INTERNATIONAL STANDARD

ISO 16354

First edition
2013-03-15

Guidelines for knowledge libraries and object libraries

Lignes directrices pour les bibliothèques de connaissance et les bibliothèques d'objets

Reference number
ISO 16354:2013(E)

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

Published in Switzerland

Contents

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 electro technical 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 document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO 16354 was prepared by Technical Committee ISO/TC 59, *Buildings and civil engineering works*, Subcommittee SC 13, *Organization of information about construction works*.

Introduction

Knowledge libraries are databases that contain modelled knowledge about kinds of things.

Knowledge libraries are intended to support business processes concerning any kind of products during their lifetime, for example to support their design, procurement, construction, operation or maintenance. There is an increasing awareness of the high potential value of knowledge libraries and of the drawbacks of the inconsistencies and lack of interoperability between different knowledge libraries.

This standard is based on Netherlands Technical Agreement NTA 8611:2008 (en), *Guidelines for Knowledge Libraries and Object Libraries, Version 3.0*.

On both a national and international level knowledge libraries exist or are being developed, such as the Gellish English Dictionary-Taxonomy (previously called STEPlib), UNETO-VNI ETIM system, LexiCon and the GWW Objectenbibliotheek [Civil Object Library] and International Framework for Dictionaries (IFD) developed by the Building Smart consortium. International efforts include IEC 61360, ISO 13584, ISO/TS 15926-4, and ISO 12006-3.

Historically, most libraries have had their own unique structure and methodology for defining their objects and they use their own naming conventions. For instance, the structure of the article classes laid down in ISO 13584-42 notably differs from that of the UNETO-VNI component classes (publication 8) or LexiCon, based on ISO 12006-3. In most cases the intrinsic definition of objects will also be different.

The major ICT developments with regard to the Internet and XML technology have increased the possibility for uniformity. From a technical point of view, it has become much easier to exchange data, which increases the need and support for this within the industry. Organizations launching new initiatives for the creation of knowledge libraries may also greatly benefit from enhanced uniformity. They may come up with questions such as: “Which existing libraries should be used?”, “Will these libraries receive sufficient support?”, “Do they fulfil my information needs?” and “Is there international support for such libraries?”

INTERNATIONAL STANDARD **ISO 16354:2013(E)**

Guidelines for knowledge libraries and object libraries

1 Scope

The aim of this standard is to distinguish categories of knowledge libraries and to lay the foundation for uniform structures and content of such knowledge libraries and for commonality in their usage. By drawing up a number of guidelines, a guiding principle is provided for new libraries as well as for upgrading existing libraries. Without these guidelines there is an undesirable amount of freedom, so that the various libraries may become too heterogeneous. This would render the comparison, linking and integrated usage of these libraries very complex, if not impossible.

— The objective of the standard is to categorize knowledge libraries and object libraries and to provide recommendations for the creation of such libraries. Libraries that are compliant with the guidelines of this standard may be more easily linked to, or integrated with other libraries.

— The target audience of the standard consists of developers of knowledge libraries, builders of translation software or interfaces between knowledge libraries, certifying bodies and builders of applications who must base their work on the knowledge libraries laid down.

NOTE 1 Knowledge libraries are databases or files that contain modelled knowledge about kinds of things. They are intended to support business processes concerning any kind of products during their lifetime, for example to support their design, procurement, construction, operation or maintenance. There is an increasing awareness of the high potential value of knowledge libraries and of the drawbacks of the inconsistencies and lack of interoperability between different knowledge libraries.

NOTE 2 This standard does not aim to standardize terminology, but to harmonize and standardize concepts. Thus the use of synonyms and synonymous phrases and one-to-one translations are allowed or even recommended, provided that alternative terms denote the same concepts and reference is made to the corresponding synonymous terms in this standard.

2 Normative references

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

None.

3 Terms and definitions

3.1 Terms and definitions for concepts

For the purposes of this document, the following terms and definitions of concepts apply.

NOTE The guidelines in this standard are expressed by using two kinds of building blocks: concepts and relation types (which are a special kind of concepts). In this clause, the definitions of those building blocks are provided. The overall description method of the building blocks bears great resemblance to the description method used in several other ISO standards, e.g. the ISO 10303 series.

For each concept a number, a term (which may be a multi-word term or phrase) and a definition is given, usually followed by an explanation in a Note and one or more examples. Each term denotes a concept in English in the context (language community) of this standard.

3.1.1
knowledge library
collection of information models that express knowledge (which may include also definition models and requirements models) about kinds of things (concepts) and that are stored and retrieved as electronic information

Note 1 to entry: A knowledge library may contain knowledge about physical objects as well as about non-physical objects, such as occurrences, activities, processes and events, or about properties, relationships, scales (units of measure), mathematical objects, etc. Each information model in a knowledge library should be retrievable as a separate model, although the content of the various models may overlap. It is not required that every information model has a separate unique identifier as a model may also be retrieved on the basis of a query.

An *object library* (in the context of this standard) is a special kind of knowledge library as it is a collection of knowledge models (possibly also including definitions and requirements) about kinds of physical objects.

3.1.2
knowledge model
information model that expresses knowledge in a computer interpretable structure

Note 1 to entry: A knowledge model consists of a number of expressions of facts about a concept, each of which expressions expresses something that can be the case. Those expressions should comply with the guidelines in this standard. A requirements model is a subtype of a knowledge model. It expresses what shall be the case in a particular context.

Note 2 to entry: Knowledge models typically define further subtypes of the concepts that are defined in this standard.

Information models are expressions of meaning in a formal language that is computer interpretable.

3.1.3
fact
state of being the case

Note 1 to entry: A fact can be represented by a fact identifier (see the concept 'unique identifier'). Something that is the case may be expressed by an expression. Such an expression may consist of a relation between representatives of related things, whereas typically that relation is classified by a kind of relation. A fact may be stated, denied or questioned or confirmed in the expression.

3.1.4
definition
representation of a concept by a descriptive statement which serves to differentiate it from related concepts

[ISO 1087-1:2000]

Note 1 to entry: A definition may be expressed as natural language text (a textual definition) or as a definition model. A textual definition should comply with the applicable guideline(s) in this standard. A definition expresses what is by definition the case for all things of the defined kind and if such an expressed constraint is not the case for a thing, then that thing is not a thing of the defined kind.

A *definition model* is a subtype of a knowledge model and consists of a number of expressions of facts about the defined concept. Those expressions should comply with the guidelines in this standard.

3.1.5
concept
(1)unit of knowledge created by a unique combination of aspects and/or components

[adapted from ISO 1087-1:2000]

(2)commonality between individual things that is defined by one or more constraints that describe the limits for the inclusion of individual things to conform to the concept

Note 1 to entry: A concept is a human idea used to categorize phenomena and to allocate knowledge that is common about those phenomena. All concepts in a knowledge library are specializations (subtypes) of concept.

A concept can be defined or described or it can be used to define or describe other concepts.

The above two definitions define the same concept from different perspectives.

This standard makes a distinction between a concept (itself) and the definition model (knowledge model) that defines the concept. The term 'unit of knowledge' should therefore be interpreted as the concept itself. In this standard characteristics are aspects that are distinguished from components.

EXAMPLE Concepts with names such as road, building, bicycle, repairing, length, organization, centimetre.

3.1.6
physical object

individual thing that has a physical nature with a limited lifespan; it may be materialized (and then may be observable and touchable) or it may be imagined (having deemed aspects, as-if observable)

Note 1 to entry: Physical objects (i.e. the concept 'physical object') is a core kind of object (or object type) with which this standard is concerned (see 5.2). The guidelines that have been included in this standard are therefore concerned mainly with knowledge libraries geared towards the description of physical matters. A physical object is to be distinguished from the stuff, such as steel, that specifies the material of construction aspect of a physical object. Physical objects may be solid or liquid or gaseous, but also electronic or electromagnetic, such as software or radiation.

EXAMPLE Subtypes of physical object are concepts such as ones that have the following names: bridge, switch, ventilator, pump, chair, ship, airplane, nut and bolt as well as liquid stream, application software, data file, document and beam of light. Examples of (individual) physical objects (exemplars) are the Eiffel Tower, Paris, V-6060 (a particular real vessel), D-101 (a particular copy of a document).

3.1.7
organization

social entity which is a physical object that consists of people in a structure that is controlled to meet some purpose

Note 1 to entry: People can be defined as living physical objects, although this does not exclude that they might have non-physical characteristics. An organization, defined as an arrangement of people, is therefore also defined as a subtype of physical object, so that all guidelines for physical objects are also applicable for organizations. An organization can possess or use material, such as equipment, machines, and buildings. The materials possessed by an organization are owned by, but not defined as parts of the organization.

EXAMPLE Subtypes of physical object are kinds of organizations such as department, project team, contractor. Examples of individual organizations that are classified by such a kind are: the United Nations, Microsoft, department X.

3.1.8
occurrence

state that is dynamic and is an interaction over time between involved things, each with its own role

Note 1 to entry: May also be called 'happening'. An occurrence can be an activity that is performed by a person or a process or an event. Essential is that an occurrence takes time. The involved objects are in begin conditions (state) at the start of the occurrence and are in end conditions (state) at the termination of the occurrence. The begin conditions and end conditions may be the same, but are usually different. An occurrence can also be called a state transition. An event usually has a very short duration. For some processes it seems as if nothing happens, apart from at their atomic scale.

EXAMPLE Subtypes of occurrence are: inspecting, pumping, flowing, fabricating, measuring, maintaining, control, project, earthquake, etc.

3.1.9
term

(1)verbal designation of a concept or individual thing in a specific subject field

[adapted from ISO 1087-1:2000]

(2)role of a physical character string or sound that may include spaces and silences, respectively, that is used to designate a concept (e.g. a kind of physical object or aspect) or to designate an individual thing in a particular language (coding system) and language community

Note 1 to entry: A spoken sound or written character string is an utterance or written or printed physical object that is used in a certain context (language community) as a linguistic designation of a concept or of an individual thing. In this standard such a concept or individual thing is assumed to be represented by a unique identifier. A term usually consists of a particular sequence of characters or sound, one or more words, abbreviations or symbols, possibly separated by spaces or silences.

A *name* is a term that is neither a code nor a symbol or abbreviation. A name is a role of a character string in a naming relation, which denotes how something is called in a particular language and language community.

Note 2 to entry: A *character string* is a physical object; it is a sequence of characters of a standardized shape, typically ink on paper.

A term does not necessarily uniquely denote a particular concept, thus homonyms are not excluded. A term only uniquely denotes a particular concept in a particular language and language community.

The above two definitions are two expressions that intend to define the same concept.

EXAMPLE Terms such as road, room, bicycle, length, price, centrifugal pump, inspection, and kg that are used in English to refer to concepts; and terms such as Eiffel Tower and New York that are used to refer to individual things.

3.1.10
language

coding system of spoken and/or written words and sentences (expressions and phrases) that is used to communicate between people or systems

Note 1 to entry: There are natural languages and artificial languages. The latter may be a formal language (being explicitly defined and computer interpretable).

EXAMPLE English, German, and Mandarin are examples of natural languages. Gellish Formal English is an example of a formal artificial language, although its vocabulary consists of normal English terms.

3.1.11
language community

community that shares terminology (terms, names, abbreviations and codes) to unambiguously refer to concepts and to individual things

Note 1 to entry: A language community, also called a speech community, does not use homonyms within their shared community vocabulary, but may include synonyms. Terms from different language communities may include homonyms.

EXAMPLE Civil engineering, finance, and control engineering are examples of language communities.

3.1.12
unique identifier
UID

role of a character string when used for unambiguous reference to a concept or to an individual thing (e.g. a physical object or an aspect or a fact or a relation type) and that is unique within a particular common context, preferably in a universal context

Note 1 to entry: The function of a unique identifier is that it is a unique, language independent, reference to a concept, relation or individual thing. Ranges or conventions for unique identifiers shall be agreed between parties to avoid overlap with other parties when data exchange or data integration is intended.

A UID of a concept is different from UIDs for particular quantities of information about that concept. A UID for a concept refers to the concept itself. Thus a UID for a pump is different from a UID for the information (collection of facts) that is expressed by an entity with its attributes in a database.

A UID for a fact (a relation) is independent of the expression (the way a fact is expressed).

The inverse expression of a relation denotes the same fact (and thus shall be indicated by the same UID).

EXAMPLE The UID 130206 in Gellish Formal English refers to the concept 'pump'. The UID 570039 refers to the concept 'kg' in the same context. In such a formal language, UIDs or ranges for unique identifiers are issued on request for individual things as well as for proprietary extensions of the concepts that are defined in the language.

IFC and IFD and others use an algorithm for generating Globally Unique Identifiers. The algorithm guarantees the uniqueness of the UID, independent of the application by which it is generated. However, every system may generate its own GUID for the same concept.

3.1.13
aspect

concept by which the existence and appearance of a thing is experienced and that cannot exist without the existence of its possessor and which is either an intrinsic non-separable facet of its possessor or a role of its possessor

Note 1 to entry: Aspects are the phenomena by which people experience the existence and appearance or value of things. Subtypes of aspects are: characteristic, with further subtypes: physical property and quality, such as material of construction (stuff), but also economic value, risk or social importance. Physical properties are quantifiable, whereas qualities are non-quantifiable. The nature of such a phenomenon is called a *conceptual aspect*. Its extent, intensity or size is called a *qualitative aspect*, also called an *aspect value* or *property value*.

The concept 'role' is an extrinsic subtype of aspect.

EXAMPLE Subtypes of aspect are: kinds of physical properties, such as the concept's shape, length and colour, and kinds of qualities (which are usually not quantified), such as flammability and corrosivity. Such kinds are called conceptual aspects. Generic values for those conceptual aspects are called qualitative aspects. Examples are: 'cylindrical shape', 'stainless steel', the length '3 m', the colour 'red' and the qualities 'flammable', 'inflammable' and 'corrosive'. Also numbers and ranges, such as 0, 1, 1.5, 1/8, '3 to 5' and ' > 10' are qualitative aspects.

3.1.14
scale

kind of relation that is used to classify relations between physical properties and numbers, thus indicating a method for quantifying sizes or extents of aspects by mathematical values or ranges

Note 1 to entry: A scale is meant to provide a mechanism to relate quantitative aspects (physical properties) to numbers or ranges that represent the sizes or intensities of the aspects on the scale. The number 1 on a scale is a unit of measure that refers to a (standard) reference value of the kind of aspect for which the scale is meant.

EXAMPLE length scale, velocity scale, temperature scale

Note 2 to entry: Qualitative scales are usually called units of measure. They are subtypes of scale.

3.1.15
unit of measure

scale that specifies how the size or extent of an aspect is unambiguously quantified by a value on a mathematical range

Note 1 to entry: A unit of measure provides a specific mechanism to relate an aspect to a number or range that represents the size or intensity of the aspect.

EXAMPLE mm, cm, m, km, bar, mbar, mmHg, psi, °C

Note 2 to entry: In fact, the term 'unit of measure' refers to a standard value on a particular scale that is used for comparison. For example, it could be argued that '1 m' is the 'unit of measure', being a standard value on the metre scale, whereas the latter is indicated just as 'm'. That standard value is (approximately) the length of the standard bar in the 'Musée de Mesures' (Museum of Measures) in Paris that was originally used to measure length by comparison.

3.1.16
role

extrinsic aspect that is possessed by a possessor as long as the possessor participates in a relation that requires that role

Note 1 to entry: A role is played by something when participating in a relation with something else. Typically roles are based on temporal situations. Thus they are extrinsic and not on intrinsic aspects. Physical objects can play various kinds of roles: they can play roles in relations with other physical objects, they can play a role as a possessor of an aspect and they can play a role in an occurrence, which is then called a kind of usage, etc. Kinds of roles of physical objects can be distinguished from kinds of physical objects by the fact that an actual role disappears when the physical object is taken away from its normal position and, for example, is put in stock in a warehouse.

Note 2 to entry: Aspects can also play various kinds of roles: they can play a role of being possessed and can play a role in a qualification relation, in a correlation or in some other kind of relation.

EXAMPLE The concepts part, whole, relator, related, involver, involved.

See also: examples of ‘role of physical object‘ and examples of ‘intrinsic aspect‘ (a kind of role of aspect).

3.1.17
role of physical object

role that a physical object plays in a relationship or the contribution that a physical object delivers in an occurrence

Note 1 to entry: A role of physical object is a role that is played by a physical object. Typically it is their usage. Roles of physical objects shall be distinguished from the physical objects that play the roles. A physical object typically loses its role when it is taken out of the context that is typical for that role. Therefore, whether a concept is a kind of role or a kind of physical object (is specifically designed with particular intrinsic aspects) can usually be determined by the answer to the question, “Is the thing on a shelf in stock still recognizable as such?”. If not, then the concept denotes a role.

EXAMPLE ‘chairman‘ is a kind of role that can be played by a person; ‘left hand wheel’ is a kind of role that can be played by a wheel; ‘player’, ‘performer’, ‘subject’, ‘tool’, usage’, ‘customer’, ‘supplier’, ‘part’, ‘whole’, are other examples of kinds of roles of physical objects.

3.1.18
intrinsic aspect

role that an aspect plays in a relationship with a possessor and that is dependent on the aspect as well as on the possessing object

Note 1 to entry: Typically the name as well as the definition of an intrinsic aspect’ includes the kind of physical object that possesses the aspect. It may also be that the aspect is possessed by a part of the assembly that is denoted as the possessor. Possessed aspect is a synonym of intrinsic aspect.

EXAMPLE 1 ‘Pipe diameter’ is an intrinsic aspect that is defined as a diameter that is by definition possessed by a pipe.

EXAMPLE 2 ‘Shaft length’ is an intrinsic aspect that is defined as a length that is by definition possessed by a shaft. ‘Motor power’ may be recorded as an intrinsic aspect of a car, although the power is an aspect that is possessed by a motor, which is a part of a car.

3.1.19
function

role of an occurrence that is intended to be performed or enabled by a physical object

Note 1 to entry: An occurrence (activity, process or event) typically has a relation with a player of a performer and possibly an enabler role. The occurrence has a role as the function (to be performed or enabled) in such a relation. The physical object will play a role as performer or enabler in that relation. So, the function denotes the occurrence.

Note 2 to entry: Sometimes the performer role of the physical object is also called its function. However, this is another concept, being a homonym.

EXAMPLE Pumping is a kind of occurrence that can be performed by a pump. In other words: pumping can be a function of a pump.

3.1.20
objective

role of a state that is intended to be achieved or that is intended to be prevented

Note 1 to entry: Something has a role as objective when it is wanted to be in that state. Typically the objective of an activity. The state can be described by information or by a number of facts that shall be the case.

EXAMPLE An example of an objective might be: 'product A is being produced'. This might be the objective of project P.

3.1.21
collection

concept that indicates a plurality, consisting of a number of things without a particular structure between the elements and not necessarily with a common discriminator

Note 1 to entry: A collection is the result of bringing items together (or as if). Collections shall be distinguished from arrangements, assemblies and classes, kinds or categories. The reason for being element of a collection should therefore not be based on being connected or having a common discriminating aspect alone. Note, the number of elements in a collection may vary over time, and may consist of zero, one or more elements, while nevertheless remaining the same collection.

Note 2 to entry: Apart from this concept 'collection' there also exists a *collection relation* that relates an element to the collection of which the element is a component.

Note 3 to entry: In the context of knowledge libraries, collections are always collections of concepts.

EXAMPLE Stock items, such as a 'stock of bolts', or a pair of items. Not systems, such as 'sewer system', because a system is not a collection of parts, but an assembly or arrangement, which means that it is composed of (physically or functionally) connected or arranged parts. An organization is an example of an arrangement of people which is not a pure collection, because the persons have a relative position towards each other.

3.1.22
individual thing

concept that classifies any real world or imaginary thing that has an individuality that is not dependent on a commonality between things

Note 1 to entry: This standard is about kinds of things that are defined as commonalities between things, defined by the constraints on aspects or ranges of values for aspects of individual things. Those kinds of things can be used to classify individual things or to derive constraining aspects for individual things.

The concept that classifies all those individual things is called 'individual thing'. The concept 'individual thing' is the supertype of all kinds of individual things.

EXAMPLE Well known individual things are: the earth, the Eiffel Tower, New York, my car, V-6060 in the Shell Pernis refinery. However, planet, tower, city, car, vessel and refinery are not individual things, but kinds of things.

3.2 Terms and definitions for kinds of relations

For the purposes of this document, the following terms and definitions of kinds of relations apply.

NOTE Kinds of relations are also called relation types or fact types. Each of the definitions of binary relation types is accompanied by the following information:

— the definition of which kinds of objects are related in such a relation (the R1 role player and the R2 role player);

— the kinds of roles that those concepts by definition play in such a relation (the R1 role and the R2 role);

— the expressions (phrases) that represent the relation type in natural language (the R1-R2 expression and the inverse R2-R1 expression).

Furthermore one or more example instances are given that illustrate the use of the relation type to express facts.

Note that the kinds of related objects and the kinds of roles they play are characterizing the relation type.

3.2.1
relation

concept that expresses a fact or opinion about a fact by specifying the things that are involved in the fact and the roles that the various involved things play in the fact

Note 1 to entry: Each fact or state of affairs can be modelled as and expressed by a relation (relationship) between related things. The kind of relation (also called relation type or fact type) specifies how something relates to something else. The related objects specify what is related. Binary relations relate two things. Higher order relations relate more than two things. Each related thing has its own role of its own kind in the relation. Thus a relation indicates that a number of things are related to each other. If one of the related things is a plurality, then the relation implies multiple facts.

EXAMPLE The Eiffel Tower and Paris are related to each other. The relation is of the kind <is located in>. The Eiffel tower has a role as located in the relation and Paris has a role as locator in the relation. Activities and processes are typical examples of things that can be expressed as higher order relations.

3.2.2
relation between individual things

relation that relates an individual thing with another individual thing

Note 1 to entry: A fact in which individual things are involved can be modelled by a relation between individual things. The kind of relation specifies how the things are related.

EXAMPLE The fact that the Eiffel Tower is located in Paris is a fact that can be expressed by a relation between individual things (the Eiffel Tower and Paris), whereas the relation can be classified by a kind of relation called ‘is located in’.

3.2.3
relation between kinds of things

relation that specifies knowledge or requirements or permissions in general terms about what can be the case, shall be the case, is allowed to be the case or is by definition the case

Note 1 to entry: A fact about kinds of things can be expressed by a relation between kinds of things. Such a fact typically expresses what can be the case for all things of those kinds, possibly within a specified context. Specializations of this kind of relation can constrain what can be the case to what shall be, is allowed to be or is by definition the case.

EXAMPLE All general knowledge, such as about possible compositions of things of a kind and about kinds of aspects that all things of a kind share.

3.2.4
relation between an individual thing and a kind of thing

relation that relates an individual thing with a kind of individual thing

Note 1 to entry: A kind of relation that specifies that an individual thing has a relation with a kind of thing or can have a relation with things of a particular kind. A classification relation is an example of a subtype of this relation.

