PUBLICLY AVAILABLE SPECIFICATION

ISO/PAS 12006-3

First edition
2001-10-15

Building construction — Organization of information about construction works —

Part 3: Framework for object-oriented information exchange

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

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

Reference number
ISO/PAS 12006-3:2001(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.ch
Web www.iso.ch

Printed 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 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*.

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

— *Part 2: Framework for classification of information*

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

ISO 12006-2 provides a framework for a classification, rather than the object-oriented approach of this part of ISO 12006. The two parts 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 works[1)].

While ISO 12006-2 reflects many years of refinement of classification systems, this part of ISO 12006 represents not so much new thinking, but a new implementation of established information modelling practice using a new ISO process which aims to bring new work of this kind into use as quickly as possible. Feedback is a vital component of the development process and will be welcomed, via the secretariat of ISO/TC 59/SC 13.

Annex A of this part of ISO 12006 is for information only.

1) ISO 12006-1 has been cancelled.

Introduction

Over the last few years much work has been undertaken in structuring technical information for construction works. In particular, two approaches have been pursued within ISO: “classification” and “object-orientation.”

The former has seen the development of a new International Standard, ISO 12006-2, which embraces many of the classification systems which have arisen since the first formal construction classification, SfB, was introduced in Sweden in 1950. The general approach taken by these is that they organize things by some characteristic or aspect, which might be described as “views” or “facets”.

The “object-oriented” approach describes the characteristics of things without a grouping preference or an ordering by specialization. In the object-oriented approach, the object is central, thus acting as a container of characteristics. Such an approach is exemplified in International Standards such as those in the ISO STEP series (ISO 10303), or EPISTLE (ISO 15926). It is also known as “product modelling”. An object can be grouped with the help of classification systems that take one or more characteristics of the object for the grouping.

A number of ISO technical committees (as well as a number of related bodies outside ISO) considered that these methodologies needed to be reconciled in order to bring together the overlapping interests of users of computer-aided drafting systems, specifiers and suppliers of reference material, and form a “content bridge” between new object-oriented software and older legacy systems, in particular databases of construction information.

In addition to these approaches, several new technologies such as XML (eXtensible Mark-up Language), and IFC (Industry Foundation Classes), which have special importance for exchange of information about construction works, are currently under active development and implementation.

At a meeting of representatives of ISO technical committees and other bodies with an interest in construction information content and exchange technology, held in Vancouver in June 1999, it was agreed that

— a common terminology about objects and their attributes would be beneficial for information exchange in the construction industry, and

— there should be an International Standard which describes a framework for such a terminology in the form of a formal information model.

ISO/TC 59/SC 13 formed a working group, WG 6, to develop the standard and it determined that the framework should provide for

— definition of concepts,

— definition of relations between concepts, and

— naming of concepts (with multilingual capacity).

Real-life objects can be grouped in classes, e.g. the four walls of a given room belong to the conceptual class “wall”. Relations, properties, etc., and groups of classes can be objects as well, e.g. the size of a wall, or the U-value of a wall. A model is a representation of a part of the real world, e.g. a model predicting the heat flow through a wall under given conditions. A meta-model is a model of the model, e.g. if grammar defines how a language should be used, then grammar is a meta-model for languages. A very formal and precisely defined language for information models is EXPRESS and its graphical representation EXPRESS-G. In this part of ISO 12006, EXPRESS and EXPRESS-G are used.

The model specified provides the ability to define concepts. Concepts are objects defined by properties. Objects and properties can have relationships and can be grouped. Objects, grouping and relationships are the basic entities of the model. The set of properties or groups 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. Values form the semantic content of a concept, thus providing, through several aggregation levels, its ultimate formal description.

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. Each object is always named in English (the default language) and 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 classification references.

The model has one Root class from which the following three classes inherit: Objects, Groups and the Relationships between them. The Root class provides the ability to assign any set of names, labels and descriptions, in any language, to its derived types, as well as identifiers and dates.

Objects are divided into Subjects, References, Activities, Units and Properties. Subjects are the things that are described. References provide the means to associate external documents, such as standards, regulations and classification systems, to the Objects. 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, Specialization, Composition, Involvement (acting upon), Property assignment, Grouping and Value assignment.

Groups provide for all kinds of groups of Objects, including nested Groups, by means of the Grouping Relationships.

Properties are classes for the storage of data. Data are Values with optionally associated Units, and are assigned to Properties by means of the Value assignment Relationship. This Relationship makes a distinction between several – enumerated – Value assignment roles, such as Nominal Values, and Upper or Lower boundary Values.

The meta-model described in this part of ISO 12006 can have numerous uses but its principal potential lies with the development of the precise “sets of words” needed to maximize the efficiency of product models and e-business for construction works.

A set of words can be called a terminology, a vocabulary or a lexicon and in this context these terms are more or less interchangeable. Sets of words have long been published as printed dictionaries but it is expected that the sets of words which this specification represent, will enable the publishing of electronic dictionaries.

