:

```
*)
  ENTITY xtdRelComposes
    SUBTYPE OF(xtdRelAssociates);
  WHERE
    WR1      :SIZEOF(QUERY(Result <* SELF.RelatedObjects |
              NOT(TYPEOF(SELF.RelatingObject) = TYPEOF(Result)))) = 0 ;
    WR2      :SIZEOF(QUERY(Result2 <* RelatedObjects| RelatingObject:=:
              Result2)) = 0;
  END_ENTITY;
(*
```

Formal Propositions:

WR1 The type of the relating object shall always be the same as the type of each related object, i.e. instances of xtdSubject are composed only of other instances of xtdSubject.

WR2 The instance to which the attribute relating object points shall not be contained in the list of related objects.

4.3.32 xtdRelDocuments

xtdRelDocuments is a specialization of xtdRelationship that handles the documentation of objects by relating objects to the documents where they are described. Each document can relate to many objects.

EXPRESS specification:

```
*)
  ENTITY xtdRelDocuments
    SUBTYPE OF(xtdRelationship);
    RelatedObjects             :SET [1:?] OF xtdObject;
    RelatingDocument           :xtdExternalDocument;
  END_ENTITY;
(*
```

Attribute definitions:

RelatedObjects
set of objects that are described in the external document

RelatingDocument
external document describing the related objects

4.3.33 xtdRelGroups

xtdRelGroups is a specialization of xtdRelAssociates that represents the grouping of a set of related instances of xtdObject into a relating xtdObject.

EXPRESS specification:

```
*)
  ENTITY xtdRelGroups
    SUBTYPE OF(xtdRelAssociates);
  WHERE
    WR1      :SIZEOF(QUERY(Result <* RelatedObjects | RelatingObject:=:
              Result)) = 0;
  END_ENTITY;
(*
```

Formal Propositions:

WR1 The instance to which the attribute relating object points shall not be contained in the List of related objects.

4.3.34 xtdRelSequences

xtdRelSequences is a specialization of xtdRelationship that represents the sequencing between two instances of xtdActivity. The relating xtdActivity precedes the related xtdActivity in a sequence.

EXPRESS specification:

```
*)
  ENTITY xtdRelSequences
    SUBTYPE OF(xtdRelationship);
    RelatingActivity          :OPTIONAL xtdActivity;
    RelatedActivity           :OPTIONAL xtdActivity;
  WHERE
    WR1      :RelatedActivity:<>: RelatingActivity;
    WR2      :EXISTS(RelatingActivity) OR EXISTS(RelatedActivity);
  END_ENTITY;
(*
```

Attribute definitions:

RelatingActivity
reference to the preceding activity in a sequence

RelatedActivity
reference to the succeeding activity in a sequence

Formal Propositions:

WR1 The relating activity shall not point to the same instance as the related activity.

WR2 Either the relating activity or the related activity must be asserted.

4.3.35 xtdRelSpecializes

xtdRelSpecializes is a specialization of xtdRelAssociates that represents the specialization of a supertype into one or more subtypes, ensuring that a subtype is more constrained than the supertype.

EXAMPLE 1 “outer door” is a specialization of “door”.

EXAMPLE 2 “space” is the generalization of “room”.

EXPRESS specification:

```
*)
  ENTITY xtdRelSpecializes
    SUBTYPE OF(xtdRelAssociates);
  WHERE
    WR1      :SIZEOF(QUERY(Result <* RelatedObjects | RelatingObject:=:
              Result)) = 0;
    WR2      :SIZEOF(QUERY(Result <* SELF.RelatedObjects |
              NOT(TYPEOF(SELF.RelatingObject) = TYPEOF(Result)))) = 0 ;
  END_ENTITY;
(*
```

Formal Propositions:

WR1 The instance to which the attribute relating object points shall not be contained in the list of related objects.

WR2 The type of the relating object shall always be the same as the type of each related object, i.e. an instance of xtdSubject can only be a specialization of another instance of xtdSubject.

4.3.36 xtdRelationship

xtdRelationship is a specialization of xtdRoot. It is an abstract entity that represents the objectification of relationships, allowing for additional attributes to be added to the entity.

