

ISO/TC 59/SC 13/ **WG6**

Date: 2002-07-22

WG6-104

ISO/PAS 12006-3 version 2
Supersedes ISO/PAS 12006-3 version 1

Secretariat: Norway

Building construction — Organization of information about construction works — Part 3: Framework for object-oriented information exchange — revised version 2

Warning

This document is not an ISO International Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an International Standard. However ISO/TC59/SC13/Workgroup 6 considers this version 2 'frozen' and has submitted this PAS for DIS.

Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

ISO PAS 12006-3 version 2 supersedes version 1 (ISO/PAS 12006-3:2001). In the new version 2 the information model is completely revised with regards to version 1.

Document type: Publicly Available Specification (PAS)
Document subtype: Not applicable
Document stage: Submitted by TC59/SC13/WG6 for Draft International Standard (DIS)
Document language: E

Copyright notice

This ISO document is a working draft or committee draft and is copyright-protected by ISO. While the reproduction of working drafts or committee drafts in any form for use by participants in the ISO standards development process is permitted without prior permission from ISO, neither this document nor any extract from it may be reproduced, stored or transmitted in any form for any other purpose without prior written permission from ISO.

Requests for permission to reproduce this document for the purpose of selling it should be addressed as shown below or to ISO's member body in the country of the requester:

*[Indicate :
the full address
telephone number
fax number
telex number
and electronic mail address*

as appropriate, of the Copyright Manager of the ISO member body responsible for the secretariat of the TC or SC within the framework of which the draft has been prepared]

Reproduction for sales purposes may be subject to royalty payments or a licensing agreement.

Violators may be prosecuted.

Contents

Foreword.....	vi
Introduction.....	ix
1 Scope	1
2 Normative references	1
3 Terms and definitions	1
4 Specification	1
ISO_PAS_12006_3_VERSION_2.....	7
xtdBoolean	7
xtdDate.....	7
xtdGlobalUniqueID	7
xtdInteger.....	8
xtdLabel	8
xtdLogical	8
xtdNumber	8
xtdReal	9
xtdString	9
xtdText	9
xtdVersionID.....	9
xtdSimpleValue	9
xtdActivity.....	10
xtdActor	10
xtdBag	10
xtdCollection	11
xtdDescription.....	11
xtdEnumeration	11
xtdLanguage	12
xtdLanguageRepresentation	13
xtdMeasureWithUnit	13
xtdName.....	13
xtdNest.....	14
xtdObject	14
xtdProperty.....	15
xtdReference	15
xtdRelActsUpon	15
xtdRelAssignsCollections	16
xtdRelAssignsMeasures	16
xtdRelAssignsProperties	17
xtdRelAssociates	17
xtdRelCollects	18
xtdRelComposes	18
xtdRelGroups	19
xtdRelSequences	19
xtdRelSpecializes	20
xtdRelationship	20
xtdRelationshipRole	21
xtdRoot	21
xtdSubject.....	22
xtdUnit	22
xtdValue	22

Conventions	26
1 Names	26
2 Single inheritance of subtypes.....	27
3 Exclusion constraint in supertypes	27
4 Many-to-many relationships	27

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 3.

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.

In other circumstances, particularly when there is an urgent market requirement for such documents, a technical committee may decide to publish other types of normative document:

- an ISO Publicly Available Specification (ISO/PAS) represents an agreement between technical experts in an ISO working group and is accepted for publication if it is approved by more than 50% of the members of the parent committee casting a vote;
- an ISO Technical Specification (ISO/TS) represents an agreement between the members of a technical committee and is accepted for publication if it is approved by 2/3 of the members of the committee casting a vote.

An ISO/PAS or ISO/TS is reviewed every three years with a view to deciding whether it can be transformed into an International Standard.

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

ISO/PAS 12006-3 was prepared by Technical Committee ISO/TC 59 Building Construction, Subcommittee SC 13 Organization of information about construction works, Working Group 6.

This second version is meant to be published as an International Standard. It cancels and replaces the first edition (ISO/PAS 12006-3 version 1 d.d. November 2000) completely. The major technical changes from the first edition are the following:

- changes in the EXPRESS model as a result of the discussion with regard to the raised issues;
- replacement of the Lex prefix by the prefix xtd, where xtd is an acronym for extensible taxonomy definition.

ISO/PAS 12006-3 is a companion to ISO/DIS 12006-2, *Organization of information about construction works – Part 2: Framework for classification of information*, which provides a framework for a classification, rather than the object-oriented approach of this standard. The two standards should be regarded as complementary rather than contradictory. Each has been developed and published in the belief that it has an important role to play in the organization of information about construction works¹.

¹ ISO work item 12006-1 has been terminated some time ago.

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 given in English (the default language). An object may also be named in terms of the language of the location in which its is determined or used. Objects may be related to formal classification systems through the provision of references.

The model has one Root class from which the following three classes inherit: Objects, Collections and the Relationships between them. The Root class 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, Measures with Units and Properties. Subjects and Activities are the things and processes that are described. The other classes are description classes 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 classes 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 to a Property through a Measure with Unit, will be stored in the Value component, which is language dependent and therefore derived from the Language Representation class. The latter class models the way any Name, Description, Value or Reference is represented on a per language base.

1 Scope

This Publicly Available Specification specifies a language-independent information model which can be used for the development of vocabularies used in information about construction works.

ISO/PAS 12006-3 enables classification systems, information models, object models and process models to be referenced from within a common framework.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this PAS. At the time of publication, the editions indicated were valid.

ISO 10303–11:1994 *Industrial automation systems and integration – Product data representation and exchange – Part 11: Description methods: The EXPRESS language reference manual.*

3 Terms and definitions

For the purposes of this part of ISO 12006, the terms and definitions provided in ISO 10303 -11 apply.

4 Specification

ISO/PAS 12006-3, version 2 is described in the next five EXPRESS-G diagrams. Following those, the formal definition is given in EXPRESS.