EXAMPLE The fact that Paris is classified as a city and the fact that the individual object V-6060 is classified as a horizontal vessel. The fact that T-6000 can be used for storage of drinking water.

3.2.5
binary relation

relation that specifies a relationship between two things, each of which is playing its own role that is of a kind that is typical for the relation type

Note 1 to entry: Facts can be expressed as a binary relation or as a collection of binary relations between things. Most kinds of facts can be expressed using a single binary relation. Some facts require ternary or higher order relations. Those relations can be expressed using multiple binary elementary relations.

Note 2 to entry: This relation type is the top of the specialization hierarchy of binary relation types. It can be used to record that things are related without knowing how they are related, but usually more specialized relation types are used.

EXAMPLE The fact that The Eiffel Tower is located in Paris is a fact that can be expressed as a binary relation between The Eiffel Tower and Paris, whereas the relation type is ‘being located in’.

A composition relation is a binary relation type that relates two things. One of those things plays a role as part and the other plays a role as whole. Each binary relation type can be denoted by a phrase, such as ‘can be a part of a’. In the inverse sequence the same relation type can be denoted by an inverse phrase, such as ‘can be a whole for a’.

Activities are higher order relations that can be expressed by a number of binary elementary relations, where each binary relation specifies the role of an involved thing in the activity.

Example instances:

John	is related to	Peter
force	is related to	acceleration

3.2.6
specialization relation

relation between kinds of things that relates two concepts whereby the subtype concept is a more specific concept than the supertype concept and has all the aspects that define the supertype concept

R1 role player: concept		**R2 role player:** concept	
R1 role: subtype		**R2 role:** supertype	
R1-R2 expression:	is a specialization of is a kind of is a subtype of	**R2-R1 expression:**	is a generalization of has as subtype is a supertype of

Note 1 to entry: The constraints by which a supertype concept is defined are also applicable for its subtype concepts. A subtype concept is distinguished from its supertype and its neighbouring subtype concepts by being defined by additional constraints. A concept may be a subtype of more than one supertype concept. An aspect (value) by which a supertype concept is defined is also an aspect of all of its subtype concepts (the aspects are ‘inherited’). The aspects of a concept shall also be applicable for the individual things that are classified by the kind. An individual thing that is classified by a concept (thus satisfying its defining constraints), is implicitly also classified by the supertypes of the concept. Knowledge about options for a concept is also knowledge about its subtypes, unless the knowledge is further constrained by the definition of the subtype.

The phrase ‘is a specialization of’ has as synonyms ‘is a kind of’ and ’is a subtype of’. The inverse phrase ‘is a generalization of’ has as synonyms ‘has as subtype’ and ‘is a supertype of’.

Thus, the expression A is a kind of B, means that the concept A is a subtype of the concept B. For example, kinds of aspects (= subtypes of aspects) are: length, width, temperature, colour, etc.

Note 2 to entry: The term that denotes the subtype has a role as hyponym. The term that denotes the supertype has a role as hypernym.

EXAMPLE Assume that ‘means of transport’ is defined as a physical object that is intended to carry load. Furthermore, it is specified that concepts with the names ‘car’ and ‘ship’ are both a specialization of ‘means of transport’. Then this implies that car as well as ship are intended to carry load (without the need to explicitly specify those facts.) Furthermore, assume that individual object #12 is classified as a ‘car’ then the specialization relation implies that object #12 is also a ‘means of transport’.

The concept ‘width’ is a specialization of ‘distance’. If for distance it holds that it can be quantified on a length scale, then that implies that width inherits from distance that it also can be quantified on a length scale.

Example instances:

car	is a specialization of	means of transport
means of transport	is a generalization of	car
ship	is a specialization of	means of transport

Note 3 to entry: The first two examples above are different expressions of the same fact.

3.2.7
qualification relation

kind of specialization relation that relates two concepts whereby the subtype is a qualitative concept and the supertype is a conceptual concept

R1 role player: *qualitative* concept	**R2 role player:** *conceptual* concept
R1 role: qualifier	**R2 role:** nature
R1-R2 expression: is a qualification of	**R2-R1 expression:** is the nature of

Note 1 to entry: This relation type is intended to distinguish qualitative concepts from their conceptual concepts as counterparts, while still being subtypes of them. Thus the relation type is a subtype of a specialization relation. In case of an aspect, it specifies that the qualitative or quantitative aspect (also called aspect value, with subtypes property value and quality value) are defined to be a qualification (specialization) of an appropriate (generic) conceptual aspect. In case of a physical object, it specifies that the subtype is a type of physical object that has a number of aspects with specified qualitative or quantitative aspect values. Qualitative concepts are concepts with one or more fixed aspect value.

EXAMPLE The concept ‘colour’ is a conceptual aspect. The concept ‘red’ is a qualitative aspect. ‘Red’ has a qualification relation with the conceptual aspect ‘colour’.

Similarly ‘37 °C’ is a quantitative aspect that is a qualification of the conceptual aspect ‘temperature’.

And ‘100’ is a qualification of the conceptual aspect ‘number’.

The concept ‘6x50 mm hexagonal galvanized bolt’ is a type of physical object (and thus a qualitative concept) that is a qualification of the concept bolt.

Example instances:

red	is a qualification of	colour
mm	is a qualification of	length scale
deg C	is a qualification of	temperature scale
37 deg C	is a qualification of	temperature
100	is a qualification of	number
6x50 mm hexagonal bolt	is a qualification of	hexagonal bolt

3.2.8
manufacturer’s model of physical object relation

special kind of qualification relation that relates two concepts whereby the subtype is a manufacturer’s model of a kind of physical object and the supertype is concept that is a kind of physical object

R1 role player: physical object	**R2 role player:** physical object
R1 role: manufacturer’s model	**R2 role:** nature
R1-R2 expression: is a model of	**R2-R1 expression:** is the nature of model

Note 1 to entry: This relation specifies that manufacturer’s models of physical objects are models (qualifications) of more generic concepts. This relation is a subtype of a qualification relation where the subtype is a manufacturer’s model of the supertype concept. A manufacturer’s model is also called ‘model and size’.

A manufacturer’s model of physical object relation should be distinguished from a classification relation, because a manufacturer’s model is still a kind of thing, whereas a classification relation is intended for the classification of individual things, such as a particular installed item, indicated by its asset registration number.

EXAMPLE A type of car, such as an ‘Audi Q7’ is a manufacturer’s model of the generic concept ‘car’.

Example instances:

Audi Q7	is a model of	car
ALLWEILER model SNH 80	is a model of	twin screw pump

3.2.9
composition relation

relation between two kinds of things that specifies that something that is classified by a concept that plays the role of whole can have one or more components that are classified by another concept that plays the role of part, in which the number of parts may be constrained by cardinality constraints

R1 role player: concept	**R2 role player:** concept
R1 role: part	**R2 role:** whole
R1-R2 expression: can be a part of a	**R2-R1 expression:** can be a whole for a

Note 1 to entry: The relation specifies that things of a kind can be components of things of the other kind. The parts may be assembled in the whole or may just be connected or arranged to form a whole.

The composition relation shall be distinguished from a collection relation.

In a particular composition relation the physical object that plays the role of part may not be the same as the physical object that plays the role of whole. Note that also this fact is inherited to subtypes of the related physical objects.

EXAMPLE A ‘wheel’ can be a *part of* a ‘car’; a ‘valve’ can be part of a ‘piping system’ and a ‘road’ can be part of a ‘road network’. Because of inheritance this means that a wheel can also be part of a Volvo S40 and that a ball valve can also be part of a piping system (or one of its subtypes).

Example instances:

car	can have as part a	wheel

3.2.10
physical object – aspect relation

relation between a kind of physical object and a kind of aspect that specifies that physical objects of the specified kind can have or has by definition an aspect of the specified kind

R1 role player: physical object	**R2 role player:** aspect
R1 role: possessor	**R2 role:** possessed
R1-R2 expression: can have as aspect a	**R2-R1 expression:** can be an aspect of a

Note 1 to entry: Normally the kind of physical object has the kind of aspect as an intrinsic aspect. If the thing that is related to the physical object appears to be a concept that exists independent of the physical object, then it is not really intrinsic. This means that other relation types should be considered. In this kind of relation the possessor (the R1 role player) shall be a physical object (and may not be something else) and the thing that is possessed (the R2 role player) shall be an aspect (and nothing else). In such a relation a physical object plays a role as possessor (R1) and an aspect plays the role of possessed (R2).

Note 2 to entry: Occurrences can also have some aspects, especially duration and timing aspects, however those relations are usually considered not relevant for knowledge libraries. Other aspects are normally aspects of the physical objects that are involved in the occurrences.

EXAMPLE A ‘wheel’ *can have as aspect a* ‘diameter’.

A manufacturer or supplier of a bicycle is not an intrinsic aspect of a bicycle (the manufacturer exists independent of the bicycle) and thus they shall be related to a bicycle by another relation type.

Example instances:

wheel	can have as aspect a	diameter

3.2.11
aspect–scale relation

relation that specifies that aspects of the specified kind can be quantified using the kind of scale

R1 role player: aspect	**R2 role player:** scale
R1 role: quantifiable	**R2 role:** quantifying
R1-R2 expression: can be quantified on scale	**R2-R1 expression:** can be a scale for a

Note 1 to entry: An aspect of a particular kind may be quantified on various units of measure that are qualifications of the same kind of scale and sometimes may even be quantified on units of different kinds of scales. This relation type can thus be used to specify that aspects of a particular kind can be quantified on all units of measure that are qualifications of a particular kind of scale.

Note that this fact is inherited to all the subtypes of the kind of aspect.

EXAMPLE It may be specified that e.g. distance can be quantified on scale (by a number using a) length scale. This fact is then inherited by the subtypes of distance, such as length, width, diameter, and radius. This implies that those subtypes can also be quantified on any length scale.

Furthermore, it will be defined that, for example, mm, km and inch are qualifications of length scale (see the example instances of a qualification relation). Then this implies that distance as well as its subtypes (such as length) may be quantified on a mm, km or inch scale.

Example instances:

distance	can be quantified on scale	length scale
length	can be quantified on scale	mm

3.2.12
aspect – unit of measure relation

relation that specifies that aspects of the specified kind *shall be* quantified using the particular unit of measure

R1 role player: aspect

R2 role player: unit of measure

R1 role: quantifiable

R2 role: quantifying

R1-R2 expression: shall be quantified on unit

R2-R1 expression: shall be a unit for a

Note 1 to entry: This relation type can be used to specify that the magnitude of aspects of a particular kind *shall be* quantified using a particular unit of measure (which is a subtype of scale). This requirement is valid only in an implicit or explicitly defined 'validity context', such as a particular project or company or knowledge library.

Note 2 to entry: This fact is inherited to the subtypes of the specified kind of aspect, unless it is overruled by a new specification.

EXAMPLE The aspect – unit of measure relation may be used to specify, for example, that every length shall be quantified on a mm scale, a requirement that is only valid in the validity context of project X.

Example instances:

distance	shall be quantified on unit	km
length	shall be quantified on unit	mm

3.2.13
involvement relation

relation between kinds of things that specifies that physical objects of a kind can play a role in an occurrence of a kind

R1 role player: physical object

R2 role player: occurrence

R1 role: involved

R2 role: involver

R1-R2 expression: can be involved in a

R2-R1 expression: can involve a

Note 1 to entry: This relation specifies that a physical object can be involved in an occurrence, but it does not specify how it is involved (in which kind of role). This might be specified by using subtypes of this relation type.

EXAMPLE A boiler can be involved in a water heating process. A locomotive can be involved in a rail transport.

Example instances:

boiler	can be involved in a	heating process
locomotive	can be involved in a	rail transport

3.2.14
collection of concepts relation

relation that specifies that a concept is an element of a collection of concepts

R1 role player: concept	**R2 role player:** collection
R1 role: collected	**R2 role:** collector
R1-R2 expression: is an element in collection of concepts	**R2-R1 expression:** is a collection of concepts including

Note 1 to entry: This relation only specifies that a kind of thing is element of a plurality of kinds. It does not say anything about the thing itself, nor about the nature of the collection nor about relations between the concepts that are elements in the collection. A collection may have only meaning within a particular context. The relation is typically used to define lists of options or pick-lists.

EXAMPLE The subtype of physical object ‘lock’ can be defined to be an element in the collection of concepts ‘ironmongery’ in a particular context. ‘red’, ‘green’ and ‘blue’ are qualitative aspects that are elements in the collection of concepts ‘RGB colours’. That collection is typically called a list of (allowed) values and may form a ‘pick list’.

Example instances:

lock	is an element in collection of concepts	ironmongery
red	is an element in collection of concepts	RGB colours

3.2.15
naming relation

relation that relates a term with something (UID) that is denoted by the term

R1 role player: term	**R2 role player:** anything
R1 role: name	**R2 role:** named
R1-R2 expression: is a name for	**R2-R1 expression:** has as name

Note 1 to entry: This relation defines how things are called in a particular language and language community. A naming relation is defined within the context of a language community as the preferred name in that community. It may be used also outside that language community. This relation allows for multiple names for the same things, thus enabling synonyms, abbreviations, codes and translations.

The same name may be allocated to different concepts, provided that they are defined in different language communities. This enables to specify homonyms.

EXAMPLE Physical object type 670171 from the taxonomy has as name ‘bicycle’ in a particular vocabulary in the English language. The same object has the name ‘fiets’ in the Dutch language. PC and personal computer are two different names for the same concept.

Example instances:

760171	has as name	bicycle (UK English)

3.2.16
description relation

relation that specifies that a textual description is a description of something

R1 role player: character string   **R2 role player:** anything

R1 role: description   **R2 role:** described

R1-R2 expression: is a description of   **R2-R1 expression:** is described as

Note 1 to entry: This relation defines that something is described by a piece of text in a particular language. This relation allows for multiple descriptions for the same things. A definition is a particular kind of description. Thus a definition relation might be defined as a subtype of a description relation.

EXAMPLE Physical object type 670171 is described using a particular description (sentence) in the English language (see the example below).

Example instances:

670171 is described as a cycle with two wheels in tandem, a steering handle, a saddle seat, and pedals by which it is propelled. It may also be propelled by a motor. (UK English).

3.2.17
synonym relation

relation between two naming relations that relates an object with a particular name in one context with the same object with another (or the same) name in another context in the same language

R1 role player: naming relation   **R2 role player:** naming relation

R1 role: synonym   **R2 role:** base

R1-R2 expression: is a synonym of   **R2 R1 description:** is a synonym for

is an abbreviation of   is abbreviated by

Note 1 to entry: This is a basic relation for linking the elements from various functional units or in different knowledge libraries. Essentially it is an equivalent for a naming relation, because if one object (indicated by a unique identifier) has more than one name, then those names are by definition each other's synonym. For clarity the relation phrase distinguishes between synonyms and abbreviations. Formally an abbreviation relation is a subtype of a synonym relation. Note that a name that is allocated via a synonym relation is defined within the context of a language and a language community (just as is the case with a naming relation). See also the translation relation.

A preferred term or preferred name is only preferred in some language community, whereas another term may be preferred in another language community. Therefore the synonym relation does not specify which term is preferred. Preferences for terms should be addressed by defining language communities with their preferred terms.

EXAMPLE The name 'bicycle' in the taxonomy is a synonym of the name 'bike' as a reference to a particular kind of physical object in the English dictionary. The name 'pump' in an Aspect Model is a synonym of the name 'pump' in a Composition Model.

Example instances:

English (130206) bike	is a synonym of	English (130206) bicycle
English (70620) PC	is an abbreviation of	English (70620) personal computer

3.2.18
translation relation

relation between two naming relations that relates an object with a particular name in one language with the same object with another (or the same) name in another language

R1 role player: naming relation	**R2 role player:** naming relation
R1 role: translation	**R2 role:** base
R1-R2 expression: is a translation of	**R2 R1 description:** is translated by

Note 1 to entry: This relation is similar to the synonym relation, but in this case the two contexts for the two names of the object are two different languages.

EXAMPLE A translation of the name ‘Pumpe’ as a reference to concept 130206 in a German–English dictionary.

Example instances:

English (130206) pump	is a translation of	Deutsch (130206) Pumpe

3.2.19
classification relation

relation that relates an individual thing to a kind of thing, indicating that the individual thing is of the specified kind, because of the fact that the individual thing has aspects that comply with the definition of the kind

R1 role player: *an individual thing*	**R2 role player:** concept
R1 role: classified	**R2 role:** classifier
R1-R2 expression: is classified as a	**R2-R1 expression:** is a classifier of

Note 1 to entry: Although this standard is not about individual things, this relation is included to support the application of knowledge libraries. It is a relation that defines for individual things of what kind they are. Thus every usage of this relation relates an individual thing to a concept, preferably a concept in a knowledge library.

Classifying relation is a synonym of classification relation.

This relation shall be distinguished from a collection relation for individual things, because a classification is based on intrinsic aspects of the classified objects, whereas a collection is not.

Note 2 to entry: In information technology terms the concept ‘instantiation relation’ is often used. However, an instantiation relation differs from a classification relation. because it is a relation between an instance and a class (being a kind of thing or a set), whereas an instance can be an individual thing, but may also be a kind of thing. (An instantiation relation seems to be a combination of a classification relation, a qualification relation and a collection relation.)

EXAMPLE ‘my car’ is classified as an ‘Audi Q7’;

‘mars’ is classified as a ‘planet’;

’V-6060’ is classified as a ‘horizontal vessel’.

Example instances:

New York	is classified as a	(700008) city
V-6060	is classified as a	(520121) horizontal vessel

4 Symbols and abbreviations

CL	Conformance Level
DTD	Document Type Definition

E/M/S	Electrical engineering, Mechanical engineering and Sanitary installations
iGBi	interactive platform for construction and ICT
IFC	Industry Foundation Classes
IFD	International Framework for Dictionaries
ISO	International Organization for Standardization
TAX	taxonomy
UID	unique identifier
UML	Unified Modeling Language
VOC	vocabulary, list of terms
XML	eXtensible Mark-up Language
XSD	XML Schema Definition

5 Objectives

5.1 Introduction

In this clause, the objectives of the standard are described. 5.2 gives an outline of the scope and preconditions for a possible solution. In 5.3 a problem definition is given. This will constitute the basis for a further discussion of the objective envisaged in 5.4. Finally, 5.5 is devoted to a discussion of how the objective is realized by means of guidelines.

5.2 Scope, conditions and target users

The following items are within the scope of this standard.

— The guidelines are aiming for harmonization between knowledge libraries in terms of content in relation to structure and architecture. The guidelines are grouped in conformance levels, called conformance level 0 and 1 (see A.1.2). Conformance level 0 will guarantee that fundamental differences between similar types of knowledge libraries are prevented and that a basis is provided for data exchange and integration. Conformance level 1 aims to provide guidance for cooperative use of different knowledge libraries.

— Apart from the limited number of guidelines, the scope of knowledge libraries is limited as well. The scope includes information concerning concepts such as kinds of *physical objects,* their *aspects* and *roles, kinds of occurrences* (kinds of activities, processes and events that may happen and in which physical objects can be involved), *collections, functions*, kinds of *organizations* and *objectives*.

— Interpretations of or guidelines on the actual exchange of object data are excluded from the scope. The guidelines are limited to the enforcement of the harmonization of information about concepts. The existence of comparable data should then be the point of departure for drawing up agreements for data exchange.

— Guidelines for the modelling of knowledge about kinds of things (concepts, classes and types) are included as well as guidelines about classification of individual things by kinds of things (although the latter do not belong to a conformance level). Guidelines for facts about individual things other than their classification are excluded from the scope

The guidelines in this standard are made with the following condition:

The guidelines in this standard should not conflict with any of the following standards, provided that those standards are not in conflict with each other:

a) ISO 10303;

b) ISO 12006-3;

c) ISO 13584;

d) ISO 15926.

The guidelines have the following target users:

- The target users consists of: developers of knowledge libraries, builders of translation programs or interfaces between knowledge libraries, certifying bodies and builders of applications who must base their work on the knowledge libraries that have been laid down.
- The guidelines are not aimed at a specific industry.

5.3 Problem definition

Despite the various initiatives that have been started in this area, the number of successful knowledge libraries is still limited. Apparently there are issues that prevent the potential value from being realized. In this subclause the most important issues are described, which will illustrate the need for guidelines on knowledge libraries.

In the first place, the various initiatives for developing knowledge libraries lack a common methodology. At the moment, many companies are unaware of current developments and the extent to which they can conform to a common language. Moreover, a library must gain a certain critical mass before it may receive sufficient support. In many cases this critical mass is still absent. Most of the existing information libraries have a limited scope and focus on island applications instead of data exchange and data integration. Altogether, this leads to fragmentation of information maintained in separate libraries. Initiatives with knowledge libraries have to struggle with the learning process of their users.

A second problem is that, paradoxically, there is a measure of resistance to migration to large libraries in the world of knowledge libraries. In many cases, organizations have invested substantially in their own libraries, and often their user groups will already have conformed to the architecture of those libraries. Migration or even the simple linking of libraries is usually found to be an (impermissible) extensive exercise. The architectures of libraries often differ, in addition to which the intrinsic definitions of concepts rarely match. For instance, the way in which a pump is defined in library A may differ substantially from its definition in library B.

A third issue concerns the quality of libraries. Setting up the architecture and filling such libraries make major demands of the competences of the responsible parties. For example, incorrect standard specification sheets may result in a considerable deterioration in the quality of a library. Organizations that work with such a library will eventually decide to resort to traditional methods and choose to circumvent the library. Limited quality will frequently stimulate other parties to set up new library type, which increases the problem of fragmentation.

Finally, experts have not yet reached consensus on the architecture for knowledge libraries. In the world of ISO there are currently various projects with regard to the structure (i.e. information model) of knowledge libraries. This means that not only the content of libraries, but also the architecture of that content is still under development.

5.4 Objective of this standard

The motivation for this effort may be deduced from 6.3, in which the added value of knowledge libraries is discussed. The effectiveness of almost all forms of applications of knowledge libraries will increase with their harmonization or integration.

A suitable solution to this problem is the harmonization of knowledge libraries by means of the use of a common dictionary and taxonomy, including the common use of standard relation types. By linking knowledge libraries to such a common dictionary and taxonomy in which the meaning of the various concepts and relation types have been defined (see Figure 1 — Integration of libraries by means of an integrating library), the relations that express knowledge in the libraries will be laid down in way in which they can be integrated and commonly searched. This means that when a knowledge model of a concept is added to a knowledge library, it will suffice to refer to the appropriate concept in the common dictionary. It follows that references to knowledge in other libraries may be realized automatically by such a reference, provided that appropriate access rights are granted.