EXPRESS specification:

```
*)
  ENTITY xtdRelationship
    ABSTRACT SUPERTYPE OF (ONEOF(xtdRelAssociates, xtdRelAssignsProperties,
                      xtdRelAssignsMeasures, xtdRelCollects,
                      xtdRelAssignsCollections, xtdRelSequences,
                      xtdRelDocuments, xtdRelAssignsUnits,
                      xtdRelAssignsValues, xtdRelAssignsPropertyWithValues))
    SUBTYPE OF(xtdRoot);
    ViewSelector               :OPTIONAL xtdName;
  END_ENTITY;
(*
```

Attribute definition:

ViewSelector
optional attribute used to qualify the relationship for the purpose of creating different views of the data

EXAMPLE The ViewSelector “IFC 2x2” can be used to identify a relationship originating from the IFC 2x2 model.

4.3.37 xtdRoot

xtdRoot is the abstract entity through which basic services of identifying, naming, versioning, description and referencing can be applied to instances of subtypes of xtdObject, xtdRelationship and xtdCollection.

EXPRESS specification:

```
*)
  ENTITY xtdRoot
    ABSTRACT SUPERTYPE OF (ONEOF(xtdObject, xtdRelationship,
                     xtdCollection));
    VersionDate                :OPTIONAL xtdDate;
    VersionID                  :OPTIONAL xtdVersionID;
    UniqueID                   :xtdGlobalUniqueID;
    Descriptions               :OPTIONAL SET [1:?] OF xtdDescription;
    Names                      :SET [1:?] OF xtdName;
  END_ENTITY;
(*
```

Attribute definitions:

VersionDate
date of the last revision of the reference data object

VersionID
version number for the reference data version of the object

UniqueID
global unique identifier for the object

Descriptions
optional set of language-dependent descriptions of the object

Names
set of language-dependent names of the object

4.3.38 xtdSubject

xtdSubject is a specialization of xtdObject that is subject to description by associated properties and activities. An xtdSubject can be physical or logical.

EXAMPLE 1 “Roof” is an xtdSubject.

EXAMPLE 2 “Lobby” is an xtdSubject.

EXAMPLE 3 “Control system” is an xtdSubject.

EXAMPLE 4 “Road” is an xtdSubject.

EXAMPLE 5 “Airport” is an xtdSubject.

EXAMPLE 6 “Software” is an xtdSubject.

EXPRESS specification:

```
*)
  ENTITY xtdSubject
    SUBTYPE OF(xtdObject);
  END_ENTITY;
(*
```

4.3.39 xtdUnit

xtdUnit is a specialization of xtdObject that represents a scale that enables a value to be measured.

EXAMPLE 1 "metre" is an xtdUnit which can be represented by the symbol "m".

EXAMPLE 2 "W/m.K" is an xtdUnit used with the property "thermal conductivity".

EXPRESS specification:

```
*)
  ENTITY xtdUnit
    SUBTYPE OF(xtdObject);
  END_ENTITY;
(*
```

4.3.40 xtdValue

xtdValue is a specialization of xtdLanguageRepresentation that can hold the description of a value of an xtdProperty.

EXAMPLE "Vertikaal schuivend" is the Dutch value for "Vertical sliding".

EXPRESS specification:

```
*)
  ENTITY xtdValue
    SUBTYPE OF(xtdObject);
    LowerTolerance              :OPTIONAL xtdText;
    NominalValue                :OPTIONAL xtdText;
    UpperTolerance              :OPTIONAL xtdText;
    ValueType                   :OPTIONAL xtdValueTypeEnum;
    ValueRole                   :OPTIONAL xtdValueRoleEnum;
    ToleranceType               :OPTIONAL xtdToleranceTypeEnum;
  END_ENTITY;
(*
```

Attribute definitions:

LowerTolerance
optional lower tolerance for the value

NominalValue
optional text string holding the nominal value

UpperTolerance
optional upper tolerance for the value

ValueType
optional select type to define the type of value

ValueRole
optional select type for the role of the value

ToleranceType
optional select type to define the type of tolerance for the value

*)

END_SCHEMA;