Because this specification has a *lexical* function — it is to do with the words used in communication about buildings — the prefix “Lex” has been adopted within the model. For sets of words, “vocabulary” has been used rather than “terminology” as being the most commonly understood term.

Building construction — Organization of information about construction works —

Part 3: Framework for object-oriented information exchange

1 Scope

This part of ISO 12006 specifies a language-independent information model which may be used for the development of vocabularies used in 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 reference

The following normative document contains provisions which, through reference in this text, constitute provisions of this part of ISO 12006. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this part of ISO 12006 are encouraged to investigate the possibility of applying the most recent edition of the normative document indicated below. For undated references, the latest edition of the normative document referred to applies. Members of ISO and IEC maintain registers of currently valid International Standards.

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

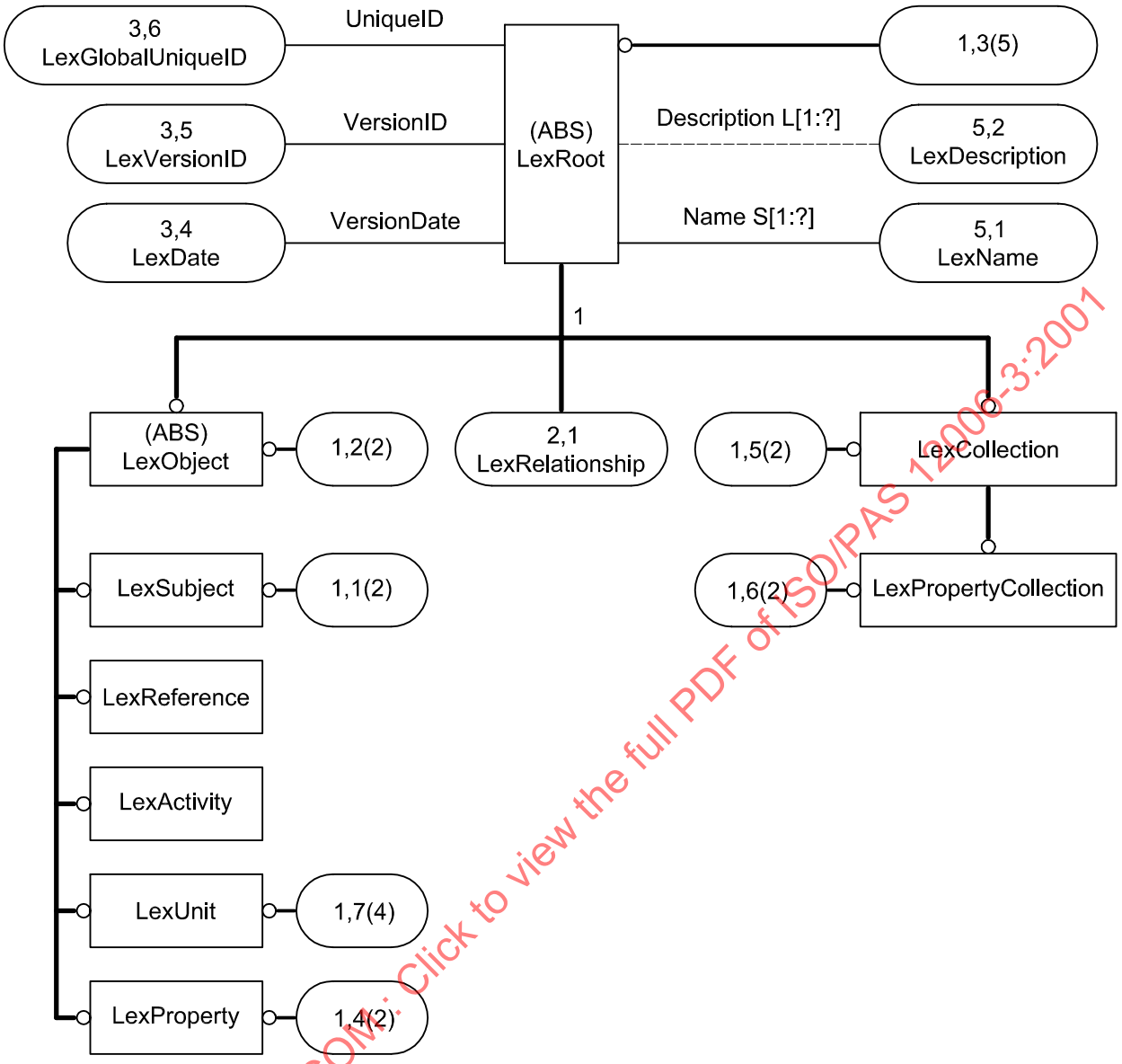
3 Terms and definitions

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

4 Specification

ISO/PAS 12006-3, version 1, is described in the EXPRESS-G diagrams shown in Figures 1 to 5. Following those, the formal definition is given in EXPRESS.