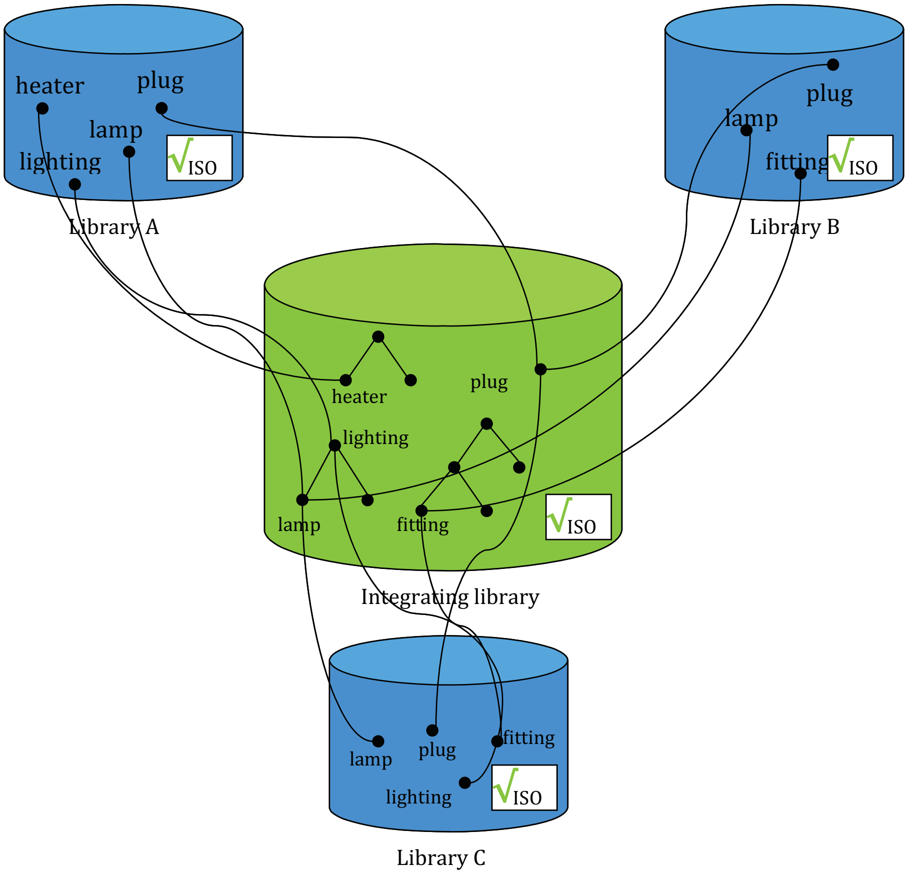


Figure 1 — Integration of libraries by means of an integrating library

This standard provides guidance for a first step towards such integration by supporting the harmonization of the structures of knowledge libraries. The extent to which a library conforms to this standard determines the extent to which harmonization can be achieved with other libraries that conform to this standard.

The objective of this standard is to provide guidelines and conformance checking criteria for storage of electronic information about concepts and relation types; these guidelines enable harmonization between knowledge libraries. This will simplify the integration of knowledge libraries and will simplify the common usage of information from different knowledge libraries.

With this objective in mind, a solution to the problems described in 5.3 can be realized as follows.

— This standard provides a basis for harmonization of the various initiatives with regard to knowledge libraries. Through compliance with this standard the diversity among libraries is reduced and the threshold towards integration is lowered.

- This standard respects existing libraries and their users. Each owner of a knowledge library may gradually upgrade his library in order to make it compliant with this standard. Taking into account the reduction of mutual differences, integration or linking may be considered at a later stage.
- This standard offers a possibility for users to know what can be expected from knowledge libraries that meet certain requirements, and to know how they will be able to make more efficient use of such a library.
- The consensus among the various experts has been increased. This standard focuses mainly on ‘that which connects us’ rather than ‘that which separates us’.

It should be noted emphatically that this standard does not pronounce on the preferred strategy for effectuating the integration of libraries. For instance, it is quite likely that there is a need for initially linking libraries by means of 1:1 (i.e. one-on-one) links and then to migrate to an application on the basis of an integrating library, also called a smart dictionary. With such 1:1 links a bridge is built so that each of the functionally different libraries may yet utilize the knowledge stored in the other. The disadvantage of this unit lies in explosive growth of the number of links, susceptibility to maintenance and the very complex quality assurance (see Figure 2 — Linking knowledge libraries by means of 1:1 links). However, for pragmatic reasons this may be a good compromise.

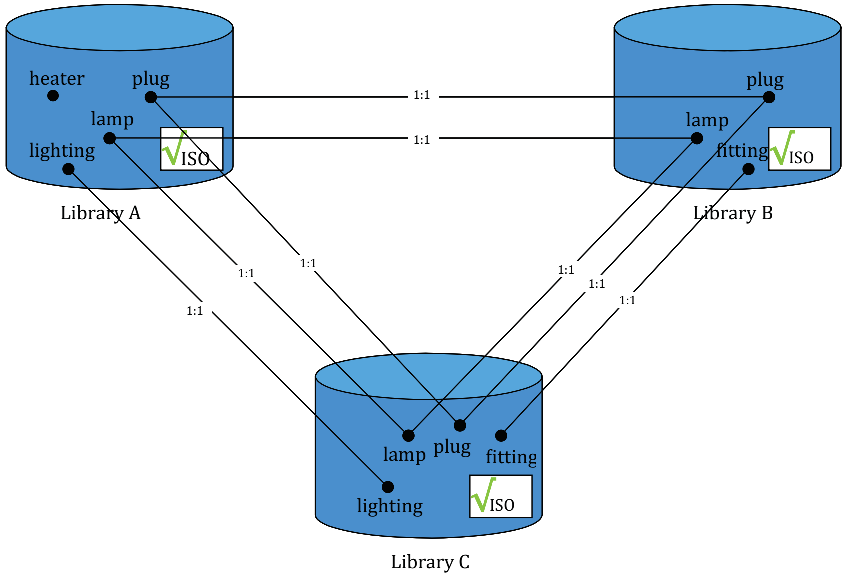


Figure 2 — Linking knowledge libraries by means of 1:1 links

5.5 Guidelines as instrument

In order to realize the objective discussed above, this standard has opted for creating *guidelines*. Provided they are sufficiently clear, guidelines offer a concrete tool for setting up and filling knowledge libraries. Through compliance with the guidelines a certain quality of the library is achieved.

Knowledge libraries are categorized in this standard in a framework of functional units as presented in 6.3. A functional unit denotes the kind of information that can be generated by the library, independent of its internal structure. Guidelines are provided specifically for each functional unit. In addition a number of guidelines apply for more than one functional unit.

5.5.1 Possible misconceptions

It has been found that the supposed consequences of the application of these guidelines to existing knowledge libraries or libraries currently under development may lead to misconceptions. In 5.5.1.1 to 5.5.1.4, four misconceptions will be discussed in order to clarify the context of the guidelines.

5.5.1.1 Internal structure of knowledge libraries

The first misconception concerns the internal structure of (existing) library systems. Should a library owner adapt the structure of his library in order to comply with the guidelines? The answer is: no, *provided the owner is capable of delivering data, according to the defined functional units in compliance with the standard*. Guidelines, therefore, provide no comments on the library itself, but only on the architecture of functional units.

A guideline in this standard does not provide any comments on the way in which the internal structure of knowledge libraries should be defined (the underlying information model), nor does it indicate which data must be recorded; the guideline only provides information on the functional unit structure and semantics that might be generated on the basis of an information model and the available data in a database.

This important remark is illustrated in further detail in Figure 3 — Relation between knowledge libraries and functional units:

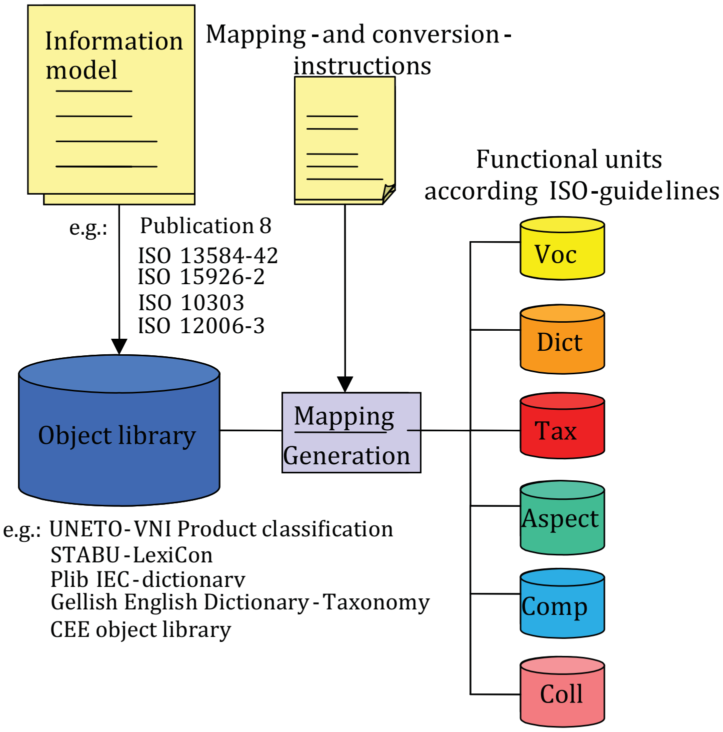


Figure 3 — Relation between knowledge libraries and functional units

On the left side of Figure 3 an (existing) knowledge library is shown. This library has been filled with concepts using a particular meta-model or information model[1]. In order to promote the exchange and integration with other libraries, the owner of that library may decide to have his library assigned the standard conformance level. To this end, it must be investigated whether information can be generated according to which functional units (the 'jars' on the right in Figure 3) *and* whether the generated information can comply with the guidelines that apply for those functional units. From this 'confrontation' a number of conclusions may be drawn.

- Information can be generated according to one or more functional units by mapping directly from the information model. This mapping will be documented by way of instructions.
- Information may be realized according to certain functional units provided certain conversions are performed. These conversions will also be laid down by means of additional instructions.

[1] In most cases the meta-model is described by a specified information model, e.g. in EXPRESS. These guidelines apply to the instances of such a model only.

- Information cannot be generated according to certain functional units since the information model does not allow for the information that is required for this purpose and is therefore absent from the library. Many existing knowledge libraries merely offer such a limited functionality.
- Information cannot be generated according to certain functional units because the knowledge library does do not comply with the guidelines. It may also occur that the knowledge library (and the corresponding information model) does provide for a certain functionality, but that it is has been supplied in a fundamentally different way, rendering conversion impossible. In that case, a library upgrade may be considered so that this becomes possible in the future.
- Certain information in the library cannot be mapped since there is no adequate functional unit provided in the other library.

After performing the analysis and possibly adding modifications, the library, together with the matching instructions, may be offered to the inspecting body for testing and possibly to obtain the quality label. This will be discussed in further detail in Clause 7.

5.5.1.2 Overlapping of information

A second, frequently observed, misconception concerns the uniformity of information within the various functional units.

Functional units can and may overlap with regard to information content; each functional unit fulfils a particular function, and some functions use the same information.

It is very tempting to abstract from the functional units and their guidelines a single, generic basic model for knowledge libraries. If one does so, the question rightly arises as to why, for example, the same words that have been defined in the functional unit 'dictionary' are also found in the functional 'taxonomy' unit. Can they really occur twice? Why would one include aspects in the Aspect Models functional unit, while they may be deduced from aspects that have been defined in the taxonomy unit?

However, these questions are irrelevant, because the guidelines have not been created in order to outline a preferred information model for knowledge libraries. The entities from the underlying information model of a knowledge library are cases used for generating multiple functional units. For instance, it is very likely that a taxonomy-oriented library system in which the various concepts have assigned aspects will be able to generate data according to the functional units Dictionary, Taxonomy and Aspect Models. In Figure 3, in the symbolic representation the sum of the 'jars' on the right will therefore contain duplicate data[2].

5.5.1.3 Quality of information

A third misconception concerns the quality of the information that is produced according to the functional units. If knowledge library A presents its data according to a certain functional unit and knowledge library B does the same, why should I believe that the same qualitative level applies to both? How do I know for sure if the underlying meaning is identical with respect to the functional unit of A and B?

The generated functional units from the different knowledge libraries are mutually comparable if these libraries formally present their data in functional units in compliance with the guidelines in this standard.

This remark implies that to draw up instructions for generating functional units in compliance with the standard in accordance with one's own judgment is not sufficient. It should be verified by means of an internal or external test centre whether the guidelines are truly complied with. In the future an official, inspecting body may play an important role. The quality label that is finally awarded should guarantee that knowledge library A will know the value of a functional unit generated by knowledge library B. The

[2] Critical readers would argue that the various guidelines and the identified objects in the various functional units might be laid down in the form of an (EXPRESS) information model. This in itself is a correct assumption. However, it must be considered as the reference model of the various functional units rather than the information model of the (underlying) object library. This reference model is illustrated in Annex D.

quality label will be discussed further in Annex C. It has been illustrated in Figure 4 — Illustration of two knowledge libraries in compliance with this standard.

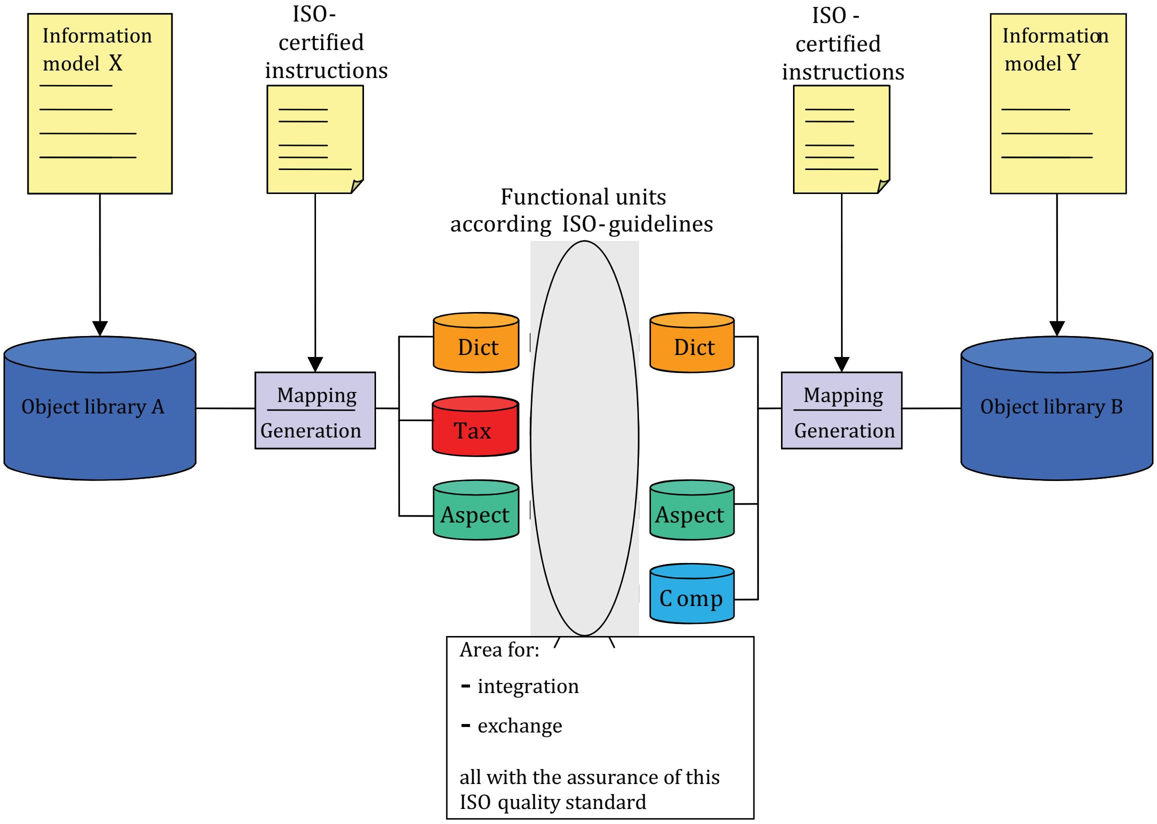


Figure 4 — Illustration of two knowledge libraries in compliance with this standard

5.5.1.4 Harmonization of content

Finally, there exists a fourth possible misconception. This concerns the similarities and differences regarding the *content* of functional units derived from various knowledge libraries.

Standard guidelines do *not* bring about an *intrinsic* harmonization of objects laid down in the various libraries; only the structure, order and restrictions may be compared.

If a pump has been defined in knowledge library A on the basis of various aspects, and the same has been done in knowledge library B, it may be that the aspects of this pump in both knowledge libraries differ completely from each another. The only thing that will definitely be similar is the way in which these aspects have been laid down or described in the relevant functional unit. This is a major step ahead through which a number of important objectives can be realized (see also 5.4). For instance, it offers a much better point of departure for migration to a common definition of 'pump'. If both the description and structure are completely dissimilar, the gap will in most cases be too wide to bridge.

6 Types of knowledge libraries

6.1 Introduction

The world of knowledge libraries is far from uniform and unambiguous. Apart from the fact that knowledge libraries may differ from one another, the image people have of such libraries will often differ as well. In this clause the concept of knowledge libraries will be discussed in further detail. In 6.2

we set out to present a number of examples of knowledge libraries. In 6.3 we proceed by discussing the added value of libraries. Finally, in Clause 7 a framework is presented in which the various functions are explicitly described.

6.2 Definitions and types of knowledge libraries

A *knowledge library* (within the context of this standard) is a collection of knowledge models and/or requirements models, which are models that express knowledge (which may include definitions) and/or requirements about kinds of things (concepts). The expressions of knowledge and requirements are stored and retrieved as electronic information (as information models). Such knowledge is intended to be used for the creation, categorization or verification of information about individual things.[3] The defined concepts are intended to be used as anchoring points for the knowledge and requirements (possibly expressed in documents) about those concepts.

On the basis of this definition, various types of knowledge libraries may be distinguished that clearly differ from a functional and content point of view. For this reason a relatively broad definition for knowledge libraries is used.

An *object library* is a collection of knowledge models (possibly also including definitions and requirements) about kinds of physical objects.

Within the world of object libraries, almost anything might be considered an object. Therefore, a clear delineation is required.

First, we would like to distinguish between individual things (also called individual instances) and kinds of things, also called concepts or classes. Examples of individual things are: my chair, computer #32131, pump #12 at a particular location, the earth and the Eiffel Tower. Examples of concepts are: chair, computer, pump, planet and tower, but also making, inspection, length, shape and relation: in short, things that you will typically find in a dictionary. This standard deals with concepts (classes). It also provides some guidance for the usage of those concepts through guidelines for the classification of individual things. The essential difference between a concept (class) and an individual thing is that a concept is a commonality of a number of individual things that can be used to denote the nature of those individual things. For example, the concept 'chair' denotes what all individual chairs have in common. A concept can also be a type or model of which many individual things are made. For example, an 'Audi Q7' is a concept that is a type or model of a car. You can never point to a concept or class or kick it, whereas you can point to or kick many individual things.

NOTE The notion of 'class' as frequently used in the IT world is different from the notion of concept (class) as described above. Because in IT a 'class' consists of a concept (often called 'entity type'), including a particular collection of 'attributes' and possibly 'behaviour'. This implies that different data models may contain different 'classes' for the same concept, each with its own attributes. In this standard a concept is distinguished from facts about the concept, whereas an 'attribute' or 'behaviour' in a data model will be represented by a fact that relates the concept to another concept.

In actual practice people are often not aware of the distinction between individual things and kinds of things. For instance, when people say that they talk about the description of 'an airplane' then they may either mean that they describe a particular airplane or they may mean to describe (the concept) 'airplane' (that which makes any airplane an airplane).

Knowledge libraries are based on this distinction. They only deal with kinds of things.

A further characteristic of knowledge libraries is that the knowledge is expressed in models. This means that the information or knowledge is expressed in the form of a data structure (also called a data model or knowledge model). Such a data structure can be visualized as a network of related things. The nodes in the network represent concepts, the vertices represent relationships (or relations) between those concepts (individual things will rarely appear in knowledge models). One relation with its related concepts together forms the expression of a single (atomic) fact, being a small piece of knowledge. Facts

[3] This definition is clearly broader in nature than the definition used in the PAIS Action Plan in the Construction and Infrastructure Sector [Dutch: 'Plan van Aanpak PAIS in de bouw en infra']. The authors of that report limit themselves in advance to defining and/or explanatory libraries containing information on *physical* objects.

are usually grouped in collections of facts about particular kinds of things (objects). Such a collection of expressions of facts together forms a model, which expresses knowledge about the kind of things. This is illustrated in Figure 5.

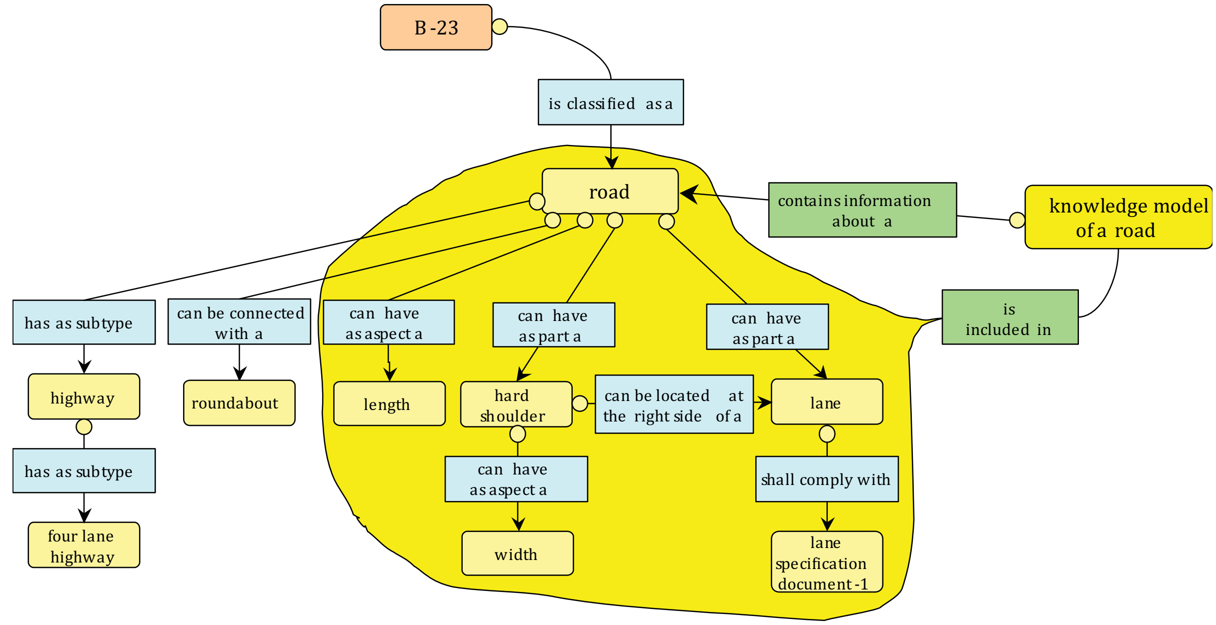


Figure 5 — Example of a knowledge model about a road

Figure 5 shows that there is a distinction between a concept (road) and a knowledge model about that concept (a road model). Furthermore, it also illustrates that the knowledge model is a collection of facts that is composed of a number of specific atomic facts, such as the fact that a lane can be part of a road. Each of those facts is expressed by a relation of a particular type. The differences between various knowledge libraries are highly due to the fact that those relation types are often not made explicit and that there is little standardization of those relation types. This is the reason why this standard contains definitions of standard relation types.