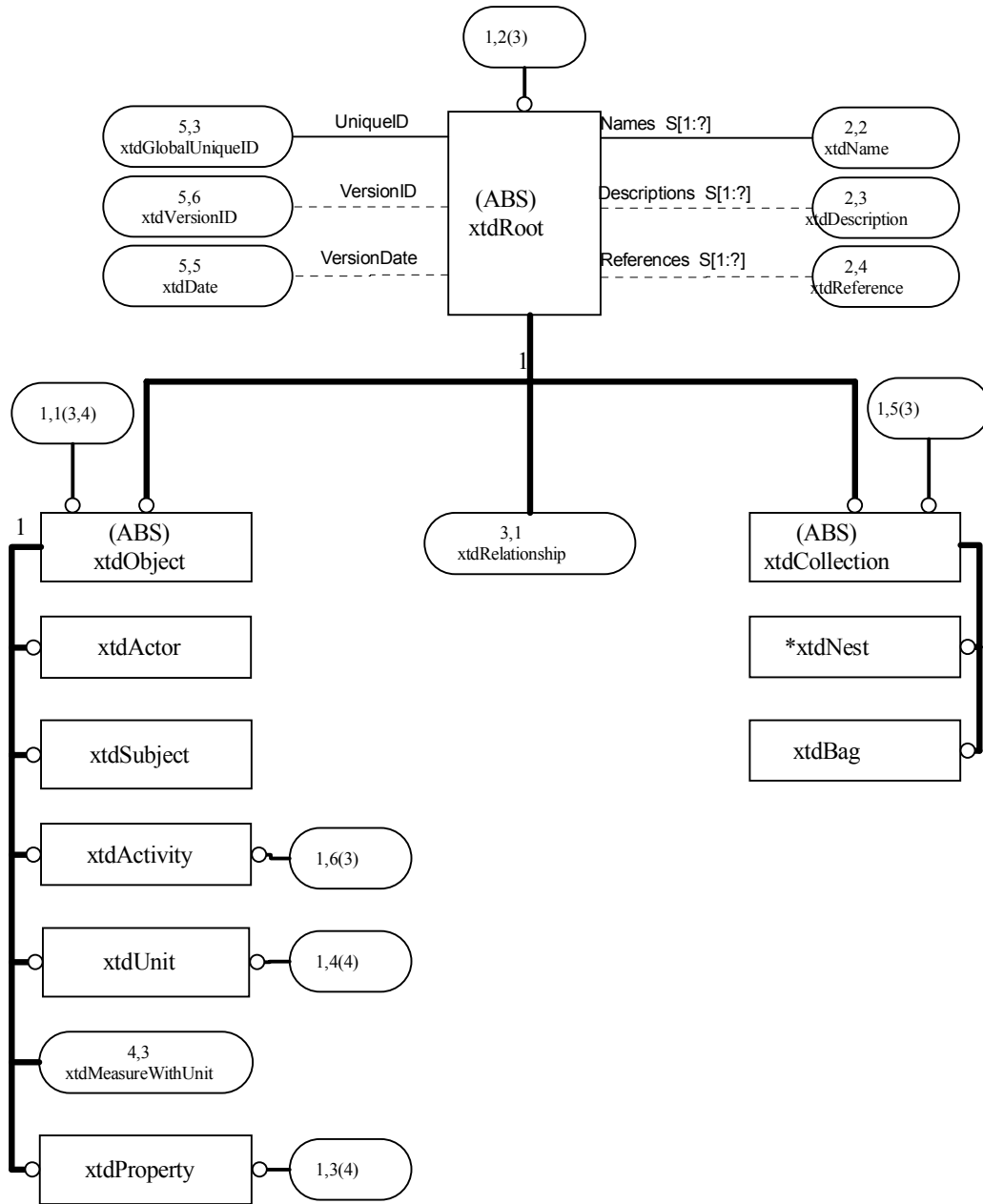


Figure D.1 - ISO_PAS_12006_3_VERSION_1 EXPRESS-G diagram 1 of 5

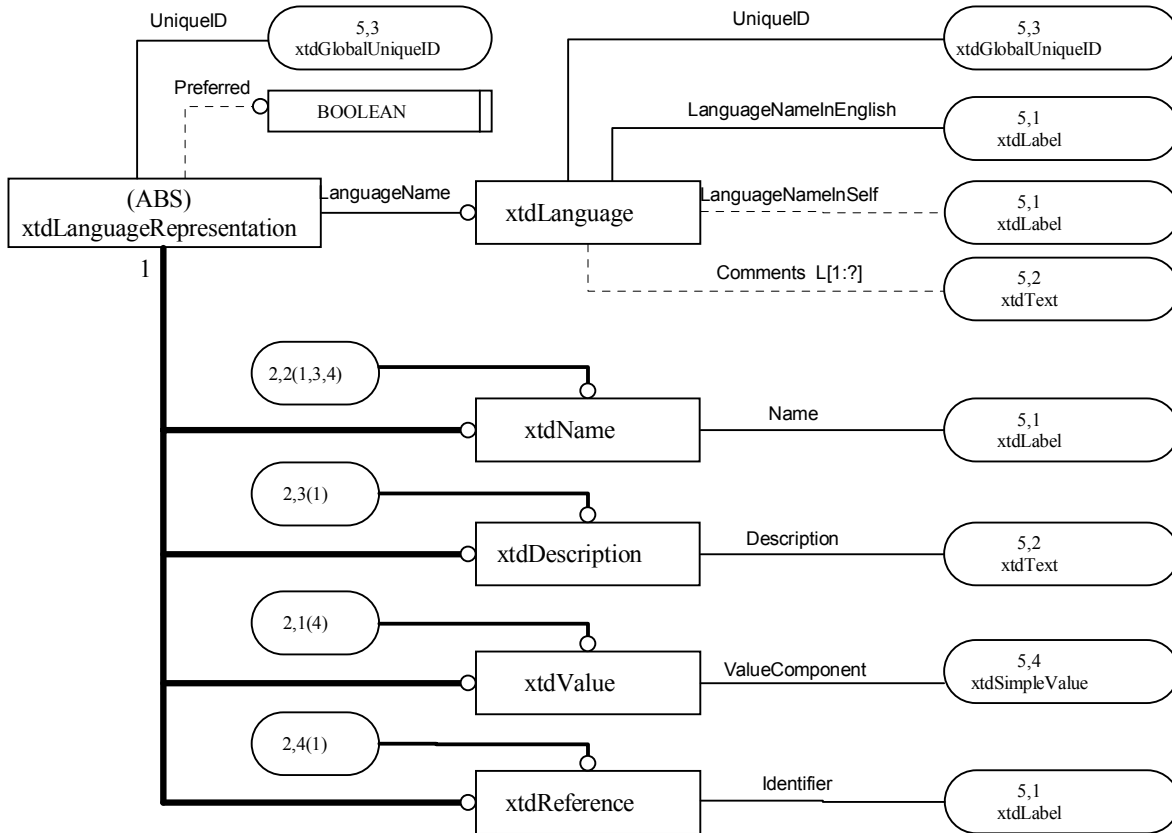


Figure D.2 - ISO_PAS_12006_3_VERSION_1 EXPRESS-G diagram 2 of 5

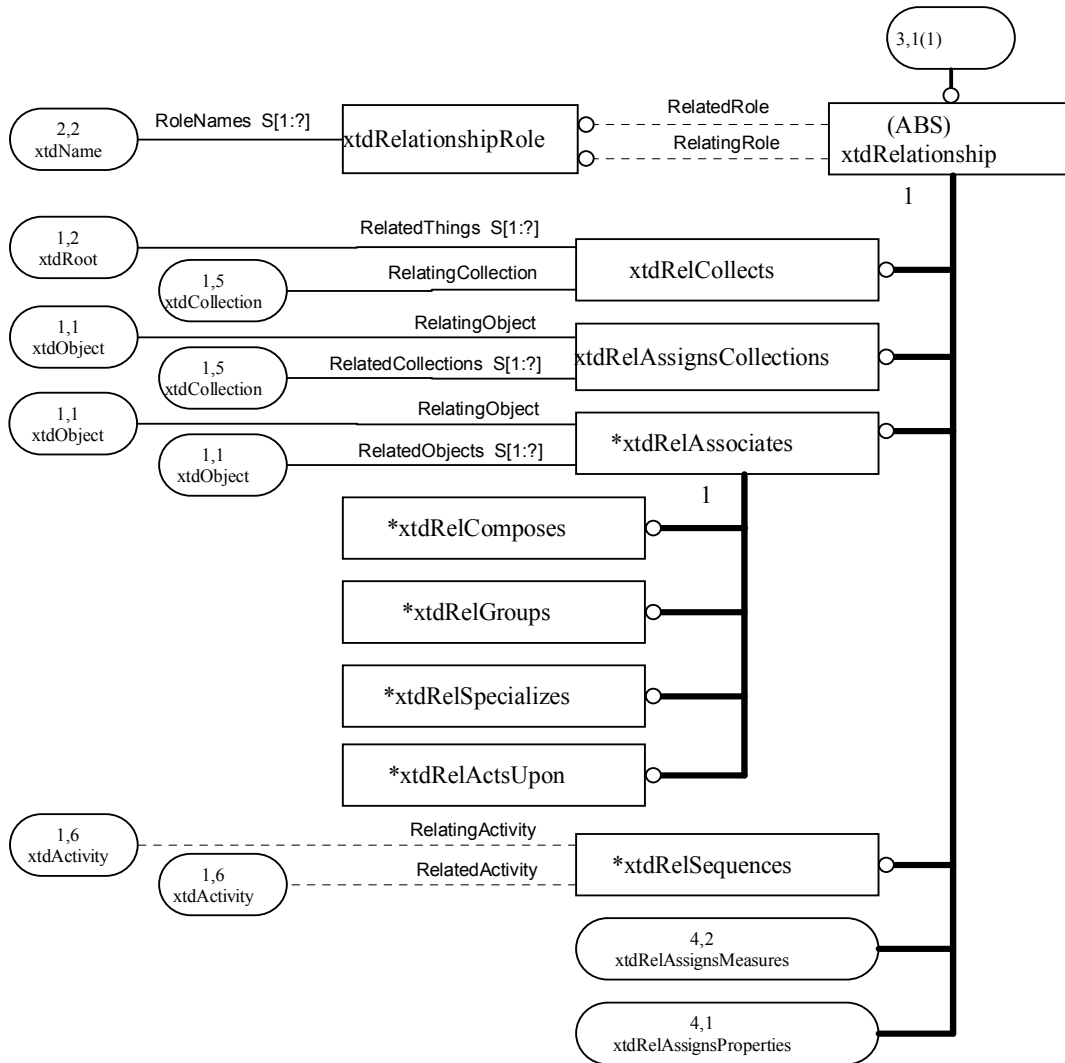