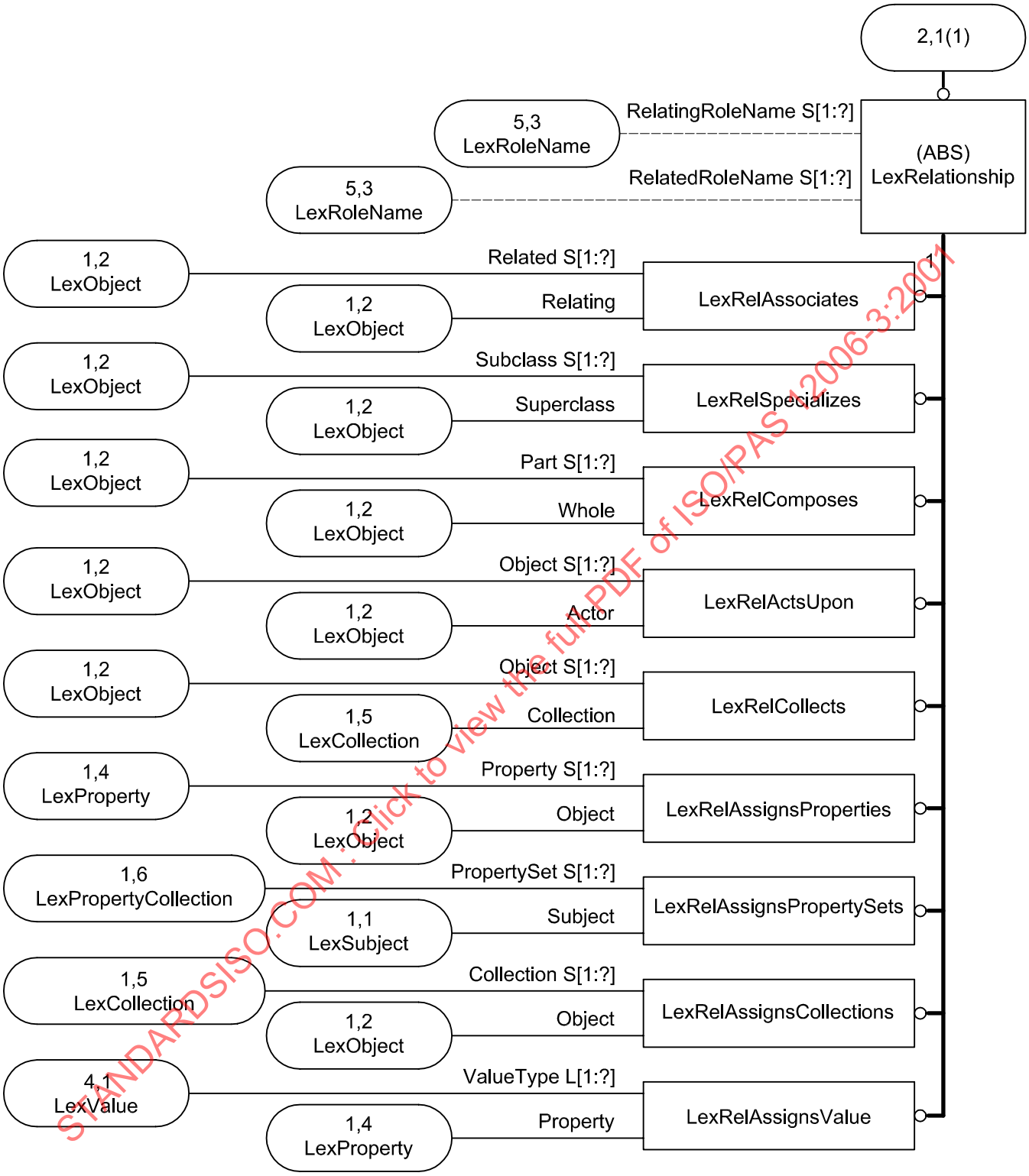
The conventions used in the development of this specification are given in annex A. These conventions do not apply to the population or use of the framework.



TOP LEVEL STRUCTURE

From the Root the model is divided into three main concepts: Objects, Collections and the Relationship between them. Names, descriptions and identifiers are inherited from the abstract Root object.