If concepts are arranged in a subtype-supertype hierarchy of concepts, then this is called a taxonomy. Knowledge libraries may go further than a taxonomy, when they also define concepts by including modelled knowledge about those concepts. This implies that a knowledge library includes relations between concepts, whereas each of such a relation expresses a fact about the related concepts and thus expresses some knowledge. A number of such relations with the same concept together form a network, which expresses knowledge about the concept and is called a knowledge model about the concept. This is why we define a knowledge library as a library of knowledge models.

To cater for the differences in functionality between different knowledge libraries, this standard defines a number of 'functional units' each of which covers a certain scope of functionality. The scope of a knowledge library may cover one or more functional units and its compliance with this standard is then compared with the guidelines for those functional units only.

In 6.2.1, a summary is given of various examples of kinds of knowledge libraries. The examples are by no means exhaustive, but are rather used for illustration purposes.

6.2.1 Product catalogues

Product catalogues are object libraries that are created for specific applications within a specific area. Examples of applications are: sales, purchasing, engineering, drawing and maintenance. Examples of domains are: construction, installation, machine building, process industry and shipbuilding. So far, most catalogues are used to support the selection and purchase of items. For this reason, such object libraries are called *product catalogues* and will usually contain, for a specific discipline, thousands of items from one or more suppliers such that a purchasing department can select and order. In the case of a light bulb, for instance, it would be possible to check which suppliers can provide it and also what the prices and delivery periods are. The models that are used in product catalogues usually include only the information that is needed for the selection and purchase of products. Typically those models do not include for example assembly structures.

NOTE Examples of such a product catalogue object library are the UNETO-VNI discipline-related product catalogue and the MESC catalogue of buying specifications at Shell.

6.2.2 Product data requirements libraries

In order to properly fill such a product catalogue, it is necessary to specify information requirements for all the kinds of products or items that will be included in the catalogue. This can be done by creating an object library of requirements models. Such requirements models may be presented on standard forms and are therefore often referred to as *standard specification sheets* (also called *data sheets* or *templates)*. To continue using the previous example, a requirements model (on a standard specification sheet) of a light bulb contains precisely which aspects may be or shall be 'filled in' by a supplier. A requirements model library, therefore, effectively forms the basis for filling a discipline-related parts library or product catalogue (see Figure 6).

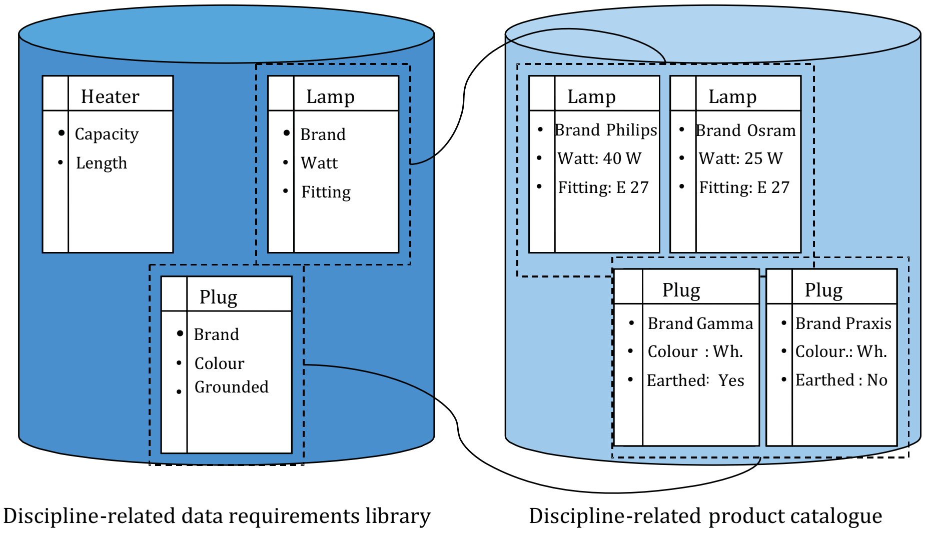


Figure 6 — Relation between discipline-related data requirements library and a product catalogue

Therefore, a collection of various requirements models for types of products in itself makes up a template library. In actual practice, also the terms *item classes* or *product classes* are used for these requirements models. Examples of data requirements libraries are the library of MESC templates (called 'formats') and the UNETO-VNI item class library in which approximately 3 500 templates have been defined. The above-mentioned MESC and UNETO-VNI product catalogues are filled according to their respective templates.

6.2.3 Project libraries

From an architectural point of view, project libraries are very similar to product catalogues. The main difference is of a more functional nature, in that a project library focuses on product models that include the composition structure of the products and is mainly applied in phases where individual things are created on the basis of knowledge about various design alternatives. A project library is therefore filled with knowledge about kinds of objects of which the organization in question wishes to create parts – the so-called 'making' parts. These making parts may in turn consist of an assembly of 'purchase parts' that have been specified in a product catalogue described earlier. Consequently the focus of a project library is on the creation stage and therefore also on the information that is needed for the assembly of products (so-called design knowledge). An example of such a library is the VNI parts library.

6.2.4 Smart dictionaries

The fourth library type, smart dictionaries, are integrating libraries, that are used mainly to provide a common terminology and language for various systems and therefore contain common definitions and have an explanatory purpose. Such libraries focus primarily on an unambiguous description of the 'meaning' of objects. They consist of models that include definitions, defining aspects and (specialization) relations between objects, such that used kinds of objects are defined unambiguously. The integrating function comes about when such libraries are commonly used by the discipline libraries mentioned earlier (see Figure 7). These integrating libraries can be distinguished in text oriented dictionaries and explicit model based dictionaries where even the definitions are expressed as models. Examples of such smart dictionaries are contained in LexiCon, the Gellish English Dictionary-Taxonomy and IFD Library Group.

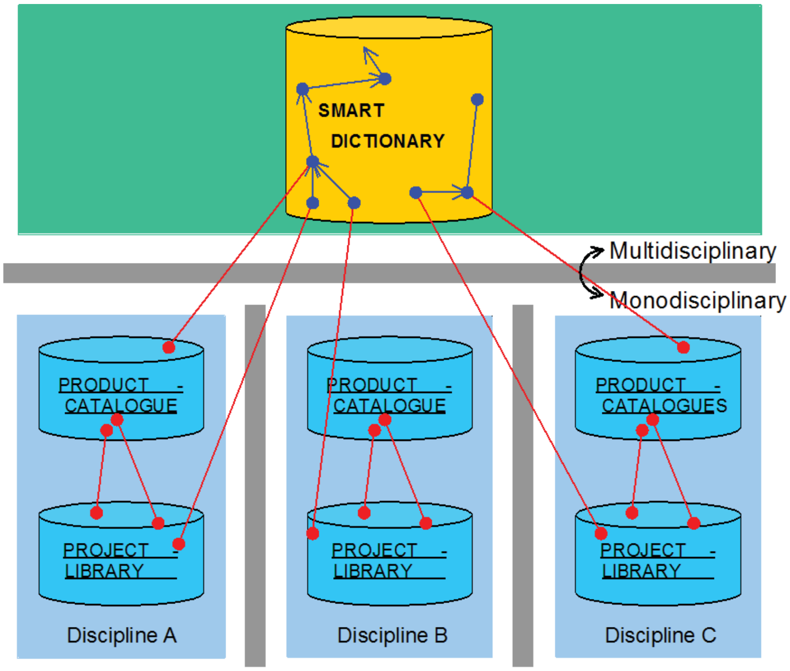


Figure 7 — Relation between discipline libraries and smart dictionaries

6.2.5 Classification systems

Classification systems have been created for the purpose of terminology harmonization and context description. They are characterized by the collection of objects based on a particular aspect in multiple and, at times, heterogeneous aspect layers. Generally speaking they are not object libraries.

Usually, this type of layer will apply a hierarchy of no more than three or four levels deep. The addition of 'context' to these libraries is mainly due to the fact that these systems have a supporting role. They are aimed at the classification or collection of kinds of objects for a particular application. Examples of classification systems include ICS, UBIM91, Elem91, SfB, Omniclass and DIN 6779-2.

6.3 Added value of knowledge libraries

6.3.1 Introduction

To some this may seem a trivial question, but why do we actually have knowledge libraries? The main purpose of this subclause is to present a number of striking examples of knowledge libraries.

6.3.2 Trade column integration

One of the most obvious examples is the integration in a trade column, also called a supply chain. Collection of item information on various suppliers in a single library may result in considerable benefits for the various parties concerned. It must be said that these advantages may only be achieved after an initial investment in setting up a library with standard specification sheets for items. Branch organizations in particular, as interest groups, are making a case for playing a supporting role in this connection.

NOTE During a presentation at the IGBI 2002, the purchasing manager of Ballast Nedam showed that on an annual basis, the company is dealing with 18 000 suppliers, 325 000 invoices and administrative costs of €24 million. These figures indicate that there is enormous potential for standardization by means of knowledge libraries in order to achieve efficiency benefits.

6.3.3 Harmonization among disciplines and parties

Knowledge libraries also form a means for improving co-operation between parties that will often have different expertise and background knowledge. Each discipline frequently uses its own vocabulary and means of interpretation. However, nowadays most projects tend to transcend the boundaries of disciplines, so that parties using different vocabularies must still jointly produce an overall product. This often results in a host of harmonization issues, since certain matters may often be interpreted in totally different ways (see Figure 8).

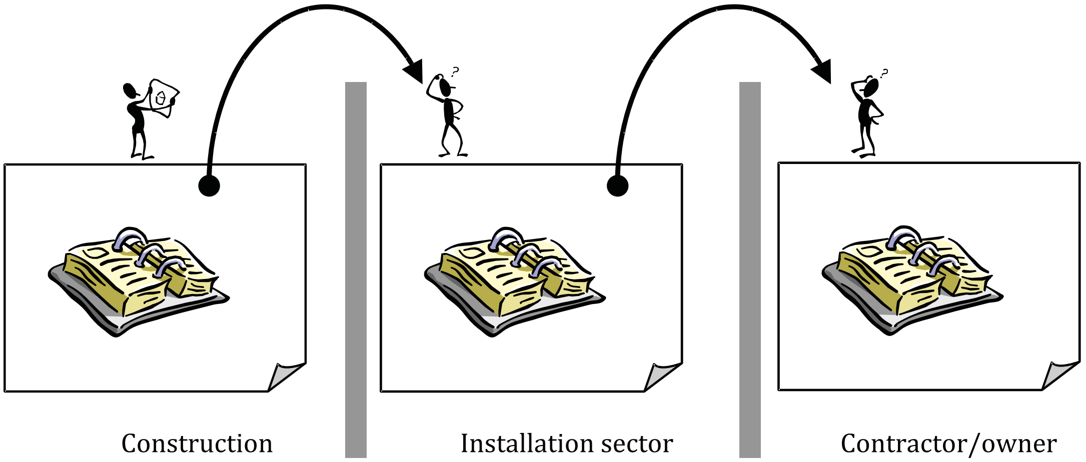


Figure 8 — Harmonization between different parties – supply chain integration

In this connection, knowledge libraries may greatly improve matters. By defining objects in a uniform way across the various disciplines and parties, it becomes possible to link the various worlds. The smart dictionary as discussed in 6.2.4, in particular, may serve this purpose. This has been further illustrated in Figure 9.

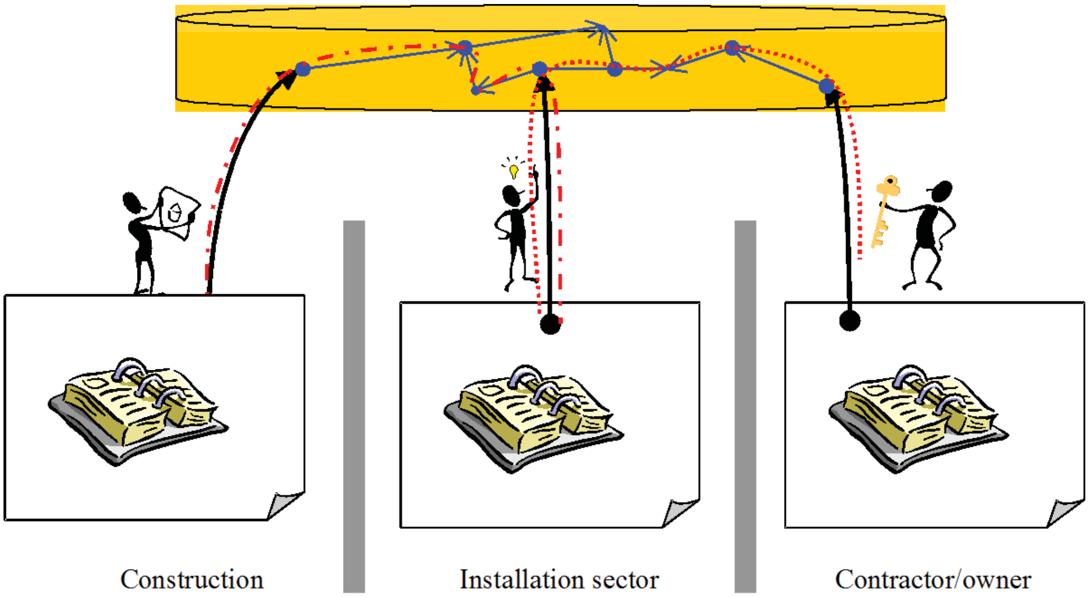


Figure 9 — Smart Dictionary

6.3.4 Harmonization in time – lifespan integration

Products have a lifespan that begins with a product specification (conceptual design, specification) and ends in demolition. In many cases, various parties will be involved during the various stages of the lifespan of a product. In analogy with the supply chain integration, knowledge libraries may offer considerable added value here as well. How often does it not happen that a completely different specification and naming convention is used for the desired products in the specification as opposed to the design stage? Integrating libraries may bridge this gap.

6.3.5 Integration of applications

Applications involving ‘product engineering’ are often characterized by ‘automation islands’. Such applications will not recognize each other’s language, and the information that is generated by the various applications is only accessible by themselves. But do a calculation program and a CAD program, for example, actually make use of intrinsically dissimilar information? In most cases objects will be concerned - objects being computed or positioned within a particular space. So, if information about the objects is provided according to common terminology that is sufficient for calculation purposes as well as includes topological information, then the way is paved for integration of product information.

6.3.6 Reuse of knowledge

Many engineers consider each project to be unique. In many cases, the product that they have designed cannot be applied to other projects on a one-on-one basis. This assumption is one of the reasons why little attention is being paid to standardization and a uniform use of concepts within projects. This means that the design of a pump installation will be called ‘pump installation’ in project A, ‘pump system’ in project B, ‘pump assembly’ in project C and simply ‘pump’ in project D. Clearly this complicates the reuse of design knowledge. It is simply difficult to disclose the knowledge because a uniform definition is lacking.

By defining a pump installation in a knowledge library once and then continuing to refer to this concept in that library, the threshold for disclosing this knowledge is considerably lowered.

6.3.7 Innovation support

Certain types of knowledge libraries focus on the unique positioning of various kinds of objects, including their relations. From these relations it may be gathered, for example, that a panel radiator is a special type of radiator, which in turn is a specialization of a heat exchanger.

A designer who intends to introduce a certain amount of heat in a room may decide to install a panel radiator in the room and may use the above knowledge library as an easy way of finding out what kinds of objects share more or less the same aspects. He[4] may find that a tubular unit radiator could be used as an alternative for a panel radiator. On a more generic level, he might even find that floor radiation heating is also among the possibilities (as an alternative to a radiator).

In other words, knowledge libraries provide an excellent search mechanism for finding alternative solutions, or for finding a very specific kind of solution.

7 Functional framework for knowledge libraries

7.1 Introduction

In Clause 6 a number of library types and their added value were presented. Although they probably suffice to provide a general picture, thinking in terms of 'library types' does not provide a solid basis for formulating guidelines. The major drawback is the fact that the various libraries have different scopes and ambitions. Existing libraries and the information models according to which they are structured cannot be unambiguously classified as a certain type, but will often harbor multiple functions (hybrids). Moreover, the extent to which the various types of functions are made operational will also differ with each knowledge library. As a result of this ambiguity, misconceptions will arise and the differences in opinion and interpretation may tend to seem bigger than they actually are.

This is why it has been decided to give the various possible *functions* of a library the central focus. Functional units are defined allowing a library to generate its content in an exchangeable way. For each functional unit only a subset of the guidelines is applicable. Therefore, this standard presents the guidelines per functional unit. It also specifies which concepts and relation types are essential for a functional unit.

In total, six distinctly different approaches have been identified and are discussed below. The approaches are indicated in this standard as *functional units*. Information according to such *a functional unit might be generated as* a subset of the information that is available in a knowledge library.

The following functional units are defined:

a) Vocabulary, being basically a list of terms;

b) Dictionary;

c) Taxonomy;

d) Aspect Models;

e) Composition Models;

f) Collection models.

Specific types of things and types of relations typically belong to a particular functional unit and some relations are typically used between or outside these functional units.

[4] In this standard, the form of 'he' is often used. In most cases, 'he' may be replaced by 'she'.

7.2 Functional unit: vocabulary or list of terms

The first function that a knowledge library may have is to generate names of concepts. Those names are usually arranged as a list of terms, also called a Vocabulary, Glossary or Nomenclature. This functional unit therefore deals with the way in which words or phrases are written in a particular language. In other words it deals with the *notation* of the words, in some cases compound terms or phrases, that are used to refer to concepts. In this functional unit the syntax is specified for the terms used.

The terms in a vocabulary do not have relations to each other, although they are usually presented in an alphabetically ordered list. This is illustrated in Figure 10 — Functional unit: Vocabulary, in which the various terms are symbolically represented by dots.

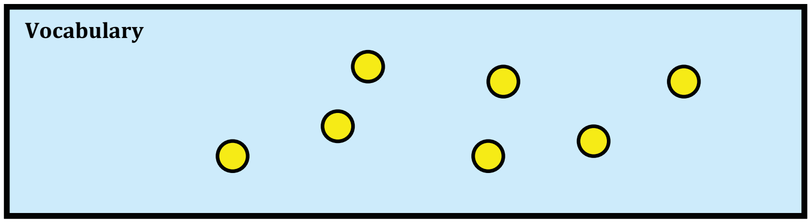


Figure 10 — Functional unit: Vocabulary

The simplicity of this functional unit in no way implies a reduction of added value. Maintaining a vocabulary in which all terms have been allocated that are used within a certain context (e.g. the installation sector) can often contribute to the elimination of harmonization issues. If a Vocabulary is maintained for a particular purpose, then it is usually called a Controlled Vocabulary.

7.3 Functional unit: Dictionary

The second function that a knowledge library may have is to generate *definitions of concepts*. In a dictionary the concepts are usually represented (denoted) by (compound) names or terms and sometimes phrases, whereas each name or term or phrase is explained by a description that defines what the concept is. This is represented symbolically in Figure 11 — Functional unit: Dictionary.

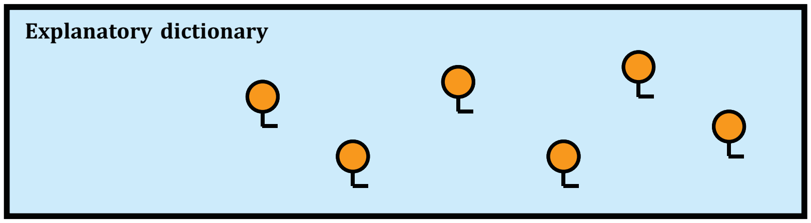


Figure 11 — Functional unit: Dictionary

Definitions always include terms which are references to (or denotations of) other concepts, which may be also included in the dictionary. These references imply relations of particular types between the defined concept and the concept that has a role in the definition. Those relations may be informally represented in natural language text to make textual definitions. Those relations may also be expressed formally in the form of Definition Models, which include explicit relations between the defined concepts and the defining concepts. Definition Models may also include relations to illustrations in the form of pictures or sound (files).

7.4 Functional unit: Taxonomy

In the third functional unit, taxonomy, the functionality does not so much concern the way of notation or a textual definition of the various concepts, but rather *model defining information about the concepts. The prime defining relation between concepts is the specialization relation, which is a relation between* *two concepts, one of which is the subtype concept and the other is the supertype concept.* A taxonomy is essentially a subtype-supertype hierarchy of concepts that is composed of multiple specialization relations. S*ubtype-supertype relations* define which subtypes can be distinguished for a particular supertype, on the basis that a subtype is defined by more constraints than its supertype. So a supertype is more generic than its subtypes.

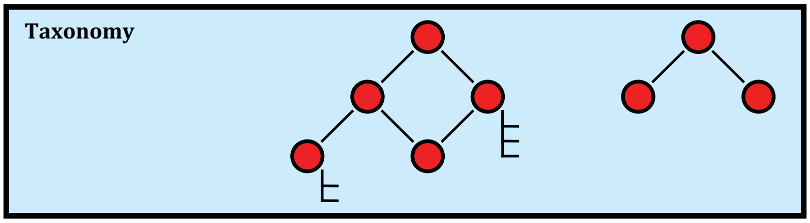


Figure 12 — Functional unit: Taxonomy

For instance, aspects may be recorded that typify an object (see the 'rakes' in Figure 12 — Functional unit: Taxonomy).

Other defining (discriminating) relations are relations with discriminating aspects and discriminating compositions. When defining aspects or parts are linked to a concept, then the model will inform users about what distinguishes a particular concept from the other concepts. For instance, a ball bearing has as discriminating parts balls as rolling element, whereas a ball has as discriminating aspect its shape, to which the value 'spherical' is assigned. On the other hand, a roller bearing (in general), being the supertype of ball bearing, has a rolling element that can have any kind of shape. This illustrates that supertype concepts have less constraints than subtype concepts.

Note that there are basically two different hierarchies: a *specialization hierarchy* and a *composition hierarchy*. One is based on the specialization relation (also called a subtype-supertype relation), the other is based on the composition relation (also called a part-whole relation). These two hierarchies should be clearly distinguished. It should be avoided that a component is defined as a subtype of the assembly of which it is a part. For example, in a specialization hierarchy it can be specified that ball bearing is a subtype of bearing and that a ball bearing has by definition as part a ball. However, the latter relation is not a specialization relation, but a composition relation that belongs to a composition hierarchy. The whole composition hierarchy may specify the decomposition hierarchy of, for example, a compressor. In this hierarchy it is also specified that a compressor may have as part a ball bearing. This functional unit (taxonomy) primarily deals with specialization relations (and may include definitions that specify obligatory components). This functional unit does not include compositions. They are dealt with in functional unit Composition Models.

Taxonomies can be distinguished in taxonomies with free text definitions and taxonomies with modelled definitions (or a combination of both). Free definitions describe the discriminating aspects and relations in natural language sentences. Modelled definitions treat discriminating aspects and related objects as separate concepts and explicitly define the types of relations with those concepts. This means that modelled definitions are textual definitions that are converted into knowledge models. Modelled definitions express meaning in the form of the various relations between concepts. These relations describe in which way a certain object relates to another object.