Figure D.3 - ISO_PAS_12006_3_VERSION_1 EXPRESS-G diagram 3 of 5

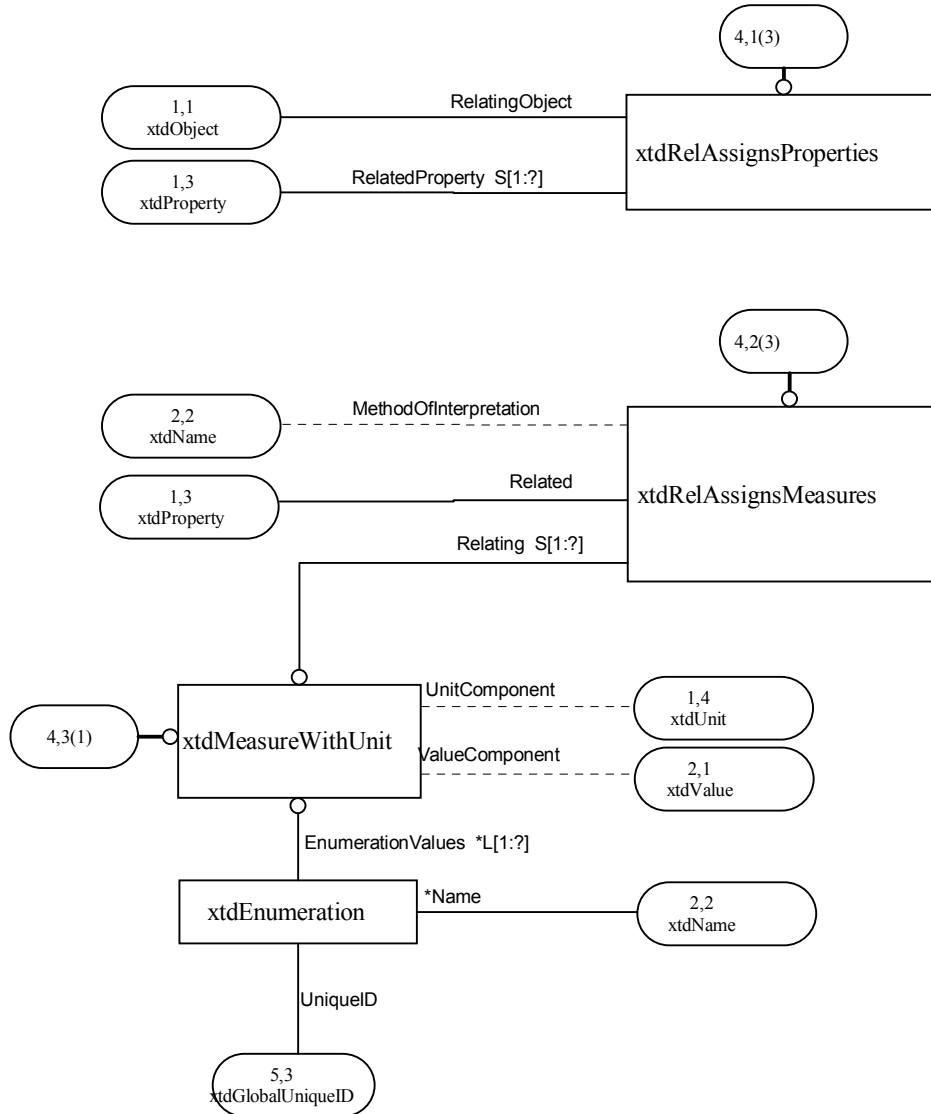


Figure D.4 - ISO_PAS_12006_3_VERSION_1 EXPRESS-G diagram 4 of 5

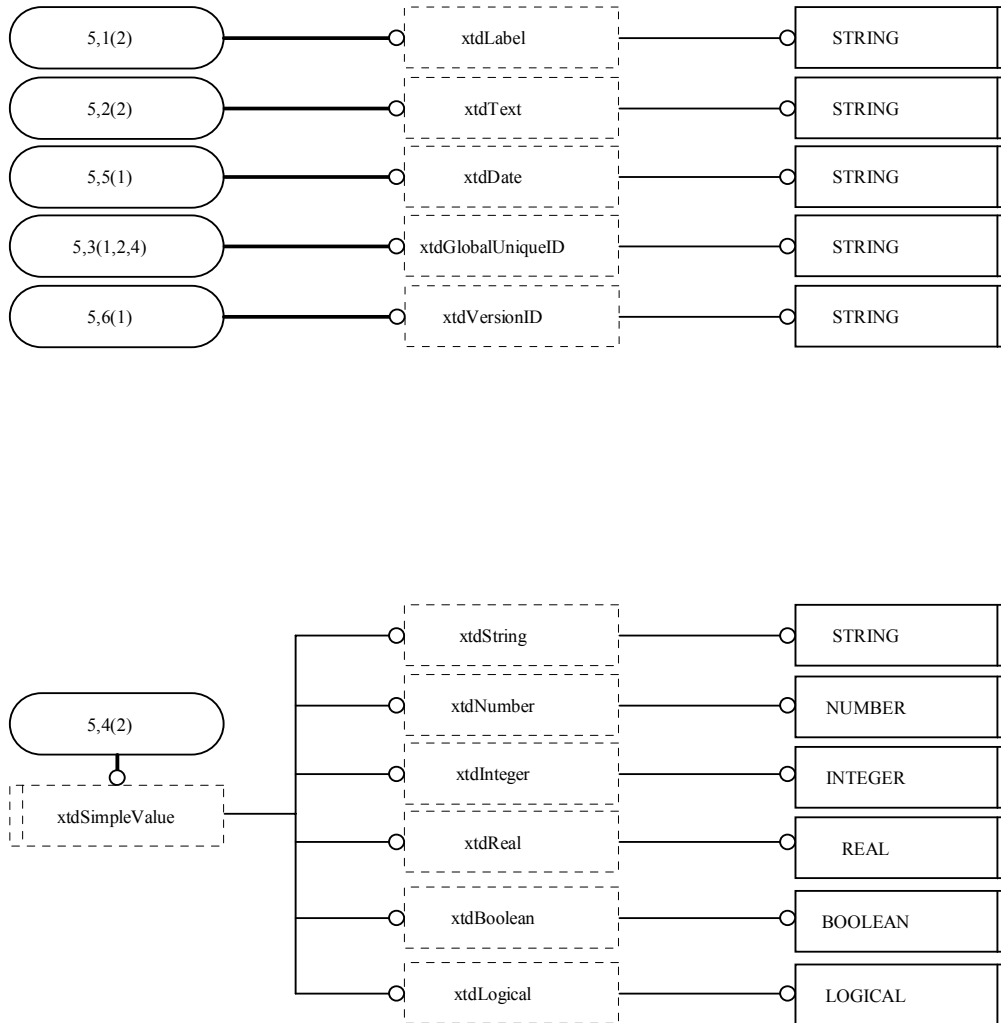


Figure D.5 - ISO_PAS_12006_3_VERSION_1 EXPRESS-G diagram 5 of 5

EXPRESS specification:

(*)

ISO_PAS_12006_3_VERSION_2EXPRESS specification:

*)

```
{XE "ISO_PAS_12006_3_VERSION_2" \b}SCHEMA ISO_PAS_12006_3_VERSION_2;
```

(*)

xtdBoolean

A defined data type of simple data type Boolean. A boolean type can have value TRUE or FALSE.