4.4 EXPRESS long form specification

This formal specification is provided in the EXPRESS language.

```
SCHEMA ISO_12006_3_VERSION_3;

  TYPE xtdDate = STRING;
  END_TYPE;

  TYPE xtdGlobalUniqueID = STRING;
  END_TYPE;

  TYPE xtdLabel = STRING;
  END_TYPE;

  TYPE xtdText = STRING;
  END_TYPE;

  TYPE xtdToleranceTypeEnum = ENUMERATION OF
    (REALVALUE,
     PERCENTAGE);
  END_TYPE;

  TYPE xtdValueRoleEnum = ENUMERATION OF
    (NOMINAL,
     MAXIMUM,
     MINIMUM);
  END_TYPE;

  TYPE xtdValueTypeEnum = ENUMERATION OF
    (XTDSTRING,
     XTDNUMBER,
     XTDINTEGER,
     XTDREAL,
     XTDBOOLEAN,
     XTDLOGICAL);
  END_TYPE;

  TYPE xtdVersionID = STRING;
  END_TYPE;

  ENTITY xtdActivity
    SUBTYPE OF(xtdObject);
  END_ENTITY;

  ENTITY xtdActor
    SUBTYPE OF(xtdObject);
  END_ENTITY;

  ENTITY xtdBag
    SUBTYPE OF(xtdCollection);
  END_ENTITY;

  ENTITY xtdCollection
    ABSTRACT SUPERTYPE OF (ONEOF(xtdNest, xtdBag))
    SUBTYPE OF(xtdRoot);
  END_ENTITY;

  ENTITY xtdDescription
    SUBTYPE OF(xtdLanguageRepresentation);
      Description       : xtdText;
```

```
    INVERSE
      is_description_of : SET [1:?] OF xtdRoot FOR Descriptions;
  END_ENTITY;

  ENTITY xtdExternalDocument;
      UniqueID : xtdGlobalUniqueID;
      Names    : SET [1:?] OF xtdName;
  END_ENTITY;

  ENTITY xtdLanguage;
      LanguageNameInEnglish : xtdLabel;
      LanguageNameInSelf    : OPTIONAL xtdLabel;
      Comments              : OPTIONAL LIST [1:?] OF xtdText;
      UniqueID              : xtdGlobalUniqueID;
  END_ENTITY;

  ENTITY xtdLanguageRepresentation
    ABSTRACT SUPERTYPE OF (ONEOF(xtdName, xtdDescription));
      LanguageName : xtdLanguage;
      UniqueID     : xtdGlobalUniqueID;
  END_ENTITY;

  ENTITY xtdMeasureWithUnit
    SUBTYPE OF(xtdObject);
      UnitComponent : OPTIONAL xtdUnit;
      ValueDomain   : OPTIONAL LIST [1:?] OF xtdValue;
  END_ENTITY;

  ENTITY xtdName
    SUBTYPE OF(xtdLanguageRepresentation);
      Name       : xtdLabel;
    INVERSE
      is_name_of : SET [1:?] OF xtdRoot FOR Names;
  END_ENTITY;

  ENTITY xtdNest
    SUBTYPE OF(xtdCollection);
    WHERE
      WR1    :    SIZEOF(QUERY(Result    <*    SELF\xtdRelCollects.RelatedThings    |
NOT(TYPEOF(SELF\xtdRelCollects.RelatingCollection) = TYPEOF(Result)))) = 0 ;
  END_ENTITY;

  ENTITY xtdObject
    ABSTRACT SUPERTYPE OF (ONEOF(xtdSubject, xtdActivity, xtdUnit, xtdProperty,
xtdMeasureWithUnit, xtdActor, xtdValue))
    SUBTYPE OF(xtdRoot);
  END_ENTITY;

  ENTITY xtdProperty
    SUBTYPE OF(xtdObject);
  END_ENTITY;

  ENTITY xtdRelActsUpon
    SUBTYPE OF(xtdRelAssociates);
    WHERE
      WR1 : SIZEOF(QUERY(Result <* RelatedObjects | RelatingObject:=: Result)) =
0;
  END_ENTITY;

  ENTITY xtdRelAssignsCollections
    SUBTYPE OF(xtdRelationship);
```

```
      RelatingObject     : xtdObject;
      RelatedCollections : SET [1:?] OF xtdCollection;
  END_ENTITY;

  ENTITY xtdRelAssignsMeasures
    SUBTYPE OF(xtdRelationship);
      RelatingProperty       : xtdProperty;
      RelatedMeasures        : SET [1:?] OF xtdMeasureWithUnit;
      MethodOfInterpretation : OPTIONAL xtdName;
  END_ENTITY;

  ENTITY xtdRelAssignsProperties
    SUBTYPE OF(xtdRelationship);
      RelatedProperties : SET [1:?] OF xtdProperty;
      RelatingObject    : xtdObject;
  END_ENTITY;

  ENTITY xtdRelAssignsPropertyWithValues
    SUBTYPE OF(xtdRelationship);
      RelatedProperty : xtdProperty;