Taxonomies can be structured as a single parent hierarchy or as multi-parent hierarchy. In single parent hierarchy (often called a tree structure) each concept has only one supertype. In a multi-parent hierarchy each concept may have more than one supertype.

From a functional viewpoint, a taxonomy particularly offers a basis for parties to reach a common definition of the meaning of objects (see Figure 7 — Relation between discipline libraries and smart dictionaries). This functional unit may also be used as a search mechanism (possibly for innovation purposes).

NOTE A thesaurus has similarity with a taxonomy, but the relation types of the relations between the concepts in a thesaurus are different. A conventional thesaurus typically uses relation types to relate a term to a wider term a narrower term or to a related term. An advanced thesaurus could use the more explicit relation types such as those for a taxonomy and for composition as defined in the following functional units.

7.5 Functional unit: Aspect Models

The fourth functional unit consists of Aspect Models that include *aspects of concepts without considering the composing parts as separate objects* (see Figure 13 — Functional unit: Aspect Models).

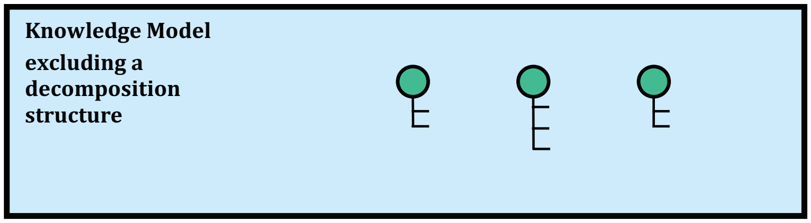


Figure 13 — Functional unit: Aspect Models

This functional unit includes aspects that describe types of objects in detail. The models may include all the aspect values that the types of objects have or they may include only aspects that are requirements from the perspective of a certain context (e.g. the installation sector). Models according to this functional unit are currently applied, for example, for design libraries and product catalogues in the field of procurement (trade), as well as in requirements models, for example with regard to calculations (calculation standard specification sheets).

Specifications of data requirements for product catalogues are typical examples of areas in which the Aspect Models functional unit plays a dominant role. The main difference between specifications of data requirements and catalogues is that in catalogues the requirements templates (standard specification sheets) that have been drawn up are filled-in with values by the suppliers and are disseminated as such. For this reason, catalogues are usually also Aspect Model-oriented.

Note that it is not a necessity to exclude a composition structure in models for procurement. On the contrary, from the perspective of business and data integration it is recommended to include assembly structures in the description of products. The composition is many times discriminating. So, the functional unit Aspect Models does not specify that the concepts are not defined by parts or do not have parts, but the Aspect Models functional unit is meant for a way of describing concepts such that the parts are not distinguished as separate concepts in the model.

7.6 Functional unit: Composition Models

The fifth functional unit, Composition Models, focuses on the *specific composition of objects as an assembly of objects* in addition to aspects of the assemblies as well as of the components (see Figure 14 — Functional unit: Composition Models).

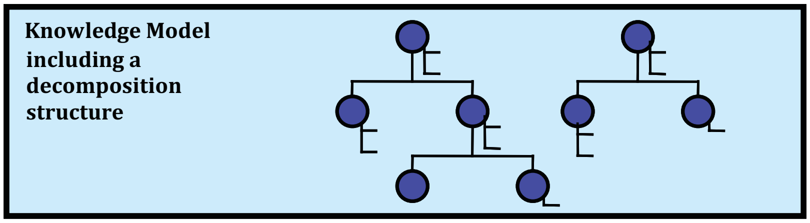


Figure 14 — Functional unit: Composition Models

This unit is an extension of the Aspect Models functional unit. The main difference between them is the fact that Composition Models also support the modelling of product configurations. This means that knowledge models that include a composition structure 'zoom in' on the products or objects by specifying how they can be created (usually assembled) from their components and what the aspects of their components are. In the Aspect Models functional unit a kind of object is specified as if a kind of object is not composed of parts, although they still may have parts, but then those parts are not mentioned explicitly in the model. Therefore, in such Aspect Models all the aspects of the parts are specified as if they are aspects of the whole (typically defined by 'intrinsic aspects'). This is usually sufficient in product models used for procurement only. If the same kind of object is described as a Composition Model, then it is specified that such a kind of object is or may be made up of various parts. This has as consequence that most aspects will be specified as being aspects of its parts. By modelling the aspects of the parts it becomes possible to integrate the supply chain businesses, where information for procurement of assemblies is integrated with information for the manufacturing, supply and procurement of the separate parts.

Apart from composition relations, a Composition Model may also include connection relations that specify which and how components are connected.

Note that this is a recursive process as the (main) parts of a kind of object are concepts themselves that may in turn consist of further parts (resulting in a 'break-down structure'). So this functional unit implies references between concepts by means of part-whole relations, while relating aspects to the proper parts.

This functional unit also covers the specification of the minimum and maximum number of components that may occur simultaneously in an assembly and whether a component can be part of zero, one or more assemblies at the same time. These constraints are called cardinality constraints.

This functional unit is widely used in cases where the integration of data from various sources or units is required, such as integration across an industrial supply chain. In this way, the function of this functional unit differs from that of the Aspect Models functional unit. An example of the application of this functional unit is case in which the producer of an armature wants to describe the total variation of his kinds of armatures in a unique way in knowledge models that include a composition structure, in order to be able to effectively configure armatures for various customers. On the other hand, a buyer of armatures will usually be satisfied with a description that conforms to an Aspect Model, as he is often only interested in an armature as a whole. However, a buyer who is involved with maintenance of large quantities of armatures and does hold spare parts will be interested in a Composition Model in order to search for parts, based on their properties. Actually, it is nevertheless quite likely that particular parts of the armature will have been specified in such a way that they conform to the Aspect Models functional unit.

NOTE Models that use more relation types than those that are used in a Taxonomy are often called *Ontologies*. In that terminology an Aspect Model is a constrained ontology. Composition Models, which include Aspect Models, that include more kinds of relations, are called full ontologies.

7.7 Functional unit: Collections

The last functional unit, Collections, mainly serves the *grouping of objects that appear in other functional units.* For instance, objects from the previous functional units may be collected per discipline in which they are primarily applied, and collections of expressions of facts that are collected according to their domain of knowledge. Moreover, groups of aspects of object may be defined, thus enabling to define ‘views’ of parts of the object information (see Figure 15 — Functional unit: Collection).

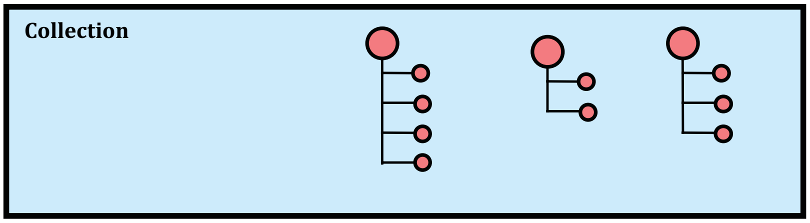


Figure 15 — Functional unit: Collection

This function unit deals with collections as well as with collection relations. A collection relation is a relation that relates an element to a collection.

As discussed in 6.2.4, Smart Dictionaries are geared mainly towards the performance of this function.

7.8 Relations between functional units

Functional units do relate to each other. Let us consider a random standard specification sheet to illustrate this. Taking into consideration the various functional units, it will be clear that the main focus of these data requirements models lies with the functional unit Aspect Models. After all, the standard specification sheets are meant to support the industry column through provision of a template to create item data. Still, we would be overlooking the potential of standard specification sheets if we say that only Aspect Model functionality has been used. With every standard specification sheet there are synonyms included (although also related terms are referred to as synonyms). The various terms are thus projected onto the vocabulary. In addition, the standard specification sheets have been collected in groups, so that a relation with the collection has been achieved. This is illustrated in Figure 16 — Mapping a data requirements library onto the framework.

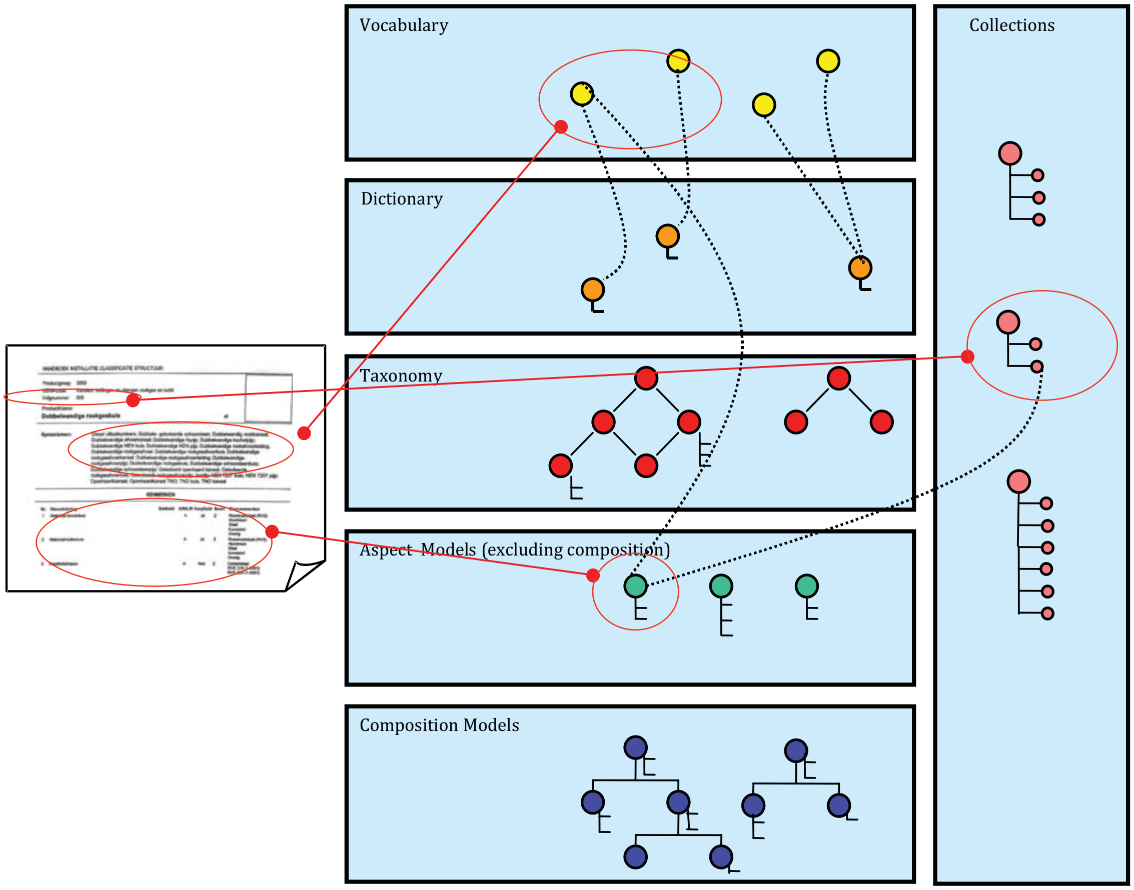


Figure 16 — Mapping a data requirements library onto the framework

The figure illustrates a data requirements library that does not use a taxonomy for the creation of the specifications. Apparently their knowledge library does not provide for this.

8 Guidelines for functional units

This clause contains the guidelines of this standard. They make up the normative agreements of this document. In 8.1 to 8.7, the various guidelines are discussed per functional unit. Each subclause starts with a list of the concepts and kinds of relations that are applicable for the functional unit. In Annex D a reference model is included in which the various object types and relation types are presented in their correlation.

8.1 Guidelines for Vocabularies or List of Terms

8.1.1 Introduction

The first functional unit that will be specified concerns the vocabulary. As described earlier, a vocabulary mainly services a syntactical purpose. It also forms a basis for linguistic matters.

8.1.1.1 Applicable Concepts

Id	Name	Role
V1.1	Term	The uniform recording of the syntax of the (possible) names and abbreviations of the concepts. The terms are allocated to the concepts in the context for which a particular vocabulary is meant. Typically as used in a context in which homonyms are avoided and synonyms and abbreviations are explicitly mentioned as such.
V1.2	Unique Identifier	A whole number that uniquely identifies and represents an object or fact (relation).
V1.3	Language	A natural language in which the term is expressed.
V1.4	Language community	A language community or discipline (within a language) in which the term is primarily used as a name or abbreviation to refer to the concept represented by the unique identifier.

8.1.1.2 Applicable Relation Types

Id	Name	Role
V2.1	Naming relation	Relates a unique identifier for an object to a term that is a name or abbreviation in a particular language, language community and expressed in a particular character set.

8.1.2 Guideline for terms

Guideline for terms A term is a character string of which the notation must be specified syntactically in an explicitly specified character set.	
Id: V3	
Explanation: it is important for vocabularies that the names (terms) of concepts are specified in an explicitly defined character set to enable computerized comparison.	**Example:** the term deg C can be displayed in the ASCII character set. The term °C requires a special character set because it cannot be displayed in ASCII.
Check: -	

8.1.3 Guideline for term singular

Guideline for term singular A base term must contain at least a non-abbreviated singular noun, unless this conflicts with the general use of the noun in question or unless the term refers to a collection.	
Id: V4	
Explanation: A concept shall be denoted by a singular term unless it is a collection. In various libraries the plural form is used, which may result in some confusion/ambiguity. Synonyms (including symbols) may include abbreviations. NOTE it is recommended to use natural language word sequences.	**Example:** ‘kettle’ instead of ‘kettles’ and ‘nominal diameter’ instead of ‘nom. diameter’. Millimetre is a base term. The term mm is a symbol (synonym). The term PC is a correct abbreviation (synonym).
Check: normal spelling rules.	

8.1.4 Guideline for term characters

<table>
<tr><td colspan="2"><b>Guideline for term characters</b>
A concept shall be at least denoted by a term that consists of one or more of the following ISO/IEC 8859-1 characters: digits, small and lower capitals, spaces, “ ‘ “, “(“, “)”, “,”, “.”, “/”. Synonyms may also consist of characters in other character sets, provided that the character set shall be specified.</td></tr>
<tr><td>Id: V5</td><td></td></tr>
<tr><td><b>Explanation:</b> The above character set is a common basis for harmonization. Other characters often have special functions or cannot be displayed on some devices. Therefore concepts shall at least have a name in the above-mentioned character set. If also another character set is used this shall be explicitly specified.</td><td><b>Example:</b> No underscores shall be used in terms, so: ‘PLC unit’ instead of ‘PLC_unit’. The unit of measure for resistance (‘Ω’) shall at least have as synonym a name in the above-mentioned character set, such as ‘ohm’.</td></tr>
<tr><td colspan="2"><b>Check:</b> Search for characters that are not included in the above-mentioned standard.</td></tr>
</table>

8.1.5 Guideline for term formatting

<table>
<tr><td colspan="2"><b>Guideline for term formatting</b>
The name of a term shall not contain formatting, such as underlined characters or characters in bold or italics.</td></tr>
<tr><td><b>Id:</b> V6</td><td></td></tr>
<tr><td><b>Explanation:</b> the exchange of these names is made considerably more complex when such a layout is added and needs to be maintained.</td><td><b>Example:</b> ‘pump’ instead of ‘pump’.</td></tr>
<tr><td colspan="2"><b>Check:</b> normal spelling rules.</td></tr>
</table>

8.1.6 Guideline for unique identification

<table>
<tr><td colspan="2"><b>Guideline for unique identification</b>
Every concept or fact shall be indicated by its own unique identifier. A unique identifier shall consist of a whole number or a character string that satisfies specified rules and that refers unambiguously to one concept, individual thing or fact and shall not refer to any other concept, individual thing or fact.</td></tr>
<tr><td><b>Id:</b> V7</td><td></td></tr>
<tr><td><b>Explanation:</b> the integration of information requires that computers can uniquely refer to everything, without the risk that the reference changes when information about the object changes and without the need to verify a context in which a term is used. This requires a meaningless unique identifier that may not be a concatenated code (in which information is encoded) and that is independent of any natural language. The identifier shall uniquely denote a concept, individual thing or fact in a specified ‘world’. Integration of concepts in different ‘worlds’ implies a mapping of the sets of unique identifiers.
It is recommended to join the identification conventions of an existing ‘world’ above creating a new ‘world’.</td><td><b>Example:</b> UID 130206 is an example of a unique identifier of the concept in a particular ‘world’ that is referred to by rotating equipment engineers in English as ‘pump’. The same concept is referred to in German as ‘Pumpe’ and in Dutch as ‘pomp’. In other ‘worlds’ the same concept may have other unique identifiers.</td></tr>
<tr><td colspan="2"><b>Check:</b> presence of language independent non-coded unique identifiers.</td></tr>
</table>

8.2 Guidelines for Dictionaries

8.2.1 Introduction

The second functional unit that will be specified concerns the dictionary. The dictionary mainly serves a defining purpose.

8.2.1.1 Applicable Concepts

Id	Name	Role
D1.1	Concept	Describing (lexically) the meaning of concepts. NOTE If multiple terms (synonyms) can be used to denote a concept, more terms will have been included in the Vocabulary. Relations for indicating this are described in **8.7**.
D1.2	Aspect	Describing (lexically) the meaning of aspects. NOTE If multiple terms (synonyms) can be used to denote an aspect, more terms will have been included in the Vocabulary. Relations for indicating this are described in **8.7**.

8.2.1.2 Applicable Relation Types

Id	Name	Role
D2.1	Specialization relation	Creates a hierarchical network between concepts with which an important part of the semantics is realized.
D2.2	Definition relation	Relates a concept (represented by its unique identifier) to a character string that is a lexical definition of the concept.
D2.3	Naming relation	Relates a unique identifier for an object to a term that is a name or abbreviation in a particular language, language community and expressed in a particular character set.

8.2.2 Guideline for usage of descriptive terms

Guideline for usage of descriptive terms *The terms in a description of a concept that refer to other existing concepts shall occur themselves as separate terms in the Dictionary.*	
Id: D4	
Explanation: this guideline must safeguard the consistency of words used in the definition if they are also known as physical objects or aspects.	**Example:** for a physical object with the description 'A centrifugal pump is a pump that …' the term pump must occur as a separate entry in the Dictionary.
Check: normal spelling rules.	

8.2.3 Guideline for lexical definition

Guideline for lexical definition *The free text description of the meaning of a concept must start with a term that refers to the direct supertype of the described concept followed by a description of the discriminators.*	
Id: D5	
Explanation: A proper definition defines a concept as a subtype of its direct supertype and further defines the discriminating aspects and relations that distinguish the subtype from that supertype and from its neighbouring subtypes. This guideline reflects the specialization relation and the discriminating relations and is their textual equivalent. Note that ideally this lexical definition can be automatically generated from an explicitly modelled definition.	**Example:** the specialization of a solid sphere can be based on its shape aspect as follows: solid sphere is a kind of solid item that has a shape that is spherical.
Check: this guideline may in fact only be used for verification purposes if the corresponding taxonomy unit is provided as well. With the help of specialization rules, it may then be verified whether this requirement has been met.	

8.3 Guidelines for Taxonomies

8.3.1 Introduction

The taxonomy is the third unit for which guidelines will be provided. It concerns the arrangement of objects in a subtype-supertype hierarchy.

8.3.1.1 Applicable Concepts

Id	Name	Role
T1.1	Concept	A concept represents a particular kind of thing. It is the basic element for recording the meaning or semantics of things that are classified by the concept.
T1.2	Aspect	Aspects are arranged in a subtype-supertype hierarchy in a taxonomy. Subtype aspects may inherit from their supertypes, e.g. which scales are applicable for their quantification. A taxonomy may also include qualitative aspects that are qualifications or quantifications of kinds of aspects.
T1.3	Role	Roles can also be arranged in a subtype-supertype hierarchy. They are used in relations and are indirect references to role players.

8.3.1.2 Applicable Relation Types

Id	Name	Role
T2.1	Specialization relation	Creates a hierarchical network between objects with which an important part of the semantics is realized.
T2.2	Physical object - aspect relation	Relates (discriminating) aspects to physical objects that possess them.
T2.3	Composition relation	Relates characterizing part physical objects to the whole physical object in question so that the meaning/semantics of the whole is further specified.
T2.4	Aspect - scale relation	Relates aspects to scales (units of measure) according to which the extent or size of quantitative values of the aspects can be quantified by a number or range.