EXPRESS specification:

```
*){XE "xtdBoolean" \b}
```

```
    TYPE xtdBoolean = BOOLEAN;
    END_TYPE;
```

(*)

xtdDate

xtdDate is the date of last revision defined on the format YYYY.MM.DD

EXAMPLE:

The 31st day of May in the year 2000 should be written as 2000.05.31

NOTE:

Date may be more precisely defined by use of a more detailed schema such as that found in the date_time schema of ISO 10303 Part 41.

EXPRESS specification:

```
*){XE "xtdDate" \b}
```

```
    TYPE xtdDate = STRING;
    END_TYPE;
```

(*)

xtdGlobalUniqueID

The xtdGlobalUniqueID holds an identifier that is unique throughout the software world. This is also known as a Universal Unique Identifier by the Open Group. The identifier is generated using an algorithm published by the Object Management Group based on the IP address of the computer that generates the identifier.

EXAMPLE:

The 32 characters string "93F09E4AA899402A99D013A518FDCAE8" is a global unique identifier (GUID)

NOTE:

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

EXPRESS specification:

```
*){XE "xtdGlobalUniqueID" \b}
```

```
    TYPE xtdGlobalUniqueID = STRING;  
    END_TYPE;
```

```
(*
```

xtdInteger

A defined type of simple data type Integer.

EXPRESS specification:

```
*){XE "xtdInteger" \b}
```

```
    TYPE xtdInteger = INTEGER;  
    END_TYPE;
```

```
(*
```

xtdLabel

xtdLabel is a set of alphanumeric UNICODE characters that identify an object.

EXPRESS specification:

```
*){XE "xtdLabel" \b}
```

```
    TYPE xtdLabel = STRING;  
    END_TYPE;
```

```
(*
```

xtdLogical

A defined type of simple type logical. Logical datatype can have values TRUE, FALSE or UNKNOWN.

EXPRESS specification:

```
*){XE "xtdLogical" \b}
```

```
    TYPE xtdLogical = LOGICAL;  
    END_TYPE;
```

```
(*
```

xtdNumber

A defined type of simple data type number.

EXPRESS specification:

```
*){XE "xtdNumber" \b}
```

```
    TYPE xtdNumber = NUMBER;  
    END_TYPE;
```

```
(*
```

xtdReal

A defined type of simple data type real. The domain of xtdReal is all rational, irrational and scientific real numbers.

EXPRESS specification:

```
*){XE "xtdReal" \b}

    TYPE xtdReal = REAL;
    END_TYPE;
(*
```

xtdString

An alphanumeric string of characters. A defined type of simple data type string.

EXPRESS specification:

```
*){XE "xtdString" \b}

    TYPE xtdString = STRING;
    END_TYPE;
(*
```

xtdText

xtdText is a set of alphanumeric UNICODE characters that describes or defines an object.

EXPRESS specification:

```
*){XE "xtdText" \b}

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

xtdVersionID

xtdVersionID is a set of alphanumeric characters used to define the identity of a version of an object. The version can be any combination of characters and numbers within the allowed UNICODE character set.

EXAMPLE:

"1A", "12", "1.1" are types of versioning.

EXPRESS specification:

```
*){XE "xtdVersionID" \b}

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

xtdSimpleValue

A select type for selection between simple value types.

NOTE:

xtdSimpleValue can be a string, number, integer, real, boolean, or logical.

EXPRESS specification:

```
*){XE "xtdSimpleValue" \b}{XE "xtdString (REF)"}{XE "xtdNumber (REF)"}{XE "xtdInteger (REF)"}{XE "xtdReal (REF)"}{XE "xtdBoolean (REF)"}{XE "xtdLogical (REF)"}
```

```
TYPE xtdSimpleValue = SELECT  
  (xtdString,  
   xtdNumber,  
   xtdInteger,  
   xtdReal,  
   xtdBoolean,  
   xtdLogical);  
END_TYPE;  
(*
```

xtdActivity

The xtdActivity can represent any activity or process. An activity is something that makes a change to a subject. The activity is similar to the concept verb in common speech.

EXAMPLE:

"door assembly" door assembly is an activity and may be used to describe the assembly of a door set.
"bricklaying" is an activity
"cleaning" is an activity
"building" is an activity
"pumping concrete" is an activity

EXPRESS specification:

```
*){XE "xtdActivity" \b}  
  
  ENTITY xtdActivity  
    SUBTYPE OF(xtdObject);  
  END_ENTITY;  
(*
```

xtdActor

xtdActor defines all actors or human agents involved in a project during its full life cycle. An actor can be, e.g. a person, profession, or organisation.

EXAMPLE:

"carpenter" is an xtdActor
"ISO" is an xtdActor

EXPRESS specification:

```
*){XE "xtdActor" \b}  
  
  ENTITY xtdActor  
    SUBTYPE OF(xtdObject);  
  END_ENTITY;  
(*
```

xtdBag

xtdBag is a generic grouping mechanism allowing grouping of any type of things. The xtdBag can

contain mixtures of different things.