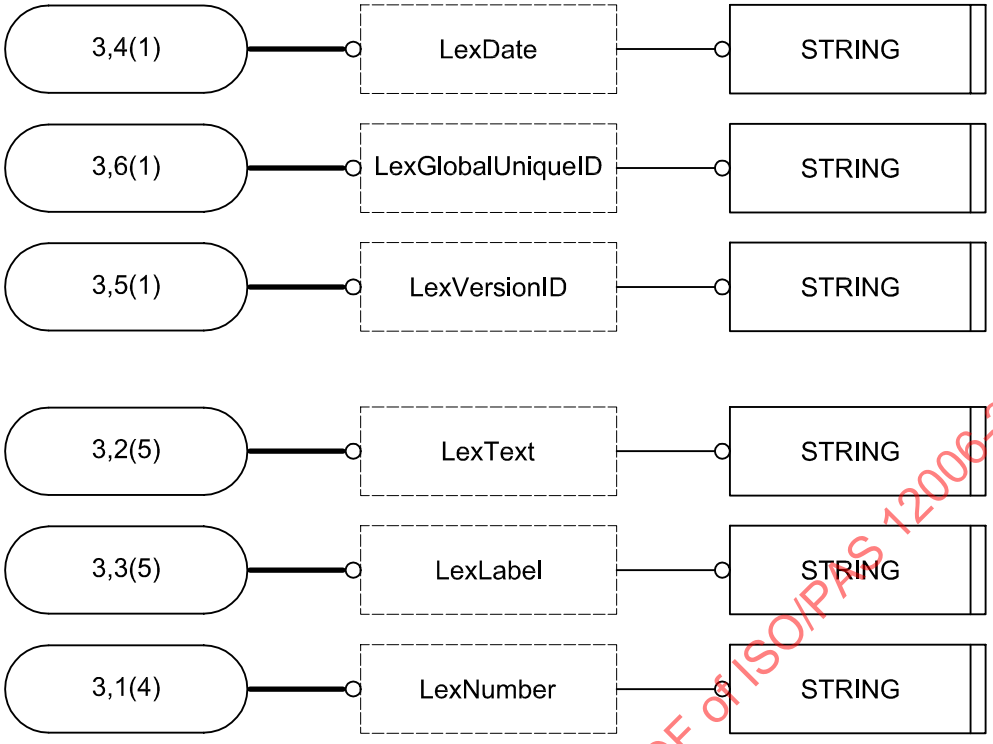
Figure 1 — EXPRESS-G diagram 1



RELATIONSHIPS

The models have both generic and specific relationships. The most generic relationship is LexRelAssociates which can be given any name and set of roles through the abstract LexRelationship.

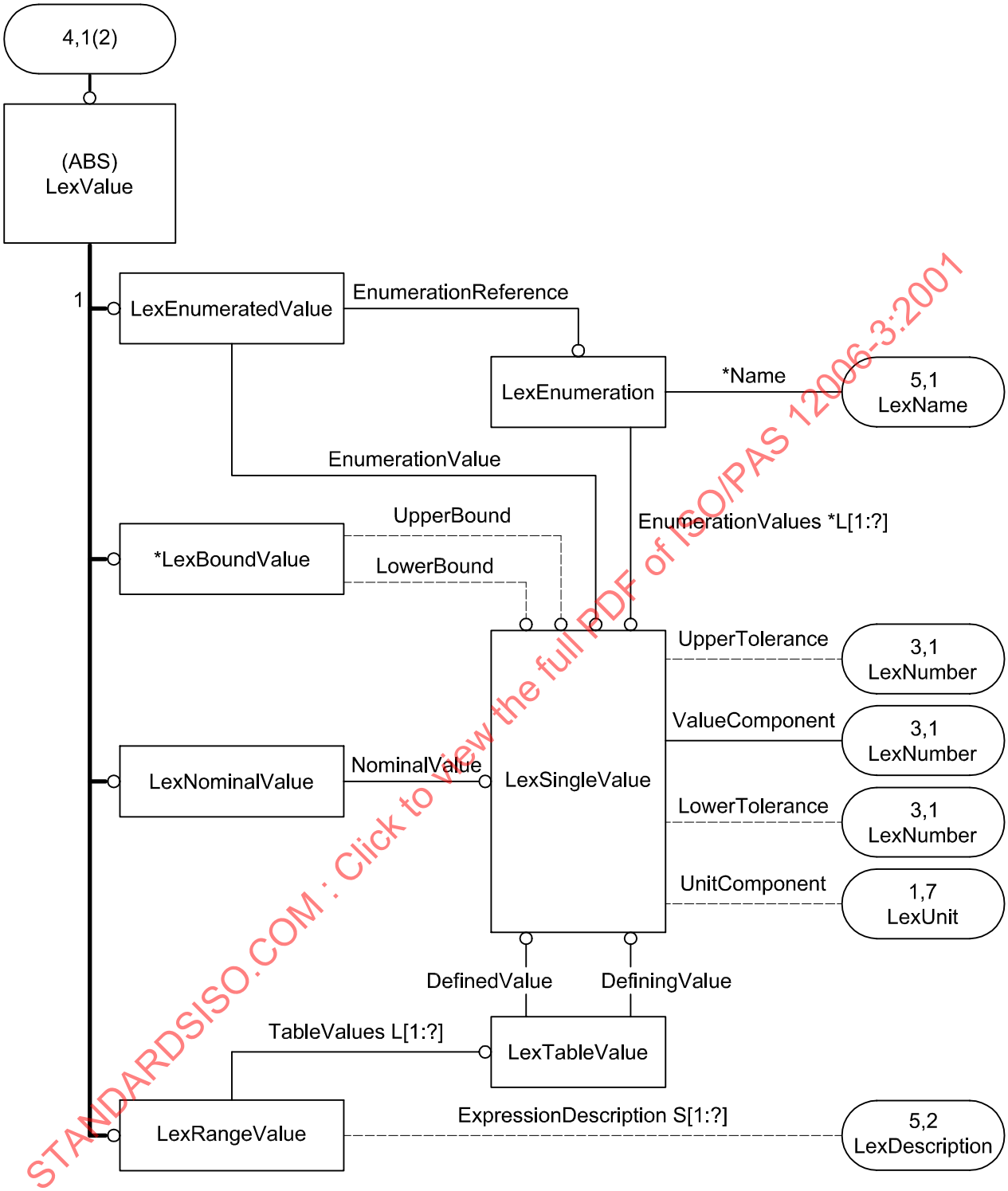
Figure 2 — EXPRESS-G diagram 2



SIMPLE TYPES

All text and numbers are assigned through the use of Defined types. All values are stored as Strings while the formatting lies in the definition of the Defined type.