      RelatingObject  : xtdObject;
      RelatedValues   : LIST [1:?] OF UNIQUE xtdValue;
  END_ENTITY;

  ENTITY xtdRelAssignsUnits
    SUBTYPE OF(xtdRelationship);
      RelatingMeasure : xtdMeasureWithUnit;
      RelatedUnits    : SET [1:?] OF xtdUnit;
  END_ENTITY;

  ENTITY xtdRelAssignsValues
    SUBTYPE OF(xtdRelationship);
      RelatingMeasure : xtdMeasureWithUnit;
      RelatedValues   : LIST [1:?] OF UNIQUE xtdValue;
  END_ENTITY;

  ENTITY xtdRelAssociates
    SUPERTYPE   OF   (ONEOF(xtdRelComposes,   xtdRelGroups,   xtdRelSpecializes,
xtdRelActsUpon))
    SUBTYPE OF(xtdRelationship);
      RelatingObject : xtdObject;
      RelatedObjects : SET [1:?] OF xtdObject;
    WHERE
      WR1 : SIZEOF(QUERY(Result <* RelatedObjects | RelatingObject:=: Result)) =
0;
  END_ENTITY;

  ENTITY xtdRelationship
    ABSTRACT SUPERTYPE OF (ONEOF(xtdRelAssociates, xtdRelAssignsProperties,
xtdRelAssignsMeasures, xtdRelCollects, xtdRelAssignsCollections, xtdRelSequences,
xtdRelDocuments, xtdRelAssignsUnits, xtdRelAssignsValues,
xtdRelAssignsPropertyWithValues))
    SUBTYPE OF(xtdRoot);
      ViewSelector : OPTIONAL xtdName;
  END_ENTITY;

  ENTITY xtdRelCollects
    SUBTYPE OF(xtdRelationship);
      RelatedThings      : SET [1:?] OF xtdRoot;
      RelatingCollection : xtdCollection;
  END_ENTITY;
```

```
  ENTITY xtdRelComposes
    SUBTYPE OF(xtdRelAssociates);
    WHERE
      WR1      :      SIZEOF(QUERY(Result      <*      SELF.RelatedObjects      |
NOT(TYPEOF(SELF.RelatingObject) = TYPEOF(Result)))) = 0 ;
      WR2 : SIZEOF(QUERY(Result2 <* RelatedObjects| RelatingObject:=: Result2)) =
0;
  END_ENTITY;

  ENTITY xtdRelDocuments
    SUBTYPE OF(xtdRelationship);
      RelatedObjects   : SET [1:?] OF xtdObject;
      RelatingDocument : xtdExternalDocument;
  END_ENTITY;

  ENTITY xtdRelGroups
    SUBTYPE OF(xtdRelAssociates);
    WHERE
      WR1 : SIZEOF(QUERY(Result <* RelatedObjects | RelatingObject:=: Result)) =
0;
  END_ENTITY;

  ENTITY xtdRelSequences
    SUBTYPE OF(xtdRelationship);
      RelatingActivity : OPTIONAL xtdActivity;
      RelatedActivity  : OPTIONAL xtdActivity;
    WHERE
      WR1 : RelatedActivity:<>: RelatingActivity;
      WR2 : EXISTS(RelatingActivity) OR EXISTS(RelatedActivity);
  END_ENTITY;

  ENTITY xtdRelSpecializes
    SUBTYPE OF(xtdRelAssociates);
    WHERE
      WR1 : SIZEOF(QUERY(Result <* RelatedObjects | RelatingObject:=: Result)) =
0;
      WR2      :      SIZEOF(QUERY(Result      <*      SELF.RelatedObjects      |
NOT(TYPEOF(SELF.RelatingObject) = TYPEOF(Result)))) = 0 ;
  END_ENTITY;

  ENTITY xtdRoot
    ABSTRACT SUPERTYPE OF (ONEOF(xtdObject, xtdRelationship, xtdCollection));
      VersionDate  : OPTIONAL xtdDate;
      VersionID    : OPTIONAL xtdVersionID;
      UniqueID     : xtdGlobalUniqueID;
      Descriptions : OPTIONAL SET [1:?] OF xtdDescription;
      Names        : SET [1:?] OF xtdName;
  END_ENTITY;

  ENTITY xtdSubject
    SUBTYPE OF(xtdObject);
  END_ENTITY;

  ENTITY xtdUnit
    SUBTYPE OF(xtdObject);
  END_ENTITY;

  ENTITY xtdValue
    SUBTYPE OF(xtdObject);
      LowerTolerance : OPTIONAL xtdText;
```

```
      NominalValue   : OPTIONAL xtdText;
      UpperTolerance : OPTIONAL xtdText;
      ValueType      : OPTIONAL xtdValueTypeEnum;
      ValueRole      : OPTIONAL xtdValueRoleEnum;
      ToleranceType  : OPTIONAL xtdToleranceTypeEnum;
  END_ENTITY;


END_SCHEMA;
```