EXPRESS specification:

```
*){XE "xtdBag" \b}

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

xtdCollection

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

EXPRESS specification:

```
*){XE "xtdCollection" \b}

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

xtdDescription

The xtdDescription is a language dependent description of the concept and may act as a textual definition of the concept.

EXAMPLE:

"A door leaf is the part of a door that opens to allow people or goods to pass" is the English explanation for the object "Door leaf"

EXPRESS specification:

```
*){XE "xtdText (REF)"}{XE "xtdDescription" \b}

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

Attribute definitions:

Description : A text being the description of the object.

xtdEnumeration

xtdEnumeration is an ordered list of measures with unit, from which a particular measure with unit may be selected.

EXPRESS specification:

```
*){XE "xtdMeasureWithUnit (REF)"}{XE "xtdName (REF)"}{XE "xtdGlobalUniqueID (REF)"}{XE "xtdEnumeration" \b}
```

```

ENTITY xtdEnumeration;
  EnumerationValues      : LIST [1:?] OF UNIQUE xtdMeasureWithUnit;
  Name                   : xtdName;
  UniqueID               : xtdGlobalUniqueID;
UNIQUE
  UR1                    : Name;
END_ENTITY;
(*

```

Attribute definitions:

EnumerationValues : List of measures with unit that form the enumeration.

Name : A set of language dependent names for the enumeration.

xtdLanguage

xtdLanguage is the language used for names, descriptions, values and references.

EXPRESS specification:

```

*){XE "xtdLabel (REF)"}{XE "xtdLabel (REF)"}{XE "xtdText (REF)"}{XE "xtdGlobalUniqueID (REF)"}{XE
"xtdLanguage" \b}

```

```

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

```

Attribute definitions:

LanguageNameInEnglish : The 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.

EXAMPLE:

"ENGLISH" is the ISO3166 name for all types of English

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

EXAMPLE:

"Deutsch" is the LanguageNameInSelf for the language "German"
 "Norsk nynorsk" is the LanguageNameInSelf for the language "Norwegian nynorsk"
 "British English" LanguageNameInSelf for the British version of "English"

Comments : An optional list of comments for the language used.

UniqueID : The global unique identifier for the language.

xtdLanguageRepresentation

An xtdLanguageRepresentation may be a name, a description, a value or a reference. It is always language dependent.

EXPRESS specification:

```
*){XE "xtdLanguage (REF)"}{XE "xtdGlobalUniqueID (REF)"}{XE "xtdLanguageRepresentation" \b}
```

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

Attribute definitions:

LanguageName : The name of the language used for the xtdLanguageRepresentation.

Preferred : An optional Boolean operator used to tag the preferred one among names, descriptions, values or references for an xtdLanguageRepresentation.

UniqueID : The global unique identifier for the language representation.

xtdMeasureWithUnit

The xtdMeasureWithUnit is a measure that can have a value component and a property dependent unit component.

EXAMPLE:

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

"AB88" is an xtdMeasureWithUnit with a value component "AB88" and no unit component.

EXPRESS specification:

```
*){XE "xtdUnit (REF)"}{XE "xtdValue (REF)"}{XE "xtdMeasureWithUnit" \b}
```

```
ENTITY xtdMeasureWithUnit
  SUBTYPE OF(xtdObject);
  UnitComponent           : OPTIONAL xtdUnit;
  ValueComponent         : OPTIONAL xtdValue;
END_ENTITY;
(*
```

Attribute definitions:

UnitComponent : The unit in which the value is expressed.

ValueComponent : The value for the measure for the given unit.

xtdName

The xtdName is the language dependent name of an object. An object may have many names in the

same language.

EXAMPLE:

"beam" and "truss" are both English names for the same object
"bjelke" is the Norwegian name for the English "beam"

EXPRESS specification:

*){XE "xtdLabel (REF)"}{XE "xtdName" \b}

```
ENTITY xtdName
  SUBTYPE OF(xtdLanguageRepresentation);
  Name : xtdLabel;
END_ENTITY;
(*
```

Attribute definitions:

Name : A label holding the name of the object.

xtdNest

xtdNest is a grouping mechanism only allowing things of the same type to be members.

EXAMPLE:

"Window properties" is an xtdNest group of only properties.

EXPRESS specification:

*){XE "xtdNest" \b}

```
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 are of the same type.

xtdObject

The xtdObject is the abstract concept holding all object classes in the model. The xtdObject can be one of xtdProperty, xtdMeasureWithUnit, xtdActor, xtdUnit, xtdSubject or xtdActivity.

EXPRESS specification:

*){XE "xtdObject" \b}

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

xtdProperty

xtdProperty is used to describe or quantify a concept.

EXAMPLE:

"Width" is a property.

"Door width" is a property used to quantify the "Width" of the subject door.

"Heat transfer" is a property.

"Colour" is a property.

"Duration" is a property.

"Comfort" is a property

EXPRESS specification:

```
*){XE "xtdProperty" \b}
```

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

xtdReference

The xtdDescription is a language dependent reference for the object. A reference is an external document or source.

EXAMPLE:

"ISO 31 Quantities and units" is the English reference for all ISO 31 units.

EXPRESS specification:

```
*){XE "xtdLabel (REF)"}{XE "xtdReference" \b}
```

```
ENTITY xtdReference
  SUBTYPE OF(xtdLanguageRepresentation);
  Identifier : xtdLabel;
  END_ENTITY;
(*
```

Attribute definitions:

Identifier : A label identifying the reference used.

xtdRelActsUpon

The xtdRelActsUpon defines a relationship that enables one object to act upon one or more other objects.