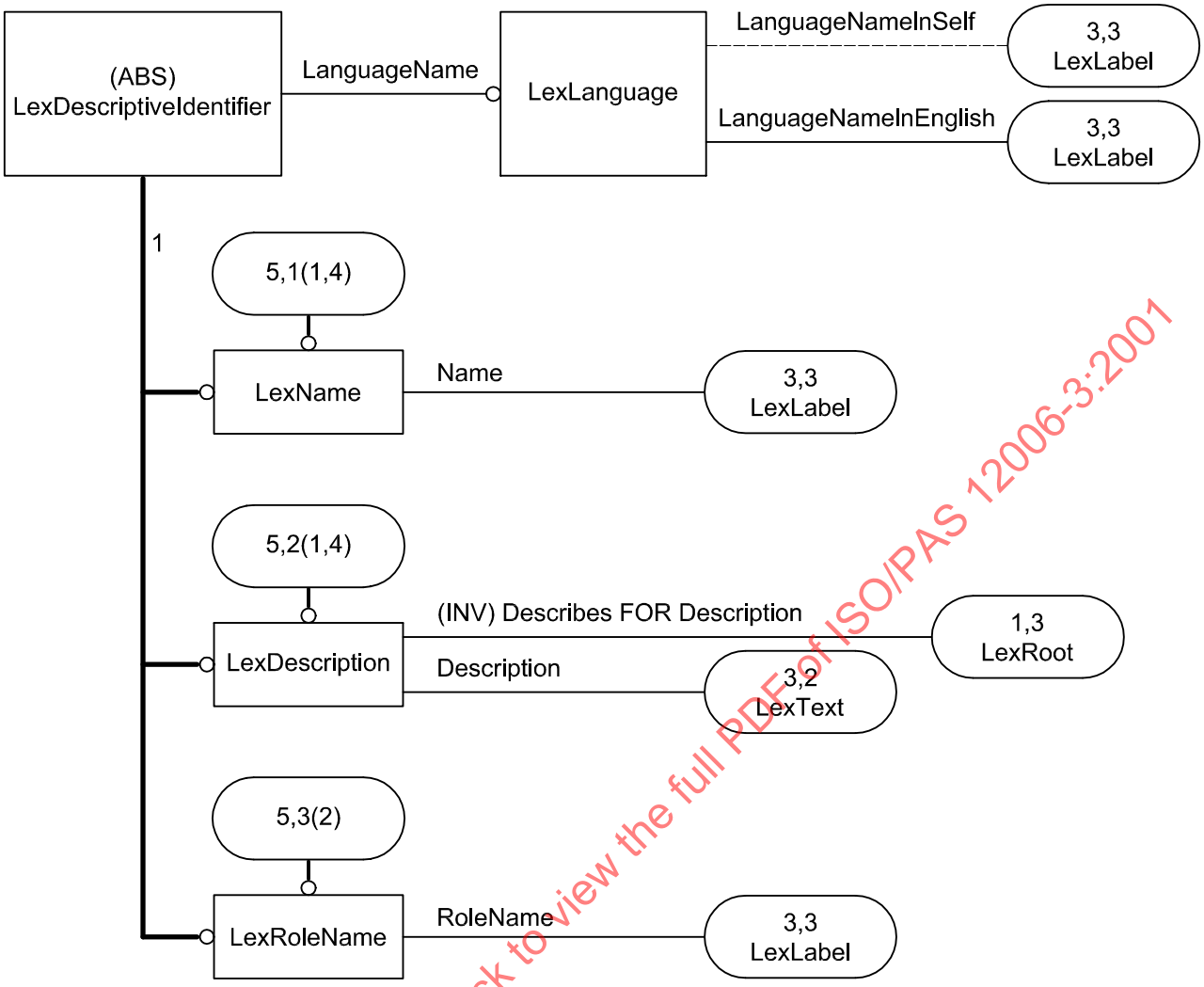
Figure 3 — EXPRESS-G diagram 3



VALUE TYPES

The mechanism to create all types of values and tolerances, which can be assigned to properties. LexValue is not a subtype of LexRoot.

Figure 4 — EXPRESS-G diagram 4



DESCRIPTIVE TYPES

Allows you to add any number of names and descriptions to any object. The names will always be of a particular language. The model also allow you to name roles of Relationships. LexDescriptiveIdentifier is not a subtype of LexRoot.