8.3.2 Guideline for flexibility of composition (cardinalities)

<table>
<tr><th colspan="4"><b>Guideline for flexibility of composition (cardinalities)</b>
<i>The composition relations in definitions may be of a typifying nature and then the number of components is not binding for individual things.</i></th></tr>
<tr><td colspan="2">Id: T3</td><td colspan="2"></td></tr>
<tr><td colspan="2"><b>Explanation:</b> if the definition of a certain (well formed) kind of physical object includes a specification that objects of this kind are composed of physical objects of other kinds, then this composition can be either binding or not binding for a particular instance. Technically speaking, this means that the minimum and maximum cardinality constraints of the composition relation may be unspecified, or the minimum cardinality may be zero.
NOTE It is recommended to make the minimum and maximum left-hand and right-hand cardinalities explicit (see cardinality guideline).</td><td colspan="2"><b>Example:</b> a bicycle is defined to have <i>normally</i> (typically) two wheels, but the cardinalities for the number of wheels are not specified. This means that a bicycle can have an unspecified number of wheels. If a bicycle is defined by the presence of at least one wheel and maximally three wheels, then the minimum cardinality is 1 and the maximum cardinality is 3. The opposite cardinalities will be 0 and 1, because a wheel can be part of no bicycle or of one bicycle (at the same time).</td></tr>
<tr><td>(0, 1) bicycle</td><td colspan="2">can have as part a</td><td>(1,3) wheel</td></tr>
<tr><td>bicycle</td><td colspan="2">can have as part a</td><td>wheel (unspecified cardinalities)</td></tr>
<tr><td colspan="4"><b>Check:</b> check whether a thing with a number of related instances that is less than the minimum cardinality or more than the maximum cardinality still is of the same kind and can still be well formed.</td></tr>
</table>

8.3.3 Guideline for inheritance through specialization

<table>
<tr><td colspan="4"><b>Guideline for inheritance through specialization</b>
<i>In a specialization relation, the subtype concept inherits all related aspects and any composition relations from the supertype concept.</i></td></tr>
<tr><td colspan="2">Id: T4</td><td colspan="2"></td></tr>
<tr><td colspan="2"><b>Explanation:</b> A subtype physical object is of the same nature as its supertype physical object and therefore it has by definition at least the same constraints as its supertype. Because the subtype is distinguished from the other subtypes of the same supertype, each subtype has at least one additional constraint (compared to the supertype) that is discriminating between the subtypes.</td><td colspan="2"><b>Example:</b> if the physical object ‘bicycle’ is a specialization of the physical object ‘means of transport’, ‘bicycle’ will have at least the same aspects as ‘means of transport’, e.g. the aspect that it can carry maximally a ‘design load’.</td></tr>
<tr><td>bicycle
M6 bolt</td><td colspan="2">is a specialization of
is a specialization of</td><td>means of transport
bolt</td></tr>
<tr><td colspan="4"><b>Check:</b> spot checks must be carried out for this purpose.
NOTE This is implied by a specialization relation anyway. It is added as a guideline in order to initiate quality checks on the specialization hierarchy.</td></tr>
</table>

8.3.4 Guideline for a single discriminator for physical object specialization

<table>
<tr><td colspan="4"><b>Guideline for a single discriminator for physical object specialization</b>
<i>The definition of a set of subtypes of a given supertype physical object shall be based on one or one set of discriminators.</i></td></tr>
<tr><td colspan="2">Id: T5</td><td colspan="2"></td></tr>
<tr><td colspan="2"><b>Explanation:</b> this guideline intends to minimize the specialization on the basis of multiple discriminators at the same time. It means that a number of subtypes shall be defined on the basis of different values for the same discriminator. In case of more than one discriminator, sets of subtypes may be specified, based on the various discriminators. Subtypes that are based on more than one discriminator typically require multiple stages of specialization, although no artificial intermediate concepts shall be created. This guideline does allow defining multiple supertypes of the same subtype object. Knowledge bases that do not allow for subtypes concepts to have multiple supertype concepts should classify one of the appropriate relations as a specialization relation and should classify the other(s) as a relation between kinds of things.</td><td colspan="2"><b>Example:</b> the concept pump has the two discriminating aspects: ‘mode of operation’ and ‘orientation of shaft’ (see the example above). This means that the concept pump should not be specialized in one step from ‘pump’ into ‘horizontal centrifugal pump’ when the concepts horizontal pump and centrifugal pump are well known concepts. In that case, the concept pump shall be specialized into ‘horizontal pump’ and ‘vertical pump’, and it shall also be specialized into centrifugal pump and reciprocating pump, whereas ‘horizontal centrifugal pump’ shall be defined as a next level subtype with two supertypes (‘horizontal pump’ and ‘centrifugal pump’).</td></tr>
<tr><td>centrifugal pump
horizontal pump
horizontal centrifugal pump
horizontal centrifugal pump</td><td colspan="2">is a specialization of
is a specialization of
is a specialization of
is a specialization of</td><td>pump
pump
centrifugal pump
horizontal pump</td></tr>
<tr><td colspan="4"><b>Check:</b> spot checks must be carried out for this purpose.</td></tr>
</table>

8.3.5 Guideline for discriminating description of specialization

<table>
<tr><td colspan="4"><b>Guideline for discriminating description of specialization</b>
<i>In case of specialization the definition of the subtype object shall specify which discriminator has been used for the specialization.</i></td></tr>
<tr><td colspan="2">Id: T6</td><td colspan="2"></td></tr>
<tr><td colspan="2"><b>Explanation:</b> it is not sufficient to discriminate on the basis of a single discriminator or set of discriminators; it should also be indicated <i>which</i> discriminator (aspect or part) is concerned. This can be done in the textual definition as free text or by explicit modelling of the relation(s) to the aspect value or part.</td><td colspan="2"><b>Explanation:</b> a ‘fired boiler’ and an ‘electric boiler’ are subtypes of the supertype ‘boiler’, where discrimination on the basis of the aspect ‘method of heat generation’ is involved. In a free text definition this can be done as follows: a fired boiler is a boiler of which the method of heat generation is ‘by fire’.</td></tr>
<tr><td>boiler</td><td colspan="2">can have as aspect a</td><td>method of heat generation</td></tr>
<tr><td>by fire</td><td colspan="2">is a qualification of</td><td>method of heat generation</td></tr>
<tr><td>fired boiler</td><td colspan="2">is a specialization of</td><td>boiler</td></tr>
<tr><td>fired boiler</td><td colspan="2">has by definition as qualitative aspect</td><td>by fire</td></tr>
<tr><td colspan="4"><b>Check:</b> verify whether the supertype has the discriminatory conceptual aspect or can have the kind of part and whether a value of this aspect or the kind of part has been defined to characterize the subtype.</td></tr>
</table>

8.3.6 Guideline for non-overlap of subtypes of a concept

<table>
<tr><td colspan="4"><b>Guideline for non-overlap of subtypes of a concept</b>
<i>The values of the discriminating aspects for different subtype concepts of the same supertype concept must not overlap when the subtyping is based on the same discriminator.</i></td></tr>
<tr><td colspan="2">Id: T7</td><td colspan="2"></td></tr>
<tr><td colspan="2"><b>Explanation</b>: By avoiding overlap in the description or aspects of subtypes, ambiguity in the classification of an instance by the possible subtypes is prevented. In other words, the subtypes that are based on the same discriminator shall have mutually exclusive values for the discriminator. Note that specialization on the basis of more than one discriminator is allowed.</td><td colspan="2"><b>Example:</b> the values ‘horizontal’ and ‘vertical’ for the aspect ‘shaft orientation’ of a pump are valid, so specialization on the basis of the shaft orientation aspect may occur.
The values ‘centrifugal’ and ‘displacing element’ for the aspect ‘operating principle’ of a pump are valid, so that specialization may occur also on the basis of the operating principle aspect.
The values ‘horizontal’ and ‘centrifugal’ are not mutually exclusive. They are not values of the same aspect. The aspect ‘type’ is therefore invalid as a basis for specializations to both subtypes. This means that two different discriminators are involved, which requires two sets of subtypes and a next level for the combined sub-subtypes.</td></tr>
<tr><td>horizontal pump</td><td colspan="2">has by definition as qualitative aspect a</td><td>horizontal shaft</td></tr>
<tr><td>horizontal shaft</td><td colspan="2">is a qualification of</td><td>shaft orientation</td></tr>
<tr><td colspan="4"><b>Check:</b> this is an arbitrary check. A tool may assist in comparison of kinds of aspects for kinds of physical objects at the same subtype level.</td></tr>
</table>

8.3.7 Guideline for consistency of discrimination and aspects

<table>
<tr><td colspan="2"><b>Guideline for consistency of discrimination and aspects</b>
<i>An aspect of a subtype may not overlap with the data element that has been included in order to discriminate the corresponding supertype.</i></td></tr>
<tr><td>Id: T8</td><td></td></tr>
<tr><td><b>Explanation:</b> a discriminator that is used for the specialization of a supertype into its subtypes shall not be reused for further specialization of the subtypes. This guideline safeguards that no aspects are added that are redundant to the discriminator on the level above them, since this may have a negative impact on the quality of the specialization structure.
If a further specialization is required on the basis of a further limitation of a value domain, then the lower level discriminator shall be distinguished from the higher-level discriminator.
If this tends to result in artificial discriminators, then this suggests that there is probably only one level of subtypes, instead of two.</td><td><b>Example:</b> if the physical object ‘pump’ is specialized into ‘centrifugal pump’ and ‘reciprocating pump’, one should not reuse the aspect ‘principle of operation’ for further specialization of these two subtypes, but all kinds of principle of operation shall be used to define subtypes on the same level.
If subtypes of bolt are defined on the basis of ‘size’ as a discriminator for example with the values: small, medium and large, resulting in subtypes small bolt, medium bolt and large bolt and when large bolt has second level subtypes medium large bolt and very large bolt, then the second level discriminator shall not be ‘size’ again. It shall be either largeness (degree of largeness) or the further subtypes should be introduced already on the first level.</td></tr>
<tr><td colspan="2">Check: -</td></tr>
</table>

8.3.8 Guideline for definition of roles of physical objects

<table>
<tr><td colspan="3"><b>Guideline for definition of roles of physical objects</b>
<i>A role of a physical object shall be defined to be a subtype of role and it shall be defined by which kind of physical object the role can be played.</i></td></tr>
<tr><td colspan="2">Id: T9</td><td></td></tr>
<tr><td colspan="2"><b>Explanation:</b> kinds of roles of physical objects (especially usages) are frequently used to refer to the kinds of things that can play roles of those kinds. Various kinds of roles of physical objects are often defined such that they are by definition played by a particular kind of thing. Those roles should be distinguished from the role players. This guideline enables to avoid that the aspects of physical objects are redefined (and are inconsistent) with the definition of the various possible roles of those physical objects.</td><td><b>Example:</b> ‘water pump’ is not a kind of pump, but it is a kind of role (because, out of use means losing its role: if the pump is put in a warehouse it is not a water pump anymore; this is the test to determine whether the concept is a role or not). This guideline specifies that It should therefore be specified that a ‘water pump’ is a subtype of role and that it ‘is by definition a role of a’ pump. (In other words: it is a role that is by definition played by a pump.)
NOTE It may also be specified that this kind of role implies that such a role is by definition played in a water pumping activity. But this extension is not required by this guideline.</td></tr>
<tr><td>water pump
water pump</td><td>is a specialization of
is by definition a role of a</td><td>usage
pump</td></tr>
<tr><td colspan="3"><b>Check:</b> whether each kind of role of a physical object is a subtype of role and is defined to have by definition a particular kind of role player.</td></tr>
</table>

8.3.9 Guideline for definition of intrinsic aspects

<table>
<tr><td colspan="4">**Guideline for definition of intrinsic aspects**
A kind of intrinsic aspect shall be defined to be a subtype of intrinsic aspect (being a role of an aspect) and it shall be defined for which kind of aspect it is a role and by which kind of physical object it is by definition possessed.</td></tr>
<tr><td colspan="2">**Id:** T10</td><td colspan="2"></td></tr>
<tr><td colspan="2">**Explanation:** aspects are phenomena that are defined independent of the physical objects that can possess them. An intrinsic aspect is a role of an aspect, being by definition possessed by a physical object. A kind of intrinsic aspect (such as ‘pipe diameter’) shall be defined by three facts: it shall be defined to be a subtype of intrinsic aspect. It shall also be specified of which aspect it is a role. Finally it shall be defined that it is by definition possessed by a particular kind of physical object.
The unit of measure of an intrinsic aspect is the unit of measure for the aspect of which it is a role. Therefore an aspect-scale relation need not be defined for an intrinsic aspect.</td><td colspan="2">**Example:** ‘pipe diameter’ shall be defined as a specialization of ‘intrinsic aspect’, which is a subtype of role. It shall also be specified that it is a role of a diameter. Furthermore, it shall be specified that it is by definition a role of a pipe.</td></tr>
<tr><td>pipe diameter</td><td colspan="2">is a specialization of</td><td>intrinsic aspect</td></tr>
<tr><td>pipe diameter</td><td colspan="2">is by definition a possible role of a</td><td>diameter</td></tr>
<tr><td>pipe diameter</td><td colspan="2">is by definition an intrinsic aspect of a</td><td>pipe</td></tr>
<tr><td colspan="4">**Check:** whether each kind of intrinsic aspect is defined as a subtype of intrinsic aspect and has a defined kind of aspect as role player and is defined as being by definition an intrinsic aspect of a kind of physical object.</td></tr>
</table>

8.3.10 Guideline for names of aspects

<table>
<tr><td colspan="2">**Guideline for names of aspects**
The name of an aspect must not contain a reference to its possessor physical objects.</td></tr>
<tr><td>**Id:** T11</td><td></td></tr>
<tr><td>**Explanation:** kinds of aspects shall be defined independent of the physical objects by which they are or can be possessed.
When a physical object possesses an aspect, then the aspect has a role in that relation of being possessed. That role carries the meaning that is determined by the usage context, being the possessor physical object. Especially aspects of parts shall either be defined as aspects that are possessed by the part or shall be defined as subtypes of ‘intrinsic aspect’ (see also guideline intrinsic aspect).</td><td>**Example: ‘**diameter’ is a proper kind of aspect, whereas diameter is defined independent of the kinds of things that can have a diameter.
‘Pipe diameter’ is not a pure aspect either, but it is an intrinsic aspect as it is by definition possessed by a pipe. ‘Impeller diameter’ is not an aspect of a centrifugal pump, but ‘diameter’ is an aspect of an impeller, which is a part of a centrifugal pump. Nevertheless ‘impeller diameter’ might be defined as an intrinsic aspect that is by definition possessed by an impeller and thus it can be an intrinsic aspect of a part of a centrifugal pump.</td></tr>
<tr><td colspan="2">**Check:** verify whether references to physical objects are found in the names of aspects.</td></tr>
</table>

8.3.11 Guideline for top of specialization hierarchy

<table>
<tr><td colspan="4"><b>Guideline for top of specialization hierarchy</b>
<i>A specialization hierarchy may have more than one top concept, but each top concept shall be a subtype of a concept in an integrating taxonomy.</i></td></tr>
<tr><td colspan="2"><b>Id:</b> T12</td><td colspan="2"></td></tr>
<tr><td colspan="2"><b>Explanation:</b> a specialization hierarchy (taxonomy) can be used to verify the semantic correctness of the usage of concepts in particular kinds of relations. Therefore it is necessary that each top-concept in a sub-hierarchy is a proper subtype of a concept in the upper taxonomy) that is used to define the relation types.
A top-concept in a sub-hierarchy should not be an artificial supertype that in fact represents a collection, because a collection is not a proper supertype of the elements in the collection.</td><td colspan="2"><b>Example:</b> if ‘civil item’ is a top-object in a knowledge library, then all its subtype shall be really subtypes of ‘civil item’ (by nature) and not just items (like bolts) that happen to be (also) used in the civil industry. Nevertheless, there is nothing against defining a collection, but a collection relation shall be used to specify the elements that belong to the collection. E.g. the collection ‘CROW objects’ can contain as element ‘civil items’, ‘piping components’, etc.</td></tr>
<tr><td>pipe
pipe</td><td colspan="2">can be an element of
can be an element of</td><td>piping components (collection)
civil items (collection)</td></tr>
<tr><td colspan="4"><b>Check:</b> every top-object shall be a (true) subtype of an object external to the library and internal to an upper taxonomy.</td></tr>
</table>

8.4 Guidelines for Aspect Models

8.4.1 Introduction

The fourth functional unit concerns the Aspect Models that exclude a composition structure. The focus in this functional unit is on the description of physical objects in order to make them suitable for use or creation in a certain application domain.

8.4.1.1 Applicable Concepts

Id	Name	Role
E1.1	Concept	Basic element for laying down application-oriented specifications.
E1.2	Aspect	Aspects in an Aspect Model specify the aspects for which values are required for the specification of a physical object for a particular application.
E1.3	Scale (unit of measure)	Scales and units of measure are used to specify according to which scale(s) an aspect can or shall be quantified.

8.4.1.2 Applicable Relation Types

Id	Name	Role
E2.1	Physical object-aspect relation	Links aspects to physical objects for the purpose of specification.
E2.2	Aspect-scale relation	Links aspects to units of measure to indicate the allowed or possible units.

8.4.2 Guideline for scales for aspects

Guideline for scales for aspects *For the specification of the quantitative value of an aspect, the scale or unit of measure must be specified on the basis of the naming conventions defined in* ISO 10303-41 *and defined within an explicitly specified character set.*	
Id: E3	
Explanation: all ISO TC 184/SC 4 norms make use of the same basic concept for modelling units of measure. Through compliance uniformity is safeguarded.	**Examples:** length scale and temperature scale are example of kinds of scales. The concepts kg, lb, m/s, mile/h and deg C are examples of qualitative scales that are usually called units of measure.
Check: -	

8.5 Guidelines for Composition Models

8.5.1 Introduction

The fifth functional unit concerns the knowledge models that include a composition structure. Functionally speaking, the added value is transferred to the description of variants of a possible composition of a physical object. The focus lies on the description of physical objects from the point of view of the designer. In the Composition Model functional unit, physical object models must be suitable for design processes.

8.5.1.1 Applicable Concepts

Id	Name	Role
I1.1	Physical object	Basic element for laying down (possible) compositions
I1.2	Aspect	Aspects in a Composition Model specify the aspects required for specifying a physical object for a particular application.
I1.3	Scale (unit of measure)	Scales and units of measure are used to specify according to which scale(s) an aspect can or shall be quantified.

8.5.1.2 Applicable Relation Types

ID	Name	Role
I2.1	Physical object-aspect relation	Links aspects to physical objects for specification purposes.
I2.2	Composition relation	This relation is mainly meant to describe the composition of physical objects (kinds of parts in kinds of physical objects), including also features, and accessories. From what parts is a well formed physical object normally made up? And from what parts are these parts made up in turn?
I2.3	Aspect-scale relation	Relates aspects to units of measure.

8.5.2 Guideline for scales for aspects

Guideline for scales for aspects *For the specification of the quantitative value of an aspect, the scale or unit of measure must be specified on the basis of the naming conventions defined in* ISO 10303-41 *and defined within an explicitly specified character set.*	
Id: I3	
Explanation: All ISO TC 184/SC 4 norms make use of the same basic concept for modelling units of measure. Through compliance uniformity is safeguarded.	**Examples:** length scale and temperature scale are example of kinds of scales. The concepts kg, lb, m/s, mile/h and deg C are examples of qualitative scales that are usually called units of measure.
Check: -	

8.5.3 Guideline for flexibility of physical object composition (cardinality guideline)

Guideline for flexibility of physical object composition (cardinality guideline) *The composition relations are not binding for instances of physical objects.*	
Id: I4	
Explanation: in this functional unit the focus will rather be on *possible* composition structures. The actual structure of an instance may actually deviate from this. Technically speaking, this means that the minimum and maximum cardinality constraints of a composition relation may be unspecified. The cardinalities may be specified such that they deviate from the cardinalities of a definition. For example for application in a particular context. This means that an individual thing may have a number of components that is outside the cardinality constraints, but is nevertheless within the definition of the concept that classifies the individual thing.	**Example:** if a cellar typically contains two pumps, it will also be possible to install a single pump assembly, according to the definition of a cellar. It may be stated that e.g. in the context of a particular manufacturer only two pump solutions are made, the minimum and maximum cardinalities are then both fixed to two in that context.
Check: Check whether cardinality constraints are within the cardinalities of the definition.	

8.5.4 Guideline for multiple wholes for a part physical object

<table>
<tr><td colspan="3"><b>Guideline for multiple wholes for a part physical object</b>
<i>A single physical object may form part of more than one encompassing physical object.</i></td></tr>
<tr><td colspan="2">Id: I5</td><td></td></tr>
<tr><td colspan="2"><b>Explanation:</b> through composition relations, part physical objects will be used for the definition or possible composition of higher assembly physical objects. A particular physical object can be part of more than one larger assembly. In that case composition relations are therefore made between one component physical object and various larger assembly physical objects.</td><td><b>Example:</b> a control valve can be part of a control loop and can also be part of a piping system.</td></tr>
<tr><td>control valve
control valve</td><td>can be part of a
can be part of a</td><td>control loop
piping system</td></tr>
<tr><td colspan="3"><b>Check:</b> this need not be checked.</td></tr>
</table>

8.5.5 Guideline for composition versus specialization of physical object

<table>
<tr><th colspan="3">Guideline for composition versus specialization of physical object
<i>A kind of part of a kind of physical object may not be also a subtype of that same kind of physical object.</i></th></tr>
<tr><td colspan="2">Id: I6</td><td></td></tr>
<tr><td colspan="2"><b>Explanation:</b> there are two different hierarchies of concepts: specialization hierarchies and composition hierarchies. These two hierarchies shall be strictly distinguished and shall be independent of each other. A decomposition hierarchy (asset hierarchy) describes how an assembly is or might be composed. This should not say anything about what subtypes of components do exist. Nevertheless, a knowledge model about a concept usually includes a composition hierarchy as well as specialization relations that indicate possible subtypes of the components.</td><td><b>Example:</b> bearing is a supertype of ‘magnetic bearing’ and ‘tilting pad bearing’ but a ‘magnetic bearing’ or a ‘tilting pad bearing’ is not a part of a bearing.
Tilting pad is not a subtype of bearing, but a subtype of pad and a part of a tilting pad bearing.</td></tr>
<tr><td>tilting pad bearing</td><td>is a specialization of</td><td>bearing</td></tr>
<tr><td>tilting pad</td><td>is a specialization of</td><td>pad</td></tr>
<tr><td>tilting pad</td><td>is by definition a possible part of a</td><td>tilting pad bearing</td></tr>
<tr><td colspan="3"><b>Check:</b> every concept that is a part shall also have a supertype that is not the same as the concept that is the whole.</td></tr>
</table>

8.6 Guidelines for Collections

8.6.1 Introduction

The last functional unit is collection. Collection is a unit that is mainly used to collect object types in collections. Typically this is done when only a limited number of subtypes may be selected in a specific context. For example, pick-lists are collections of allowed values.

8.6.1.1 Applicable Concepts

Id	Name	Role
C1	Collection	To specify that elements are brought together without being related to each other in a particular way.

8.6.1.2 Applicable Relation Types

Id	Name	Role
C2	Collection of concepts relation	To specify that concepts are elements of collections.

8.6.2 Guideline for membership of collections

Guideline for membership of collections *The membership of a collection shall not be solely based on intrinsic aspects of or a particular relation type for the elements, except for their location.*	
Id: C3	
Explanation: A query on any aspect or relation type will have a result that can be considered to be an ad hoc collection. Collections however should have a more permanent nature and their definition should not be based on the same criteria as such a query. That would duplicate the query.	**Example:** a collection of bolts (in stock) is a valid collection, because it is based on the location of the items. A collection of all bolts in a database is not a valid collection, because that ad hoc collection can be found using a query on the classification relation (select all items that are classified as a bolt). The membership of such a collection would duplicate the classification relation, because it would use the same criterion for membership.
Check: verify whether the criterion for membership of a collection duplicates an existing relation type.	

8.7 Guidelines for Cross Functional Units

8.7.1 Introduction

In this last subclause, relations and guidelines are discussed that do not specifically fall under a particular functional unit.

8.7.1.1 Applicable Concepts

Id	Name	Role
G1	Individual thing	A concept that can be used to indicate that something is an individual thing and not a kind of thing.

8.7.1.2 Applicable Relation Types

Id	Name	Role
G2.1	Collection relation	To specify that items are element of a collection.
G2.2	Synonym relation	To specify that an object that is referenced with a particular name in one context is the same object as an object that is referenced in another context with possibly a different name.
G2.3	Naming relation	To specify how something is called in a particular context.
G2.4	Classification relation	To relate individual things to kinds of things that classify the individual things.