EXAMPLE:

"A column supports a beam"

"A bricklayer lays bricks"

EXPRESS specification:

*){XE "xtdRelActsUpon" \b}

```

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.

xtdRelAssignsCollections

xtdRelAssignsCollections is a relationship used to assign one or more collections of objects to another object.

EXPRESS specification:

*){XE "xtdObject (REF)"}{XE "xtdCollection (REF)"}{XE "xtdRelAssignsCollections" \b}

```

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

```

(*

Attribute definitions:

RelatingObject : The object to which the collection is assigned.

RelatedCollections : The collection of things that is assigned.

xtdRelAssignsMeasures

xtdRelAssignsMeasures is a relationship used to assign one or more measures with units to a property.

EXPRESS specification:

*){XE "xtdProperty (REF)"}{XE "xtdMeasureWithUnit (REF)"}{XE "xtdName (REF)"}{XE "xtdRelAssignsMeasures" \b}

```

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

```

(*

Attribute definitions:

Related : A property to which one or more measures with unit is assigned.

Relating : A set of measures with unit that is assigned to the property.

MethodOfInterpretation : Method of interpretation is the 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.

xtdRelAssignsProperties

xtdRelAssignsProperties is a relationship that enables one or more properties to be assigned to an object.

EXAMPLE:

"door height" is a "property" assigned to the "object" "door"

EXPRESS specification:

```
{XE "xtdProperty (REF)"}{XE "xtdObject (REF)"}{XE "xtdRelAssignsProperties" \b}
```

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

Attribute definitions:

RelatedProperty : A set of properties being properties of the relating object.

RelatingObject : The object to which the properties are assigned.

xtdRelAssociates

The xtdRelAssociates is a generic relationship linking a relating object to a set of related objects. The semantic of the relationship is given by the attribute FormOfAssociation.

EXPRESS specification:

```
{XE "xtdObject (REF)"}{XE "xtdObject (REF)"}{XE "xtdRelAssociates" \b}
```

```
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 : An object whose semantics are derived from the associated object(s)

i.e. the object that is the target of the relationship.

RelatedObjects : A set of related object(s) giving meaning to the associated object.

Formal Propositions:

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

xtdRelCollects

The xtdRelCollects relationship handles the assignment of group members to group objects. It allows for grouping arbitrary objects within a group. xtdCollection can be one of xtdNest or xtdBag.

EXPRESS specification:

`*){XE "xtdRoot (REF)"}{XE "xtdCollection (REF)"}{XE "xtdRelCollects" \b}`

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

Attribute definitions:

RelatedThings : One or more objects or relationships making up a collection.

RelatingCollection : A collection of objects, where each instance in the collection is an object or relationship

xtdRelComposes

The xtdRelComposes relationship defines the general concept of object being composed of other objects. The composition relationship can be applied in a recursive manner, i.e. a composed object can act as a part of another composed object.

EXAMPLE:

A "door leaf" is a "part" of a "whole" door.

EXPRESS specification:

`*){XE "xtdRelComposes" \b}`

```
ENTITY xtdRelComposes
  SUBTYPE OF(xtdRelAssociates);
WHERE
  WR1      : SIZEOF(QUERY(Result <* SELFxtdRelComposes.RelatedObjects |
    NOT(TYPEOF(SELFxtdRelComposes.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. xtdSubjects are composed only of other xtdSubjects.

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

xtdRelGroups

EXPRESS specification:

```
*){XE "xtdRelGroups" \b}
```

```
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.

xtdRelSequences

This objectified relationship handles the concatenation of processes or activities over time. xtdRelSequences is defined as the relationship between two activities. The related object is the successor of the relating object, being the predecessor.

EXPRESS specification:

```
*){XE "xtdActivity (REF)"}{XE "xtdActivity (REF)"}{XE "xtdRelSequences" \b}
```

```
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 activity that came before.

RelatedActivity : Reference to the activity that follows or takes place after.

Formal Propositions:

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

WR2 One of relating activity or related activity must be asserted.

xtdRelSpecializes

The xtdRelSpecializes is a relationship linking a subclass and a superclass indicating that the subclass has more constrained criteria than the superclass; and that the superclass is more generic than the subclass.

EXAMPLE:

"outer door" is a subclass of "door".
 "space" is the superclass of "room".

EXPRESS specification:

```
*){XE "xtdRelSpecializes" \b}

ENTITY xtdRelSpecializes
  SUBTYPE OF(xtdRelAssociates);
WHERE
  WR1      : SIZEOF(QUERY(Result <* RelatedObjects | RelatingObject:=: Result)) = 0;
  WR2      : SIZEOF(QUERY(Result <* SELFxtdRelSpecializes.RelatedObjects |
    NOT(TYPEOF(SELFxtdRelSpecializes.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. xtdSubjects can only be a specialization of another xtdSubject.

xtdRelationship

xtdRelationship is the abstract generalization of all objectified relationships in the model. This allows relationship specific properties to be kept directly at the relationship.

EXPRESS specification:

```
*){XE "xtdRelationshipRole (REF)"}{XE "xtdRelationshipRole (REF)"}{XE "xtdRelationship" \b}

ENTITY xtdRelationship
  ABSTRACT SUPERTYPE OF (ONEOF(xtdRelAssociates, xtdRelAssignsProperties,
    xtdRelAssignsMeasures, xtdRelCollects, xtdRelAssignsCollections,
    xtdRelSequences))
  SUBTYPE OF(xtdRoot);
  RelatingRole                    : OPTIONAL xtdRelationshipRole;
  RelatedRole                    : OPTIONAL xtdRelationshipRole;
END_ENTITY;
(*
```

Attribute definitions:

RelatingRole : The relationship role holding the names of the relating role in a

relationship.

RelatedRole : The relationship role holding the names of the related role in a relationship.

xtdRelationshipRole

An explanation for the roles of a relationship. This allows for naming of roles of relationships as well as relationships in the model.

EXAMPLE:

"is a part of" is the English name of the role "RelatedObjects" in the "xtdRelComposes" relationship.

EXPRESS specification:

```
*){XE "xtdName (REF)"}{XE "xtdRelationshipRole" \b}
```

```
ENTITY xtdRelationshipRole;
  RoleNames          : SET [1:?] OF xtdName;
END_ENTITY;
(*
```

Attribute definitions:

RoleNames : The language dependent role names for the roles of a relationship.

xtdRoot

The xtdRoot is the top level and most abstract concept in the model.