Figure 5 — EXPRESS-G diagram 5

EXPRESS specification:

*)

```
   SCHEMA ISO_PAS_12006_3_VERSION_1;
(*
```

LexActivity

The LexActivity 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" 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:

*)

```
   ENTITY LexActivity
     SUBTYPE OF(LexObject);
   END_ENTITY;
(*
```

LexBoundValue

A LexBoundValue is a value that may exist within a range that is bounded by upper and lower limits. At least one of the upper bound or lower bound must be asserted.

EXAMPLE:
The value range 13 kg to 120 kg is a LexBoundValue.

EXPRESS specification:

*)

```
   ENTITY LexBoundValue
     SUBTYPE OF(LexValue);
     UpperBound                :OPTIONAL LexSingleValue;
     LowerBound                :OPTIONAL LexSingleValue;
   WHERE
     WR1      :EXISTS(UpperBound) OR EXISTS(LowerBound) ;
   END_ENTITY;
(*
```

Attribute definitions:

UpperBound:	A value being the upper bound of the bounded value
LowerBound:	A value being the lower bound of the bounded value

LexCollection

The LexCollection is a grouping mechanism allowing collections containing mixtures of any subtype of LexObject. A LexCollection can be used to group objects for special purposes or professions.

EXAMPLE:
"steel work" is an example of a LexCollection of all objects, properties and units used in connection with steel work
"pumps" is a LexCollection grouping all types of pumps regardless of where they appear in different hierarchies
"ironmongery" is an example of a LexCollection of objects that have no functional relationship

EXPRESS specification:

*)

```
  ENTITY LexCollection
    SUBTYPE OF(LexRoot);
  END_ENTITY;
(*
```

LexDate

LexDate 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 10303-41.

EXPRESS specification:

*)

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

LexDescription

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

EXAMPLE:
"A door panel is normally a door leaf that opens to allow people or goods to pass" is the English explanation for the object "Door panel".

EXPRESS specification:

*)

```
  ENTITY LexDescription
    SUBTYPE OF(LexDescriptiveIdentifier);
    Description               :LexText;
  INVERSE
    Describes                 :LexRoot FOR Description;
  END_ENTITY;
(*
```

Attribute definitions:

Description:	A text being the description of the object
Describes:	The object to which the description applies

LexDescriptiveIdentifier

A LexDescriptiveIdentifier may be a name, a description or a role name. It is language dependent.

EXPRESS specification:

*)

```
ENTITY LexDescriptiveIdentifier
  ABSTRACT SUPERTYPE OF (ONEOF(LexName, LexDescription, LexRoleName));
  LanguageName                :LexLanguage;
END_ENTITY;
(*
```

Attribute definitions:

LanguageName:	The name of the language used for the LexDescriptiveIdentifier

LexEnumeratedValue

LexEnumeratedValue is a reference to a value selected from a defined list of options that are stored in an enumerated list of LexSingleValues.

NOTE See LexEnumeration for the collection containing the enumerated values.

EXPRESS specification:

*)

```
ENTITY LexEnumeratedValue
  SUBTYPE OF(LexValue);
  EnumerationReference        :LexEnumeration;
  EnumerationValue            :LexSingleValue;
END_ENTITY;
(*
```

Attribute definitions:

EnumerationReference:	Enumeration from which a value has been selected by an index.
EnumerationValue:	Enumeration value, which is listed in the referenced LexEnumeration.

LexEnumeration

LexEnumeration is an ordered list of values from which a particular value may be selected.

EXPRESS specification:

*)

```
  ENTITY LexEnumeration;
    EnumerationValues          :LIST [1:?] OF UNIQUE LexSingleValue;
    Name                       :LexName;
  UNIQUE
    UR1      :Name;
  END_ENTITY;
(*
```

Attribute definitions:

EnumerationValues:	List of values that form the enumeration
Name:	A set of language-dependent names for the enumeration

LexGlobalUniqueID

The LexGlobalUniqueID 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 character 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:

*)

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

LexLabel

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

EXPRESS specification:

*)

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

LexLanguage

LexLanguage is the language used for names, role names and descriptions.

EXPRESS specification:

*)

```
   ENTITY LexLanguage;
     LanguageNameInEnglish      :LexLabel;
     LanguageNameInSelf         :OPTIONAL LexLabel;
   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

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"

LexName

The LexName 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:

*)

```
   ENTITY LexName
     SUBTYPE OF(LexDescriptiveIdentifier);
     Name                       :LexLabel;
   END_ENTITY;
(*
```

Attribute definitions:

Name: A label holding the name of the object

LexNominalValue

LexNominalValue is a idealized representation of a value between tolerances.

EXPRESS specification:

*)

```
   ENTITY LexNominalValue
     SUBTYPE OF(LexValue);
     NominalValue               :LexSingleValue;
   END_ENTITY;
(*
```

Attribute definitions:

NominalValue:	A nominal value is the idealized value that is normally quoted

LexNumber

The LexNumber is a UNICODE string that can hold any number, or character.

EXPRESS specification:

```
*)

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

LexObject

The LexObject is the abstract concept holding all object classes in the model. The LexObject can be one of LexSubject, LexReference, LexActivity, LexUnit or LexProperty.