8.7.2 Guideline for naming relations

<table>
<tr><td colspan="2"><b>Guideline for naming relations</b>
<i>Concepts in different functional units may share a naming relation with a term from the functional unit Vocabulary.</i></td></tr>
<tr><td>Id: G3</td><td></td></tr>
<tr><td><b>Explanation:</b> a term (name or abbreviation) in the vocabulary can be allocated to any concept in any functional unit.</td><td><b>Example:</b> the concept 130030 in an Aspect Model is related to a term ‘bearing’ in the vocabulary. If there is also a taxonomy about types of bearing separate from the Aspect Model, then the concept for bearing in the taxonomy shall use the same unique identifier 130030 as the concept in the Aspect Model and shall be related to the same term or its synonym in the vocabulary.</td></tr>
<tr><td colspan="2">Check: -</td></tr>
</table>

8.7.3 Guideline for synonyms

<table>
<tr><td colspan="2"><b>Guideline for synonyms</b>
<i>Synonyms may be specified by linking terms in the functional unit Vocabulary that share the same meaning through naming relations that relate the terms to a concept (UID).</i></td></tr>
<tr><td>Id: G4</td><td></td></tr>
<tr><td><b>Explanation:</b> if a concept in the dictionary has a naming relation with more than one term in the vocabulary, apparently these terms may all be used to refer to the concept (and therefore they are synonymous). Terms may be codes, names, abbreviations and symbols.</td><td><b>Example:</b> the terms ‘computing unit’ and ‘computer’ both refer to physical object 70051.</td></tr>
<tr><td colspan="2">Check: -</td></tr>
</table>

8.7.4 Guideline for homonyms

<table>
<tr><td colspan="2"><b>Guideline for homonyms</b>
<i>Homonyms may be specified using a single term from the functional unit Vocabulary by relating such a term to multiple concepts (UIDs).</i></td></tr>
<tr><td>Id: G5</td><td></td></tr>
<tr><td><b>Explanation:</b> if a term from a vocabulary has a naming relation with two concepts (UIDs) in a dictionary, this means that these different concepts apparently may be denoted using the same term (meaning they are homonyms). The context in which the term is used determines which concept is meant. It is recommended to denote the usage context by a ‘language community’.
NOTE If the concepts are related to the (different) unique identifiers in the Vocabulary, then the homonym is already defined in the Vocabulary.</td><td><b>Example:</b> the term ‘bank’ has both a relation with physical object #2, being a financial institution, and physical object #43, being a part of a river.</td></tr>
<tr><td colspan="2">Check: -</td></tr>
</table>

8.7.5 Guideline for classification of individual things

<table>
<tr><td colspan="4"><b>Guideline for classification of individual things</b>
<i>Each individual thing shall be classified by a kind of thing.</i></td></tr>
<tr><td colspan="2">Id: G6</td><td colspan="2"></td></tr>
<tr><td colspan="2"><b>Explanation:</b> this guideline provides guidance on the use of a knowledge library. It ensures the unambiguous definition of what kind individual things are. For computer interpretation it is essential that the indication of what things are is not dependent on the interpretation of the parts of the content of text strings, but on the unique identifiers of concepts or on one-to-one identity between a term and a standardized reference term in a particular context.</td><td colspan="2"><b>Example:</b> object B is classified as a bearing with UID 130030. This means unambiguously that B is a physical object and not an activity, so that a computer can act accordingly (e.g. by asking for aspects and not for timing and performer or subject).</td></tr>
<tr><td>(1) B</td><td colspan="2">is classified as a</td><td>(130030) bearing</td></tr>
<tr><td colspan="4"><b>Check:</b> whether every individual thing is classified by a concept and that every kind of concept is defined as a subtype of concept.</td></tr>
</table>

8.7.6 Guideline for multiple specifications for physical objects

<table>
<tr><td colspan="2"><b>Guideline for multiple specifications for physical objects</b>
<i>If a concept in the functional units Taxonomy, Aspect Model or Composition Model (with different aspects in those functional units) is related to the same concept in a Dictionary, then the definition of that concept shall allow for those differences in aspects.</i></td></tr>
<tr><td>Id: G7</td><td>CC: 0.2</td></tr>
<tr><td><b>Explanation:</b> The specification of aspects of concepts in a Taxonomy, Aspect Model and Composition Model can be different, because the aspects in those units do not define the concepts, but only support various options to create individual things. Multiple specifications may therefore be applicable to the same concept.</td><td><b>Example:</b> a valve may have a ‘stem diameter’ in an Aspect Model, whereas in Composition Model it may have a part that is a ‘closure member’, whereas the closure member has a ‘material of construction’. Both can be related to ‘valve’ in a dictionary, as both are valid for any valve.</td></tr>
<tr><td colspan="2"><b>Check:</b> whether the definitions in a dictionary include too many constraints (that are in fact only applicable to particular subtypes of the concept or are optional)</td></tr>
</table>

Annex A
(informative)

Quality labels

A.1 General

A.1.1 Introduction

In Clause 5 it has been indicated that an inspection organization may play a role in demonstrating that a knowledge library complies with these standard guidelines. By means of a quality label a particular guarantee can be given, so that libraries may be integrated or parts of libraries may be exchanged among those libraries. In this clause a first step is made in investigating how a *quality label* may be obtained with which this quality guarantee is upheld. It needs to be stated emphatically that the content of this clause must be regarded as *a first exploration*. It has not been planned that on the basis of this version of the standard quality labels can be awarded. At a later stage this must be further scrutinized.

In A.1.2 the guidelines that have been drawn up are related to a conformance level. In A.2 it is discussed which parts of a library may be assigned a quality label. A.2.2 forms a description of the various institutions or bodies that are involved in this process. In A.2.5 and A.2.6 the various evaluation processes are discussed.

A.1.2 Conformance levels

The objective of conformance levels is to add a gradation to the quality level of knowledge libraries. The higher the class to which the library complies, the higher the quality level will be. In 5.2 it has been indicated that the guidelines and descriptions in this standard have the specification level of conformance level 0 and 1. This effectively means that quality level 1 has been realized if all guidelines are complied with.

In addition, it has been decided to add a further differentiation within class 0. In other words: within this class a further subdivision has been made on the basis of quality sublevels. In this way it becomes possible to stipulate growth scenarios for knowledge libraries, so that their applicability and quality are improved. The differentiation between the various conformance levels is therefore partly meant to lower the threshold for library owners to comply with the current standard. If the effort and the lead-time required to reach the first 'standard compliance level' can still be kept track of, the willingness to take that step will increase.

Within conformance level 0 there are 3 sub-levels defined. Conformance level 1 has no subdivision and the guidelines on classification of individual things do not belong to a conformance level.

- *Conformance level 0.1*: This sub-level specifies basic requirements to provide a foundation for harmonization of different libraries.
- *Conformance level 0.2*: This sub-level specifies additional requirements to enable sharing of subsets of terminology, definitions.
- *Conformance level 0.3*: This sub-level specifies requirements to enable shared use of concept definitions and some basic knowledge.
- *Conformance level 1.0*: This conformance level specifies requirements to enable shared use of models that distinguish roles from role players and to enable automated semantic verification.

One should take into account that the conformance levels are not related to the library in its entirety, but are differentiated in turn on the basis of the various functional units. This means that a knowledge library that is exclusively meant for information in only one functional unit can still be considered a library of a particular conformance level. For example, an object library for requirements specifications that contains standard specification sheets (i.e. Aspect Model functionality) can still be considered a library of conformance level 0.3. This then only holds for the functional unit Aspect Models. The other functional units might be lacking and therefore need not even qualify for class 0.1. Moreover, there is also a conformance level for the various concepts and relation types. The presence of certain concepts and relation types also forms an indication of the richness and quality of knowledge libraries.

In Table A.1 below a schematic outline is given of which conformance level has been assigned to the various guidelines, concepts and relation types.

Table A.1 — Guidelines, concepts and relation types and their conformance levels

Functional Unit	Id	Guideline/Object/Relation	Conformance level
Vocabulary	V1.1	term	0.1
	V1.2	unique identifier	0.2
	V1.3	language	0.3
	V1.4	language community	0.3
	V2.1	naming relation	0.2
	V3	guideline for terms	0.1
	V4	guideline for term singular	0.2
	V5	guideline for term characters	0.2
	V6	guideline for term formatting	0.2
	V7	guideline for unique identifiers	0.3
Dictionary	D1.1	concept	0.1
	D1.2	aspect	0.2
	D2.1	specialization relation	1.0
	D2.2	definition relation	0.2
	D3	guideline for dictionary presentation	0.1
	D4	guideline for usage of descriptive terms	0.3
	D5	guideline for lexical definition	0.3
Taxonomy	T1.1	concept	0.1
	T1.2	aspect	0.2
	T1.3	role	0.3
	T2.1	specialization relation	0.1
	T2.2	concept-aspect relation	0.2
	T2.3	composition relation	0.2
	T2.4	aspect-scale relation	0.2
	T3	guideline flexibility of composition	0.3
	T4	guideline inheritance through specialization	0.2
	T5	guideline single discriminator for physical object specialization	0.2

Table A.1 *(continued)*

Functional Unit	Id	Guideline/Object/Relation	Conformance level
Taxonomy	T6	guideline for discriminating description of specialization	0.3
	T7	guideline for non-overlap of subtypes of a concept	0.2
	T8	guideline consistency of discrimination and aspects	0.3
	T9	guideline for definition of roles of physical objects	1.0
	T10	guideline for definition of intrinsic aspects	1.0
	T11	guidelines for names of aspects	1.0
	T12	guideline for top of specialization hierarchy	1.0
Knowledge models excluding composition	E1.1	physical object	0.1
	E1.2	aspect	0.1
	E1.3	scale	0.1
	E2.1	physical object-aspect relation	0.1
	E2.2	aspect-scale relation	0.2
	E3	guideline for scales for aspects	0.1
Knowledge models including composition	I1.1	physical object	0.1
	I1.2	aspect	0.1
	I1.3	scale	0.1
	I2.1	physical object-aspect relation	0.1
	I2.2	composition relation	0.1
	I2.3	aspect-scale relation	0.2
	I3	guideline for scales for aspects	0.1
	I4	guideline for flexibility of physical object composition	0.2
	I5	guideline for multiple wholes for a part physical object	0.2
	I6	guideline for composition versus specialization	0.2
Collection	C1	collection	0.1
	C2	collection relations	0.1
	C3	guideline for collections	0.2
Cross-functional units	G1	individual thing	1.0
	G2.1	collection relation	1.0
	G2.2	synonym relation	0.3
	G2.3	naming relation	0.1
	G2.4	classification relation	
	G3	guideline for naming relations	0.1
	G4	guideline for synonyms	0.2
	G5	guideline for homonyms	0.3
	G6	guideline for classification of individuals	
	G7	guideline for multiple specifications for physical objects	1.0

A.2 Components qualifying for a quality label

A.2.1 Introduction

In Clause 5 it has been indicated that these standard guidelines are not directly aimed at particular databases, but rather at the information according to the functional units that may be generated from such a database. This shift of guidelines does not imply that no quality label can be assigned to one of the components of a knowledge library database system. However, when assigning a quality label, we must take into consideration the *combination* of the system as well as the conversion instructions for generating information according to the functional units.

For a proper understanding of the coverage of the guidelines we briefly describe the components of a knowledge library system.

A knowledge library *system* is illustrated in Figure A.1.

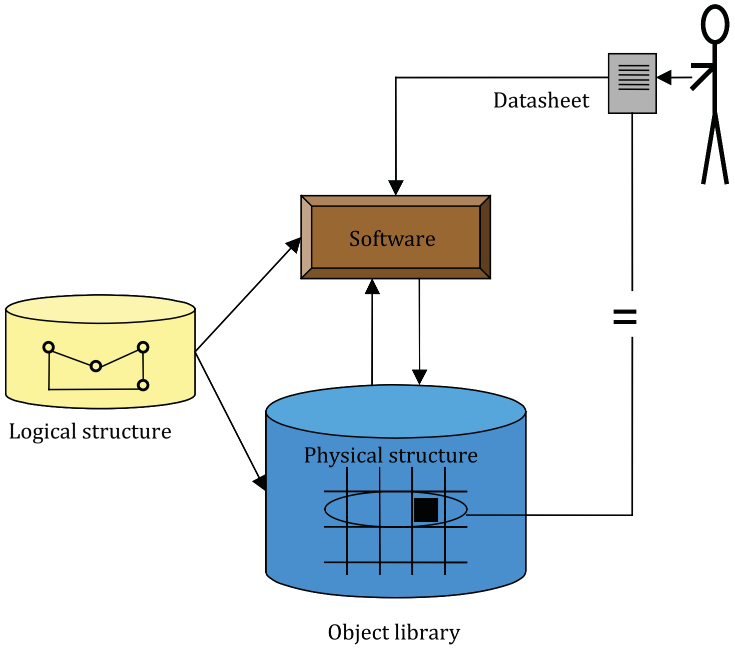


Figure A.1 — Knowledge library system

The above figure illustrates that a knowledge library *system* may roughly be divided into the following five components.

- The logical structure: this is the information model of the knowledge library.
- The physical structure: this is the implementation of the information model used in order to make it possible for data (i.e. object information) to actually be stored. This may be a database structure, but can also be an XML file structure.
- The software: this is the collection of computer interpretable procedures used for managing the content of the knowledge library and communicating with users and with other systems. The entry of concepts and relations, the generation of information according to the functional units as well as various import and export features belong to this component.
- The knowledge library (content): the actual data that represent the information about the objects in the library. These data are stored in a physical structure according to a logical structure with the help of the corresponding software.
- A user interface by which a user can communicate with the system.

The following example illustrates in short for which components the quality labels are applicable.

Assume that a particular knowledge library always specifies its physical object classes in the plural form. The owner of the knowledge library would like to generate a standard-tested Vocabulary that can obtain a quality label for the Vocabulary functional unit.

While reading the guidelines for Vocabularies he notices that guideline V4 requires that the naming of terms must take place in the singular form as a criterion to comply with conformance level 0.2 (see Table A.1 — Guidelines, concepts and relation types and their conformance levels). Thus it seems as if his library does not comply with the guideline in question. However, upon further investigation of his knowledge library, he concludes that eliminating plural suffixes such as ‘s’ at the end of the names is sufficient for compliance with this guideline. He therefore decides to adopt this as a required conversion instruction. Taking this instruction into account, the certifying body *is* able to assign a quality label for the functional unit in question.

Bearing this in mind, within the current standard the following components have been selected to which a quality label may be assigned.

— The knowledge library. The most important quality label by far concerns the knowledge library. In this connection, the evaluation consideration is: *Can the knowledge library be converted (by means of the conversion instructions) to the selected functional units and there comply with the standard in force?* If this evaluation step has been finished positively, the library then may be assigned a quality label.

— The software. The software used for filling and managing the knowledge library, too, may be assigned a quality label. In this connection, the evaluation consideration is: *Does the software monitor the integrity of the library during entry and changes so that the knowledge library remains compliant with the first quality label?* Software that passes this test will disable users from corrupting the standard status of the knowledge library. This means it is impossible to ‘mess up’ data so that it no longer is compliant with the standard. The specific (technical) guidelines to which the software must comply will be scrutinized at a later stage.

— The information model. The third quality label concerns the written information model that forms the basis for the knowledge library. In this connection, the evaluation consideration is: *Does the information model allow for the recording of object data by means of which a knowledge library can be set up that is incompliant with the first quality label?* This evaluation is a much more drastic step than the ‘software evaluation’ described above. A quality label for the information model implies that it is impossible to lay down data that cannot be converted to a correct implementation of the functional units. Here as well, separate evaluation guidelines must be drawn up.

A.2.2 Roles of parties concerned

The entire certifying process requires that at least a number of parties are involved. In this subclause these parties or organizations will be briefly described.

A.2.2.1 Knowledge library owner/manager

The owner of the knowledge library takes the initiative of offering his library to the inspecting body. Such an initiative may result from the need to provide a certain quality guarantee to the users or to enable harmonization/integration with other libraries.

A.2.2.2 Knowledge library system software owner

Since the content and the product in which the content is managed are not identical, the maker/owner of the software has been included as an actor. This organization will also be able to offer its product to the inspecting body. The quality label borne by his software guarantees its users that any objects entered or managed are always compliant with this standard.

A.2.3 Inspecting body

This body has been authorized by a standardization body or similar organizations to perform the inspection and to award the quality label. For this purpose the inspecting body makes use of a 'test laboratory' in order to carry out the required conformity tests (including the required instructions). If necessary, this body is also authorized to ask other 'test laboratories' for a second opinion.

A.2.4 Test laboratory

The test laboratory carries out the actual tests and therefore evaluates whether the knowledge library, the software or the information model is compliant with the guidelines. The party offering the material for testing must indicate in advance to which component he would like to have the quality label assigned. For instance, it must be known which functional units are concerned, which conformance level is desired as well as the nature of the context and application of the knowledge libraries. The point of departure is of course formed by the guidelines that have been specified in Clause 6.

A.2.5 Evaluation of knowledge libraries and conversion instructions

In order to verify whether a knowledge library is compliant with this standard, the owner must provide the inspecting body with the library as well as the conversion instructions. The inspecting body will forward the material to a test laboratory where the required tests are performed. Broadly speaking, the test laboratory carries out the following activities.

— Analyses the knowledge library and the corresponding information model, conversion instructions and the specified test domain.

— Performs the conversion process on the basis of instructions in order to generate the specified functional units. The owner may already have offered supporting software for this purpose.

— Verifies the generated units on the basis of the guidelines. Specific test protocols have yet to be developed for this purpose.

— Reports the results to the certifying body.

Depending on the result, the inspecting body will decide whether or not to assign a quality label.

The owner shall provide the material as follows:

— The knowledge library with a physical structure in compliance with either (1) ISO 10303-21, (2) 'standard' XML, (3) ASCII text file or (4) paper copy.

— An associated information model that has been described on the basis of either (1) Express: ISO 10303-11, (2) UML-2, (3) XML XSD, (4) XML DTD or (5) RDF(S)/OWL. If the knowledge library has been laid down in a model approved by this standard (currently ISO 15926-2, ISO 10303-221, ISO 13584), a reference will suffice.

— Conversion instructions. These will in many cases consist of a mapping table in which the entries are listed that must be mapped from their proper information model to the functional units. In some cases changes may need to be made.

It will be clear that the way in which libraries are provided for testing has a bearing on the envisaged effort (and consequently the costs involved in awarding the quality label). Libraries that are based on existing information models will therefore be easier to test.

A.2.6 Verification of information model and software

As indicated in A.2, the information model and the software may also be certified. In such a case, the test will rather be aimed at guaranteeing that the software and the information model force users to comply with these standard guidelines.

For this purpose, the test laboratory will make use of a so-called *verification object*. In this model, objects are specified that do not comply with the criteria on purpose. In this way, it may be evaluated to what extent the information model or the software allows for the use of test data.

It must be stated that this verification object model is not immediately applicable. First it must be converted to knowledge library storage characteristics. For this purpose the provided instructions must be used. The instructions will then exclusively be applied in an inverted fashion (from a functional-unit environment to the native environment). In this way, the result will be suitable for entry into the provided library. The extent to which this is (im)possible will decide the level of sustainability for (continued) compliance with the guidelines.

A.3 Software testing

For the testing of the software the following steps must be taken.

— Analysis of, for example, the knowledge library and the corresponding information model, software, conversion instructions and the specified test domain.

— Generation of the usable test data by means of the standard test objects model.

— Testing of the various software functionalities by trying to enter/adapt the test data. Error messages should be produced.

— Reporting of the results to the certifying body.

The material that is to be provided is identical to the material for testing the knowledge library. The software itself is also included.

A.4 Information model testing

For the testing of the information model the following steps must be taken:

— Analysis of, for example, the knowledge library and the corresponding information model, software, conversion instructions and the specified test domain.

— Generation of the test data by means of the standard test objects model.

— Testing of the various software functions by recording test data by means of import facilities. This must result in the display of the correct error messages. If no software has been provided, the information model will be analysed.

— Reporting the result to the certifying body.

The material that is to be provided is identical to the material for testing the knowledge library; the software itself is also included.

Annex B
(normative)

Reference model

B.1 Reference model

B.1.1 Introduction

The concepts (or ‘classes’) that have been defined in this standard are interrelated according to the structure schemes below. The relations shown in these schemes can be deduced from the text of the standard. The structure schemes should not be interpreted as a ‘data model’ with ‘entity types’ and ‘attribute types’, but as a ‘reference model’ (or an ‘instance model’). Concepts that appear in knowledge libraries are therefore not ‘instances’ of these concepts, but they shall be defined as their specializations (= subtypes). The arrows indicate the reading direction for the phrases that represent the relation types.

B.1.2 Explanatory notes to the figures

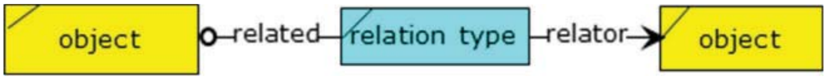
— Each square represents a concept (also called a ‘kind’, ‘class’, ‘sort’ or ‘type’). For instance, ‘physical object’ does not represent ‘a physical object’, but the concept ‘physical object’ that can be used to classify something as being of that kind. This means that an instance in knowledge libraries is not an instance of ‘physical object’, but a specialization (= subtype) of it.

— Each blue square represents a concept that is a subtype of ‘relation’.

— Each purple square represents a particular instance of a specialization relation. This means that the various (purple) instances of a specialization relation represent different relations between kinds of things, each of which is qualified as a specialization relation.

— A slanted line in an upper left corner of a square box indicates that the box represents an instance and not an entity type.

Relations (expressions of facts) are represented in the figures as follows:



A figure of this kind should be read as one (binary) relation (an expression of one fact). The relation starts with the object in the box to which a small circle is connected, it is read in the direction of the arrow, using the phrase in the middle box (which represents the relation type), and terminates with the object in the box to which the arrow is connected. The first object plays a role of the first kind in that relation (‘related’), the second object plays a role of the second kind in the relation (‘relator’). The definitions of the kinds of relations demonstrate that there are also inverse phrases defined that can be used to create expressions that are read in the opposite direction.

— The names in the boxes represent concepts, although a concept is in fact represented by a unique identifier (UID), which differs from its name(s). This is clarified by the naming relation.

— The numbers attached to the boxes refer to the fact ids in Table B.2.

— The collection relation and the naming relation are in fact defined as relations with anything, which may be a concept or an individual thing. Consequently subtypes that are specializations of ‘concept’ as well as individual things may be grouped and named. As this standard is intended for knowledge libraries, which deal with concepts only, the reference model shows these relations as being related to ‘concept’ instead of ‘anything’.

— The translation relation and the binary relation are not shown in the figures. The concept binary relation can be treated as equivalent with the concept relation, as this mainly deals with binary relations. The translation relation can be treated as equivalent to the synonym relation, because a translation of a term is in fact a kind of synonym in different languages.