Annex A
(informative)

Naming conventions

A.1 General

The following naming convention has been used in the development of this part of ISO 12006. These conventions do not apply to the population or use of the framework.

A.2 Characters used for names

The characters [A-Z] upper case, [a-z] lower case and [0-9] numeric are used for names.

A.3 Case of names

Names are written in upper and lower case characters as a single name without spaces.

The first character of each word in normal usage following the schema prefix is written as an upper case character. All other characters forming part of the same word in normal usage are written in lower case characters.

A.4 Length of names

The length of names is not restricted.

A.5 xtd prefix

Names are prefixed by the term “xtd” to identify their usage within the ISO 12006-3 information model. “xtd” is an acronym for extensible taxonomy definition.

A.6 Names of entities other than relationship entities

The name of an entity is a noun or combination of nouns denoting the “content” or “type” of the entity.

A.7 Names of relationship entities

Entities acting as relationships within the model contain the term “Rel” following the “xtd” prefix and before the name of the entity in normal usage. The inserted “Rel” is treated as a word in normal usage and the “Case of names” rule applied. Relationships are named starting from the related (SET [1:?]) of objects ending on the relating object, i.e. xtdRelComposes means that the relationship composes the parts (SET [1:?]) into a whole.

The name of the relationship entity is a verb or verb phrase that denotes the “function” of the relationship.

A.8 Plural naming of aggregation

Names of attributes and relationships within an aggregation are expressed in the plural form.

Bibliography

[1] ISO 3166-1, *Codes for the representation of names of countries and their subdivisions — Part 1: Country codes*

[2] ISO 10303-41 *Industrial automation systems and integration — Product data representation and exchange — Part 41: Integrated generic resource: Fundamentals of product description and support*

[3] ISO 10303-221, *Industrial automation systems and integration — Product data representation and exchange — Part 221: Application protocol: Functional data and their schematic representation for process plants*

[4] ISO/IEC 10646:2003, *Information technology — Universal Multiple-Octet Coded Character Set (UCS), Annex D* (Technically equivalent to the definitions in the Unicode Standard)

[5] ISO 12006-2, *Building Construction — Organization of information about construction works — Part 2: Framework for classification of information*

[6] ISO 15926-2, *Industrial automation systems and integration — Integration of life-cycle data for process plants including oil and gas production facilities — Part 2: Data model*

[7] ISO/PAS 16739, *Industry Foundation Classes, Release 2x, Platform Specification (IFC2xPlatform)*

[8] The Unicode Consortium. *The Unicode Standard, Version 4.0*. Boston, MA, Addison-Wesley, 2003, ISBN 0-321-18578-1

ICS 91.010.01

Price based on 32 pages