EXPRESS specification:

```
*)

   ENTITY LexObject
     ABSTRACT SUPERTYPE
     SUBTYPE OF(LexRoot);
   END_ENTITY;
(*
```

LexProperty

LexProperty is used to describe or quantify a concept. A property can be a property of a subject, reference or activity.

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:

```
*)

   ENTITY LexProperty
     SUBTYPE OF(LexObject);
   END_ENTITY;
(*
```

LexPropertyCollection

The LexPropertyCollection is a grouping mechanism allowing collections of properties. A LexPropertyCollection will typically be used to group properties covering different aspects of an object.

EXAMPLE:
"concrete strength" is a collection of properties used to describe the strength of concrete
"fire barrier" is a collection of properties that provide the possibility for the object to act as a fire barrier

EXPRESS specification:

*)

```
ENTITY LexPropertyCollection
  SUBTYPE OF(LexCollection);
END_ENTITY;
```
(*

LexRangeValue

The LexRangeValue is a two-dimensional array of values in which the value of each member of the first dimension of the array is dependent on the value of the corresponding member of the second dimension of the array. The range consists of table values, and is described by the language-dependent ExpressionDescription.

EXPRESS specification:

*)

```
ENTITY LexRangeValue
  SUBTYPE OF(LexValue);
  TableValues                :LIST [1:?] OF LexTableValue;
  ExpressionDescription      :OPTIONAL SET [1:?] OF LexDescription;
END_ENTITY;
```
(*

Attribute definitions:

TableValues: The list of values for a LexRangeValue expressed as a table in which for each defining value of 'y', the derived value of 'x' is provided

ExpressionDescription: A set of language-dependent descriptions for the range value

EXAMPLE:
ExpressionDescription can be the mathematical function from which the LexValueRange is determined. The description will normally be expected to identify the function specifically in the form y = f(x).

LexReference

The LexReference is the link to classification systems, standards or other written information about a concept. The class can be used to store types of documents and their inner structure.

EXAMPLE:
"ISO 31" is a LexReference and a subtype of LexReference "ISO Standard"
"2.3.2.1 Base units" is a LexReference and a part_of the LexReference "ISO 31"

EXPRESS specification:

*)

```
ENTITY LexReference
  SUBTYPE OF(LexObject);
END_ENTITY;
```
(*

LexRelationship

LexRelationship 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:

*)

```
  ENTITY LexRelationship
    ABSTRACT SUPERTYPE OF (ONEOF(LexRelSpecializes, LexRelComposes,
                      LexRelAssociates, LexRelActsUpon, LexRelAssignsProperties,
                      LexRelCollects, LexRelAssignValues,
                      LexRelAssignsCollections, LexRelAssignsPropertySets))
    SUBTYPE OF(LexRoot);
    RelatingRoleName           :OPTIONAL SET [1:?] OF LexRoleName;
    RelatedRoleName            :OPTIONAL SET [1:?] OF LexRoleName;
  END_ENTITY;
(*
```

Attribute definitions:

RelatingRoleName:	A set of language-dependent names for the relating objects
RelatedRoleName:	A set of language-dependent names for the related object

LexRelActsUpon

The LexRelActsUpon 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:

*)

```
  ENTITY LexRelActsUpon
    SUBTYPE OF(LexRelationship);
    Actor                      :LexObject;
    Object                     :SET [1:?] OF LexObject;
  END_ENTITY;
(*
```

Attribute definitions:

Actor:	The actor performing the action
Object:	The objects being acted upon

LexRelAssignsCollections

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

EXPRESS specification:

*)

```
   ENTITY LexRelAssignsCollections
     SUBTYPE OF(LexRelationship);
     Object                    :LexObject;
     Collection                :SET [1:?] OF LexCollection;
   END_ENTITY;
(*
```

Attribute definitions:

Object:	The object to which the collection is assigned
Collection:	The collection of LexObjects that is assigned

LexRelAssignsProperties

LexRelAssignsProperties 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:

*)

```
   ENTITY LexRelAssignsProperties
     SUBTYPE OF(LexRelationship);
     Property                  :SET [1:?] OF LexProperty;
     Object                    :LexObject;
   END_ENTITY;
(*
```

Attribute definitions:

Property:	A set of properties being properties of the relating object
Object:	The object to which the properties are assigned

LexRelAssignsPropertySets

LexRelAssignsPropertySets is a relationship that enables a collection of properties to be assigned to an object.

EXAMPLE:
"thermal resistance" and "thermal reflectivity" are properties that quantify the performance of a subject as a thermal barrier

EXPRESS specification:

*)

```
   ENTITY LexRelAssignsPropertySets
     SUBTYPE OF(LexRelationship);
     Subject                   :LexSubject;
     PropertySet               :SET [1:?] OF LexPropertyCollection;
   END_ENTITY;
(*
```

Attribute definitions:

Subject:	A subject to which one or many property sets are assigned
PropertySet:	A property set that is assigned to the object

LexRelAssignValues

LexRelAssignValues is a relationship used to assign one or more values to a property.