EXPRESS specification:

```
*){XE "xtdDate (REF)"}{XE "xtdVersionID (REF)"}{XE "xtdGlobalUniqueID (REF)"}{XE "xtdDescription (REF)"}{XE "xtdName (REF)"}{XE "xtdReference (REF)"}{XE "xtdRoot" \b}
```

```
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;
  References             : OPTIONAL SET [1:?] OF xtdReference;
END_ENTITY;
(*
```

Attribute definitions:

VersionDate : The date of the last revision of the reference data object.

VersionID : The version number for the reference data version of the object.

UniqueID : The global unique identifier for the object.

Descriptions : A set of language dependent descriptions of the object.

Names : A set of language dependent names of the object.

References : A set of language dependent references for the object.

xtdSubject

The xtdSubject is anything that can exist. An xtdSubject can be a physical thing or a logical thing.

EXAMPLE:

-"Roof" is an xtdSubject
-"Lobby" is an xtdSubject
-"Control system" is an xtdSubject
-"Road" is an xtdSubject
-"Airport" is an xtdSubject
-"Software" is an xtdSubject

EXPRESS specification:

*){XE "xtdSubject" \b}

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

xtdUnit

The xtdUnit is any type of unit assigned to a value. A unit can e.g. be a base unit from the SI system, derived unit or user defined units.

EXAMPLE:

"metre" is xtdUnit, which can be represented by the symbol "m"
"W/m.K" is an xtdUnit used with the property "thermal conductivity"

EXPRESS specification:

*){XE "xtdUnit" \b}

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

xtdValue

The xtdValue is the language dependent value of an object. An object may have many values in the same language.

EXAMPLE:

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

EXPRESS specification:

*){XE "xtdSimpleValue (REF)"}{XE "xtdValue" \b}

```
ENTITY xtdValue
  SUBTYPE OF(xtdLanguageRepresentation);
  ValueComponent : xtdSimpleValue;
END_ENTITY;
```

(*

Attribute definitions:

ValueComponent : A select type holding the value

*)

END_SCHEMA;

(*

Bibliography

- [1] ISO 15926-2 Industrial automation systems and integration – Integration of life-cycle data for oil and gas production facilities – Part 2: Data model
- [2] IAI/IFC International Alliance for Interoperability, Industry Foundation Classes, IFC 2x model.
- [3] ISO 10303-11 Industrial automation systems and integration – Product data representation and exchange – Part 11: Description methods: EXPRESS language reference manual.
- [4] ISO 10303-41 Industrial automation systems and integration – Product data representation and exchange – Part 41: Fundamentals of product description and support.
- [5] ISO/CD 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.
- [6] ISO/FDIS 12006-2 Building Construction – Organization of information about construction works – Part 2: Framework for classification of information.

Appendix

Conventions

The following conventions have been used in the development of this specification. These conventions do not apply to the population or use of the framework.

1 Names

1.1 Characters used for names

The characters [A-Z] upper case, [a-z] lower case, [0-9] numeric shall be used for names of schema, classes, property sets, relationships, enumerations, select types, defined data types, attributes, functions and rules.

1.2 Initial characters

The initial character of names of schema, classes, property sets, relationships, enumerations, select types, defined data types, attributes, functions and rules shall be an upper case character [A-Z].

1.3 Case of names

Names of schema, classes, property sets, relationships, enumerations, select types, defined data types, attributes, functions and rules shall be written in upper and lower case characters as a single name without spaces.

The first character of each word in normal usage shall be written as an upper case character. All other characters forming part of the same word in normal usage shall be written in lower case characters.

1.4 Lengths of names

The lengths of names for schema, classes, property sets, relationships, enumerations, select types, defined data types, attributes, functions and rules shall not be restricted.

1.5 xtd prefixes

Names of schema, classes, enumerations, select types, defined data types, functions and rules shall be prefixed by the term 'xtd' to identify their usage within the ISO PAS 12006-3 version 2 Information Model. The prefix 'xtd' shall be treated as a word in normal usage and the 'Case of names' rule applied to its use. 'Xtd' is an acronym for extensible taxonomy definition.

1.6 Names of classes

The name of a class shall be a noun or combination of nouns, denoting the 'content' or 'type' of the class.

1.7 Names of relationship classes

Classes acting as objectified relationships classes within the object model shall contain the term 'Rel' following the 'xtd' prefix and before the name of the class in normal usage. The inserted 'Rel' shall be treated as a word in normal usage and the 'Case of names' (4.1.3) rule applied. Relationships should be named starting from the related (SET [1:?]) of objects ending on the relating object. I.e. xtdRelComposes means that the relationship compose the parts (SET [1:?]) into a whole. The inverse relationship could be named xtdRelDecomposes.

The name of the objectified relationship class shall be a verb or verb phrase that denotes the 'function' of the objectified relationship class.

1.8 Names of attributes in relationship classes

Each objectified relationship class defines two major attributes, the relating (left side) and the related (right side) class, which are put into a relationship by virtue of the objectified relationship class. The following naming convention applies for both attributes:

- In case of a one-to-one relationship, the name shall be given in singular.
- In case of a one-to-many relationship, the name shall be given in plural.

1.9 Plural naming of aggregation

Names of attributes and relationships within an aggregation shall be expressed in the plural form. For instance, if a class could have one or many associated documents, the relationship would be expressed by Documents.

2 Single inheritance of subtypes

A subtype shall be a specialization of exactly one supertype.

3 Exclusion constraint in supertypes

A supertype shall be constrained so that it can be instantiated exclusively by one of its subtypes.

4 Many-to-many relationships

All many-to-many relationships shall be resolved to one-to-many or one-to-one relationships objectified through the introduction of a Relationship Class.