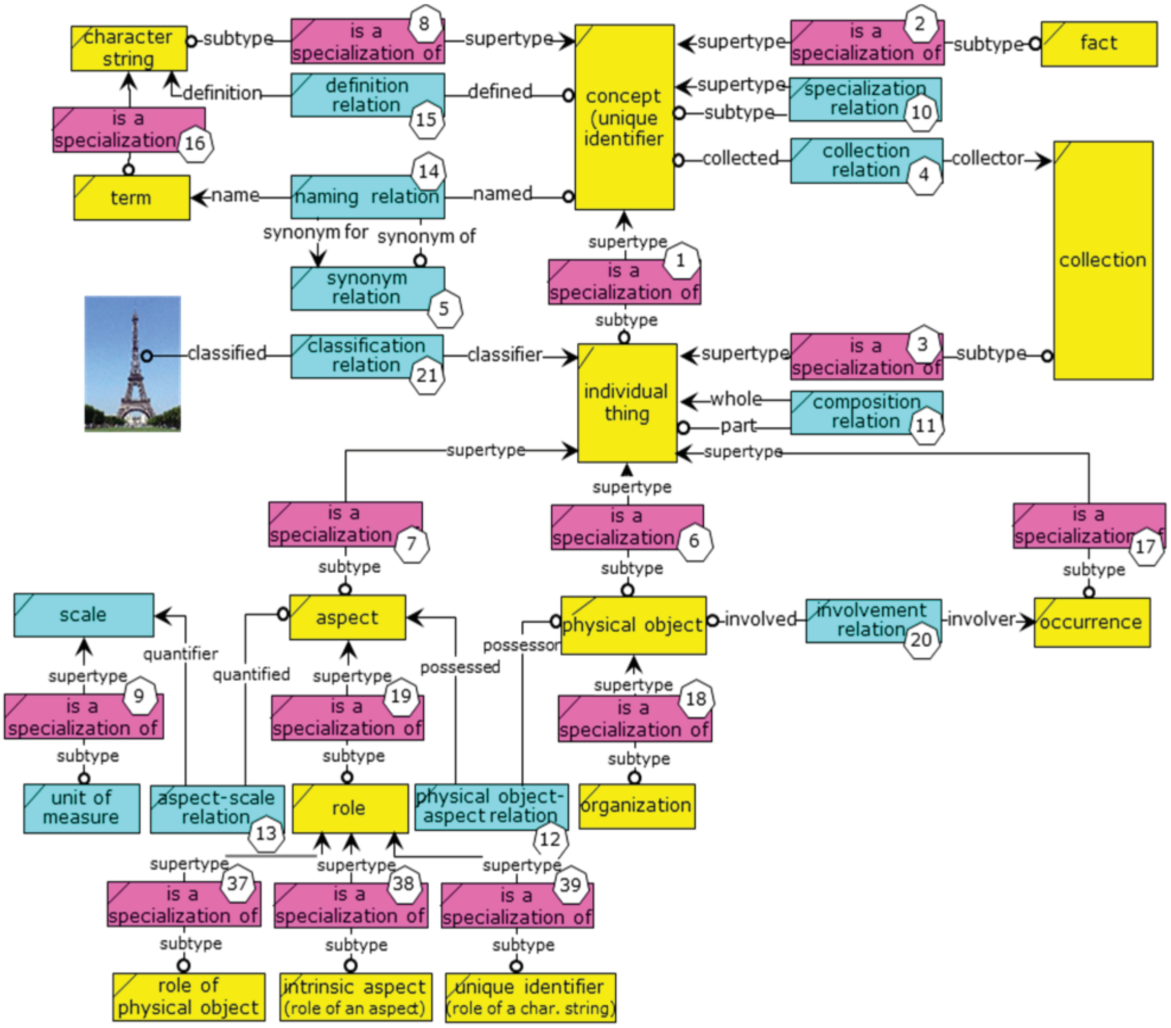


Figure B.1 — Reference model of the object hierarchy

— Figure B.1 focuses on the taxonomy of concepts not being relation types. Figures B.2 and B.3 focus on the taxonomy of kinds of relations.

— The concept ‘individual thing’ in Figure B.1 does not represent *an* individual thing, but a kind of thing that can be used to classify all individual things. The picture of the Eiffel Tower refers to a UID that represents any individual thing, for example the Eiffel Tower.

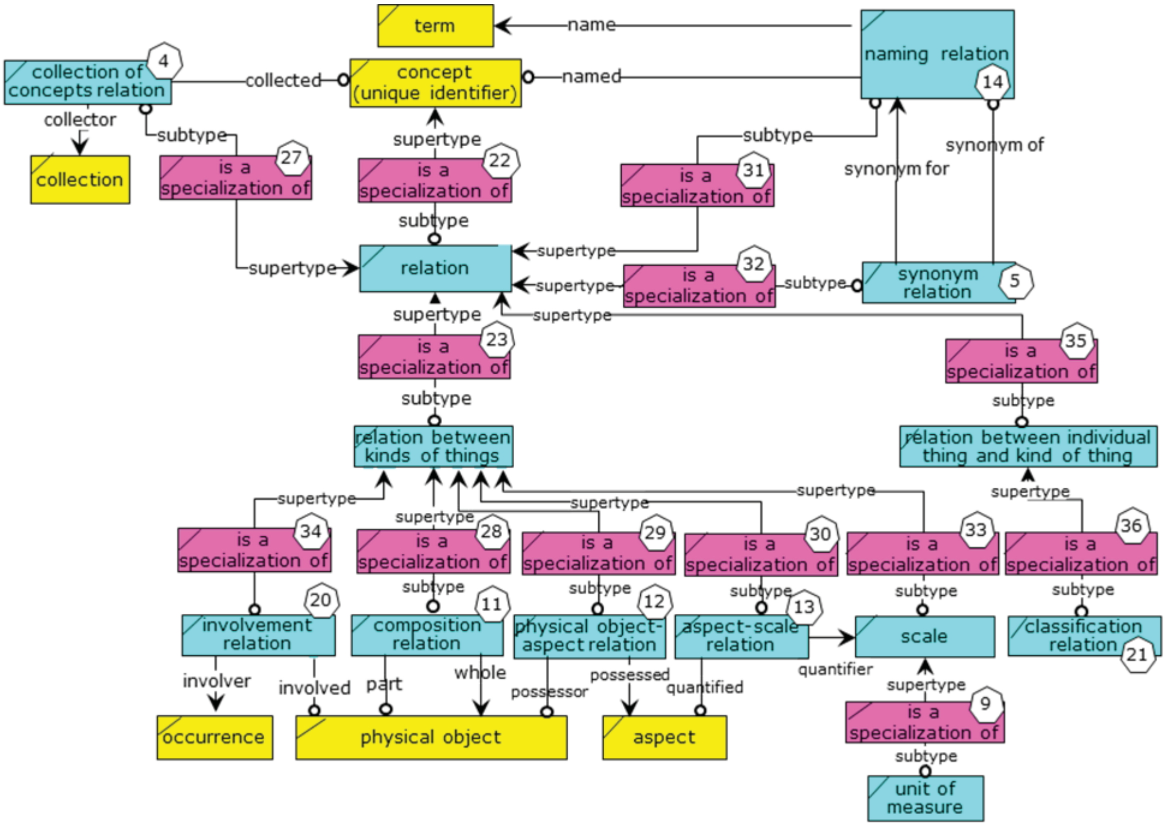


Figure B.2 — Reference model of the relation hierarchy

Figure B.3 presents a taxonomy of specialization relations.

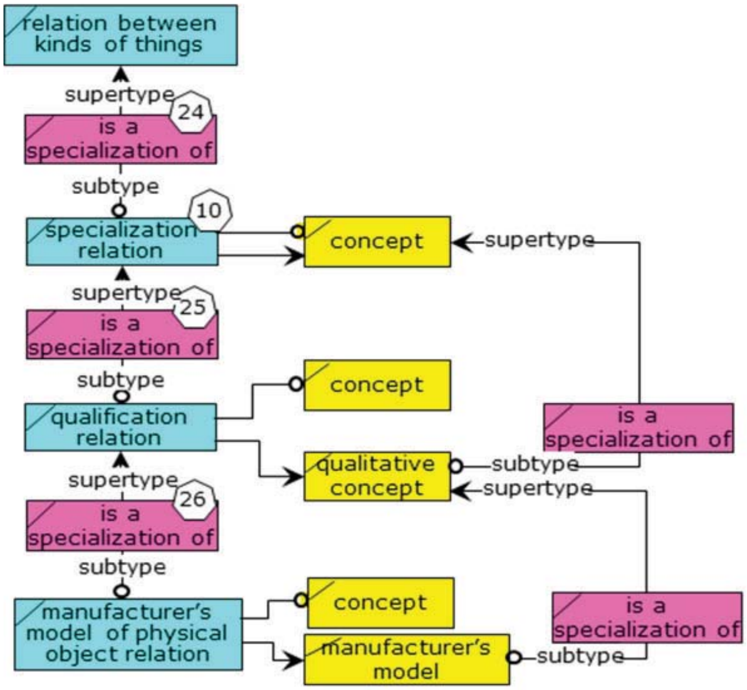


Figure B.3 — Taxonomy of specialization relations

Figure B.3 illustrates that the specialization relation as well as its subtypes relate two concepts, one with a role as subtype and the other with a role as supertype. For the qualification relation and the ‘manufacturer’s model of physical object relation’, the subtype concepts are further constrained by being a qualitative concept and a manufacturer’s model, respectively. It should be noted that this allows a qualitative concept to be defined as a subtype of conceptual concept, or a subtype of another qualitative concept (enabling a taxonomy of qualitative concepts). The ‘manufacturer’s model of physical object relation’ allows a manufacturer’s model to be defined as a subtype of another manufacturer’s model, or a subtype of a qualitative concept, or a subtype of a conceptual concept.

Table B.1 provides an overview of which concepts and relation types are relevant for each functional unit.

Table B.1 — Applicable object types per functional unit

Object Name	Functional Unit						
	Vocabulary	Dictionary	Taxonomy	Aspect Models	Composition Models	Collection	Cross Functional Units
anything (unique identifier)	x						
object			x				
collection						x	
physical object		x	x	x	x		
aspect		x	x	x	x		
term	x						
language	x						
language community	x						
unit of measure (scale)			x	x	x		
specialization relation			x				
collection of concepts relation			(x)			x	x
composition relation			(x)		x		
aspect-unit relation			(x)	x	x		
physical object-aspect relation			(x)	x	x		
naming relation							x
synonym relation							x

Table B.2 provides an overview of the relations, especially the taxonomy of concepts and relation types as presented in Figure B.1 and Figure B.2.

Table B.2 — Summary table, including the reference fact IDs from the figures

Unique id	Left Object Name	Fact ID	Relation Type Name	Right Object Name
730067	individual thing	1	is a specialization of	concept
970015	fact	2	is a specialization of	concept
2849	collection	3	is a specialization of	individual thing
4990	concept	4	can be an element of a	collection
5117	name relation	5	is a synonym of	naming relation
730044	physical object	6	is a specialization of	individual thing
790229	aspect	7	is a specialization of	individual thing
910827	character string	8	is a specialization of	concept

Table B.2 *(continued)*

Unique id	Left Object Name	Fact ID	Relation Type Name	Right Object Name
5467	unit of measure	9	is a specialization of	scale
4990	concept	10	is a specialization of	concept
730044	physical object	11	can be a part of a	physical object
730044	physical object	12	can have as aspect a	aspect
790229	aspect	13	can be mapped on scale to a	scale (unit of measure)
4990	concept	14	has as name	term
4990	concept	15	has as definition	character string
970050	term	16	is a specialization of	character string
193671	occurrence (activity)	17	is a specialization of	individual thing
990008	organization	18	is a specialization of	physical object
160170	role	19	is a specialization of	aspect
730044	physical object	20	can be involved in a	occurrence (activity)
	an individual thing (e.g. the Eiffel Tower)	21	is classified as a	individual thing
2850	relation	22	is a specialization of	concept
4718	relation between kinds of things	23	is a specialization of	relation
1146	specialization relation (is a specialization of)	24	is a specialization of	relation between kinds of things
1726	qualification relation (is a qualification of)	25	is a specialization of	specialization relation
5396	manufacturer's model of physical object relation (is a model of)	26	is a specialization of	qualification relation
4730	collection of concepts relation (is an element in collection of classes)	27	is a specialization of	relation
1191	composition of relation (can be a part of a)	28	is a specialization of	relation between kinds of things
2069	physical object-aspect relation (can have as aspect a)	29	is a specialization of	relation between kinds of things
4757	aspect-scale relation (can be quantified on scale)	30	is a specialization of	relation between kinds of things
5117	naming relation (has as name)	31	is a specialization of	relation
1981	synonym relation (is a synonym of)	32	is a specialization of	relation between kinds of things
1733	scale	33	is a specialization of	relation between kinds of things
4648	involvement relation	34	is a specialization of	relation between kinds of things
4719	relation between an individual thing and a kind of thing	35	is a specialization of	relation
1225	classification relation	36	is a specialization of	relation between an individual thing and a kind of thing
5354	role of physical object	37	is a specialization of	role

Table B.2 *(continued)*

Unique id	Left Object Name	Fact ID	Relation Type Name	Right Object Name
4289	intrinsic aspect	38	is a specialization of	role
970047	unique identifier	39	is a specialization of	role
1146	is a specialization of	40	is a synonym of	specialization relation
1191	can be a part of a	41	is a synonym of	composition relation
4730	is an element in collection of classes	42	is a synonym of	collection relation
4757	can be quantified on scale	43	is a synonym of	aspect-unit relation
2069	can have as aspect a	44	is a synonym of	physical object-aspect relation
5117	has as name	45	is a synonym of	naming relation
1981	is a synonym of	46	is a synonym of	synonym relation
1726	is a qualification of	44	is a synonym of	qualification relation
5396	is a model of	45	is a synonym of	manufacturer's model of physical object relation
4648	can be involved in a	46	is a synonym of	involvement relation

Annex C
(informative)

Guidelines under scrutiny

C.1 Introduction

In this annex an overview is given of the considered guidelines, objects and relation types that as such have not yet been approved by the committee. In a next version of this standard these guidelines will be reconsidered.

C.2 Definitions

C.2.1 Definitions of concepts

class

a concept that indicates the nature of ideas that represent commonalities between individual things. All classes are intended to be used for the classification of individual things.

NOTE Each subtype of the concept that is called 'individual thing' is by definition a kind of class that can be used to classify individual things.

EXAMPLES Each of the following concepts – switch, armature, electric, mechanical engineering, portal, processing industry – is an example of a thing that can be classified as a class.

NOTE The concept 'class' is used in this standard, especially its subtypes 'conceptual class' and 'qualitative class'. This may justify the definition of their common supertype. The above definition of 'class' may be irrelevant for the scope of this standard. The term 'object type' as synonym of 'class' is confusing as this standard uses the term 'object' to indicate a type or kind of individual thing.

C.2.2 Definitions of relation types

none

C.3 Functional Unit: Dictionary

C.3.1 Guideline natural language

<table>
<tr><td colspan="2"><b>Guideline natural language</b>
<i>Additional data elements linked to terms that have been specified in the functional unit Vocabulary must be laid down in compliance with the guidelines used in official language guidelines.</i>
<i>For Dutch this is the 'Woordenlijst Nederlandse taal' [official list of Dutch words and their spelling, also known as 'het groene boekje'].</i></td></tr>
<tr><td><b>Id:</b> xxx</td><td></td></tr>
<tr><td><b>Explanation:</b> word gender, notation, plural, etc.</td><td><b>Example:</b> ball m., balls, small balls</td></tr>
<tr><td colspan="2">Check: -</td></tr>
</table>

NOTE Because of the possibility of synonyms, this guideline seems superfluous. It might be introduced as a recommendation for those languages where there is an officially published list of words.

C.4 Functional Unit: Taxonomy

C.4.1 Guideline names of intrinsic aspects

Guideline names of intrinsic aspects *The fact that an intrinsic aspect is defined as a role of an aspect and is by definition possessed by a particular kind of physical object shall be reflected in the name of the intrinsic aspect, so that the name of the kind of aspect and the name of a kind of physical object is included.*	
Id: T14a	
Explanation: an aspect that by definition is only possessed by a particular kind of physical object is an aspect in the role of being by nature possessed (i.e. is intrinsic). It is therefore a subtype of 'intrinsic aspect' that in turn is a subtype of role in a relation (role of an aspect in a possession relation).	**Example:** 'pipe diameter' is a role of a diameter in a possession relation that relates a pipe to a diameter. Therefore it is a correct intrinsic aspect.
Check: verify whether references to physical objects are found in the names of intrinsic aspects.	

NOTE This is closely related to guideline T10.

C.4.2 Guideline naming of specialized physical objects

Guideline naming of specialized physical objects *The specified discriminating aspect must not be inconsistent with the naming of the physical object.*	
Id: T17	CC: 0.3
Explanation: Preferably the value of the discriminating aspect is included in the name of the physical object that is defined by that value.	**Example**: 'Reciprocating pump' and 'centrifugal pump' are valid names for physical objects if the discriminating aspect 'operating principle' has the respective values of 'reciprocating' and 'centrifugal'.
Check: verify whether the value of the discriminating aspect matches the name of the physical object.	

NOTE This seems rather trivial. Example of mismatches are required to justify this guideline.

C.4.3 Guideline categorization of discriminating aspects

Guideline categorization of discriminating aspects *The specified discriminating aspect (from T11) shall be qualified by a basic category.*	
Id: T18	CC: 0.3
Explanation: a discriminating aspect on the basis of which a subtype is defined, must be categorized by predefined types to support design.	**Example:** predefined categories may be 'application', 'objective', 'technology', 'form', 'function/solution', 'construction method', 'material'.
Check: verify whether the aspects refer to a category of an (agreed) list.	

NOTE This might be a recommendation rather than a guideline, especially as long as the categories are not an agreed standard set of categories.

C.5 Functional Unit: Aspect Models

none

C.6 Functional Unit: Composition Models

C.6.1 Applicable Relation Types

Id	Name	Role
	Spare relation	This relation indicates that a physical object normally is or can be a spare for another physical object. Note that an installed spare can also have a composition relation with the assembly in which it is installed.

NOTE This seems to be a specification of a requirement rather than a definition of knowledge. In the context of this standard, where we only describe kinds of things this relation become rather strange. For example, bearing can be a spare for a bearing, which is trivial. This may better be indicated by an aspect (of a part) that is called 'number of spares'.

C.6.2 Guideline physical objects as aspects

Guideline physical objects as aspects *The value of an aspect must not be a reference to a kind of physical object.*	
Id: I10	CC: 0.3
Explanation: an aspect is not meant to describe compositions of physical objects. If parts have a particular role for a specific kind of physical object, this must be handled by way of relations.	**Example:** The fact that a particular kind of connector is used as connecting method for a particular kind of lighting system should not be described by means of an aspect but by a standard composition. So, a lighting system should not have an aspect called connector type. It should have a composition relation with a connector and that connector can be classified.
Check: verify whether references to physical objects are found in the values of aspects.	

NOTE 1 This seems to be a specification of a requirement rather than a definition of knowledge.

NOTE 2 This seems relevant only if not all subtypes are allowed in a particular context. Then the part 'shall be classified by one of the' elements in a collection of allowed classes. To refer to the part in that context this requires a role to be defined for the part. For example: a 'pump bearing' in the context of manufacturer X 'shall be classified by one of the' collection 'allowed pump bearing types'.

C.6.3 Guideline uniformity of physical objects

Guideline uniformity of physical objects *Multiple different variants of kinds of physical objects may be included in the definition of a single kind of physical object.*	
Id: I11	CC: 0.2
Explanation: parts, especially features, accessories and (installed) spare parts, are the cause that kinds of physical objects will be used in the description of other kinds of physical objects. Those parts, etc. are separate kinds that shall have their own definition.	**Example:** For instance, a spare wheel may come optionally with a car (number of spares is 0 or 1). However, a spare wheel may require a particular specification that includes that only particular subtypes of wheel are allowed in the context of this kind of car. This type of car shall be defined to have 4 parts of type ‘normal wheel’ and 0 or 1 parts of type ‘exceptional wheel’.
Check: this need not be verified.	

NOTE This is a requirement in a particular context and may be out of scope for this standard.

C.7 Functional Unit: Cross Functional Units

C.7.1 Guideline additional object types

Guideline additional object types *The inclusion of object types in functional units that have not been defined in this standard (objective, function, etc.) must be explicitly defined by a specialization relation to such a supertype.*	
Id: xx	CC: 0.3
Explanation: other kinds of objects might be necessary than those defined in this standard.	**Example:** the objective ‘fire fighting’.
Check: -	

NOTE The guidelines on specializations of objects also apply to objects that are not defined in this standard. It is unclear what this proposal would add to the existing guidelines, especially if no additional relation types are proposed.

C.7.2 Guideline for references to standards

Guideline for references to standards *Objects should have a reference relation with a nationally or internationally approved classification standard if the definition or aspects are (partly) derived from such a standard.*	
Id: xx	CC: 0.3
Explanation: references are valuable as a source for standards that might be applicable in an application context.	**Example:** the International Standard for Classification 2001 (ICS).
Check: verify whether all physical objects have a classification relation with the standard in question.	

NOTE Differences in textual definitions may occur in different standards, whereas still the same concept is meant. This is addressed in this standard by the introduction of a language independent unique identifier for each concept that allows multiple text strings to describe (define) what is meant.

Annex D
(informative)

Index

Bibliography

[1] ISO/IEC 646:1991, *Information technology — ISO 7-bit coded character set for information interchange*

[2] ISO 1087-1:2000, *Terminology work — Vocabulary — Part 1: Theory and application*

[3] ISO/IEC 8859-1:1998, *Information technology — 8-bit single-byte coded graphic character sets — Part 1: Latin alphabet No. 1*

[4] ISO 10303 (all parts), *Industrial automation systems and integration — Product data representation and exchange*

[5] ISO 12006-3:2007, *Building construction — Organization of information about construction works — Part 3: Framework for object-oriented information*

[6] ISO 13584 (all parts), *Industrial automation systems and integration — Parts library*

[7] ISO 15926 (all parts), *Industrial automation systems and integration — Integration of life-cycle data for process plants including oil and gas production facilities*

[8] ISO/PAS 16739:2005, *Industry Foundation Classes, Release 2x, Platform Specification (IFC2x Platform)*

[9] DIN 6779-2, *Kennzeichnungssystematik für technische Produkte und technische Produktdokumentation — Teil 2: Kennbuchstaben*

[10] IEC 61360 (all parts), *Standard data element types with associated classification scheme for electric components*

[11] Elem91, Elementen methode '91 (Bond van Nederlandse Architecten [Association of Netherlands Architects])

[12] GWW Objectenbibliotheek, CROW. Ede, The Netherlands

[13] ICS, International Standard for Classification

[14] Installatiedelen-bibliotheek VNI, Uneto-VNI, Zoetermeer, The Nethelands

[15] LexiCon, STABU, Ede The Netherlands

[16] MESC (Material and Equipment Standards Catalogue), Shell buyers specifications object library

[17] Merriam-Webster online, http://www.merriam-webster.com/

[18] Rosetta, RosettaNet Technical Directory

[19] UBIM91, Uniforme basisindeling installatiemateriaal '91

[20] UNETO-VNI classificatiesysteem

[21] Van Dale, Groot Woordenboek der Nederlandse Taal; Van Dale On-line Woordenboek, http://www.vandale.nl/opzoeken/woordenboek/

[22] Gellish, A Generic Extensible Ontological Language - Design and Application of a Universal Data Structure, by dr. ir. Andries van Renssen, 2005, Delft University Press. Also available via http://www.gellish.net/downloads

[23] IFD Library Group, BuildingSmart

ICS 91.010.01

Price based on 72 pages