EXPRESS specification:

*)

```
  ENTITY LexRelAssignValues
    SUBTYPE OF(LexRelationship);
    Property                  :LexProperty;
    ValueType                 :LIST [1:?] OF LexValue;
  END_ENTITY;
(*
```

Attribute definitions:

Property:	A property to which one or more values is assigned
ValueType:	A set of value types that is assigned to the property

LexRelAssociates

The LexRelAssociates is a generic association, which can be used to model any kind of relationship between objects. The association has two "roles"; related and relating. Each role can be named to give the role its semantic.

EXAMPLE:
"BelongsTo" is a relationship with i.e. "Owner" as role name of the relating object and "Item" as the role name of the related objects

EXPRESS specification:

*)

```
  ENTITY LexRelAssociates
    SUBTYPE OF(LexRelationship);
    Relating                  :LexObject;
    Related                   :SET [1:?] OF LexObject;
  END_ENTITY;
(*
```

Attribute definitions:

Relating:	An object whose semantics are derived from the associated object(s), i.e. the object that is the target of the relationship
Related:	A set of related object(s) giving semantic to the associated object

LexRelCollects

The LexRelCollects relationship handles the assignment of group members to group objects. It allows for grouping arbitrary objects within a group.

EXAMPLE:
"gas tank" is an "object" in the "inflammable equipment" collection

EXPRESS specification:

*)

```
   ENTITY LexRelCollects
     SUBTYPE OF(LexRelationship);
     Collection                :LexCollection;
     Object                    :SET [1:?] OF LexObject;
   END_ENTITY;
(*
```

Attribute definitions:

Collection:	A collection of objects, where each instance in the collection is a subtype of LexObject
Object:	One or more objects making up a collection

LexRelComposes

The LexRelComposes 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:
"door leaf" is a "part" of a "whole" door

EXPRESS specification:

*)

```
   ENTITY LexRelComposes
     SUBTYPE OF(LexRelationship);
     Part                      :SET [1:?] OF LexObject;
     Whole                     :LexObject;
   END_ENTITY;
(*
```

Attribute definitions:

Part:	A set of objects being the parts of the whole object
Whole:	The whole object being composed of parts

LexRelSpecializes

The LexRelSpecializes 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:

*)

```
  ENTITY LexRelSpecializes
    SUBTYPE OF(LexRelationship);
    Superclass                :LexObject;
    Subclass                  :SET [1:?] OF LexObject;
  END_ENTITY;
(*
```

Attribute definitions:

Superclass:	The parent object in a specialization relationship
Subclass:	Child objects in a specialization relationship

LexRoleName

The LexRoleName is a language-dependent name of the role for each end of a relationship.

EXAMPLE:
"part" and "whole" are English role-names assigned to the relationship "LexRelComposes"

EXPRESS specification:

*)

```
  ENTITY LexRoleName
    SUBTYPE OF(LexDescriptiveIdentifier);
    RoleName                  :LexLabel;
  END_ENTITY;
(*
```

Attribute definitions:

RoleName:	A label that is the name of a role in a relationship

LexRoot

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

EXPRESS specification:

*)

```
  ENTITY LexRoot
    ABSTRACT SUPERTYPE OF (ONEOF(LexObject, LexRelationship, LexCollection));
    VersionDate               :LexDate;
    VersionID                 :LexVersionID;
    UniqueID                  :LexGlobalUniqueID;
    Description               :OPTIONAL LIST [1:?] OF LexDescription;
    Name                      :SET [1:?] OF LexName;
  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
Description:	A set of language-dependent descriptions of the object
Name:	A set of language-dependent names of the object

LexSingleValue

The LexSingleValue is a value that can have upper and lower tolerances, a value component and a property-dependent unit component.

EXAMPLE:
"1m" is a LexSingleValue with a unit component "m" and a value component "1"
"AB88" is a LexSingleValue with a value component "AB88" and no unit component

EXPRESS specification:

*)

```
ENTITY LexSingleValue;
  ValueComponent              :LexNumber;
  LowerTolerance              :OPTIONAL LexNumber;
  UpperTolerance              :OPTIONAL LexNumber;
  UnitComponent               :OPTIONAL LexUnit;
END_ENTITY;
```

(*

Attribute definitions:

ValueComponent:	A number being the value of a property
LowerTolerance:	The lower numeric value of tolerance by which a value may be allowed to vary from the nominal value and still be considered as having the nominal value
UpperTolerance:	The upper numeric value of tolerance by which a value may be allowed to vary from the nominal value and still be considered as having the nominal value
UnitComponent:	The unit in which the value is expressed

LexSubject

The LexSubject is anything that can exist. A LexSubject can be a physical thing or a logical thing.

EXAMPLE:
— "Roof" is a LexSubject
— "Lobby" is a LexSubject
— "Control system" is a LexSubject
— "Road" is a LexSubject
— "Airport" is a LexSubject
— "Software" is a LexSubject