Gellish Modelling Method

Part 7

General Principles and Guidelines

(Guidelines for the use of Gellish English)

by

Dr. Ir. Andries van Renssen

24 June 2005

Table of Content

1. Introduction

Gellish English is defined in the Gellish English 'Smart Dictionary'. The Gellish Modeling Method parts provides an extensive guidance on the use of Gellish[1]. However, for practical application of Gellish there appeared to be a need for short guidelines on a number of specific topics. This document has grown from that need. It provides a limited but growing number of guidelines on the use of Gellish English.

The guidelines in this document assume that a Gellish Database is used that is compliant with the specifications in the document 'The Gellish Database Definition'. The guidelines are directly translatable to other versions of Gellish, such as Gellish Dutch, Gellish German, etc.

The guidelines distinguish between Gellish facts in a vocabulary, a dictionary, a taxonomy (with textual definition or with modeled definitions), knowledge models and in product models of various complexities. The table below illustrates the object types and relation types that are involved in the various kinds of models. Crosses in brackets mean that the presence of such objects or relation types are optional.

		1	2	3	4	5	6
Object name	Gellish synonym	Vocabulary	Dictionary	Taxonomy	Knowledge models without product structure	Knowledge models with product structure	Individual product Models
language / language community		(x)	(x)	(x)	(x)	(x)	(x)
identifier of concept (or of individual object)		(x)	(x)	x	x	x	x
term (name)		x	x	x	x	x	x
definition			x	x	x	x	
naming relation	is referenced as		(x)	x	x	x	x
alias relation	is a synonym of		x	x	x	x	x
specialization of concept relation	is a specialization of			x			
collection relation	is an element of					x	x
conceptual assembly relation	can be a part of a					x	
scale for characteristic relation	can be mapped on scale				x	x	
conceptual possession of aspect relatio	can have as aspect a				x	x	
quantification relation	is quantified on scale as				x	x	x
requirement for possession of aspect	shall have as aspect a				x	x	
possession of aspect relation	has as aspect						x
assembly relation	is a part of						x
classification relation	is classified as a						x
"other" relations (to be specified)	etc.					x	x

An information collection (a 'story') in Gellish English consists of a collection of Gellish expressions presented in one or more Gellish Database tables. Every Gellish Database table has the same table definition as is defined in the document 'The Gellish Database Definition'. Definitions and names of the concepts used in this document can be found in the Gellish Dictionary. That smart dictionary includes a taxonomy (a subtype – supertype hierarchy) and other relations between the concepts. This means that the rule that aspects of supertypes are inherited by all their subtypes, sub-subtypes, etc. For example, a guideline about a concept is also applicable to all its subtypes, such as physical object, characteristic, occurrence, etc.

[1] Documentation about the Gellish language and its application can be found at www.gellish.net, on https://sourceforge.net/apps/trac/gellish/wiki whereas the dictionary /taxonomy content and other documents are available on http://sourceforge.net/project/showfiles.php?group_id=28353.

2. What is correct Gellish English

A correct Gellish English information collection is a collection of Gellish expressions presented in a Gellish Table that complies with the rules and guidelines in this document.

Rule 1: Each expression of a main fact relates only individual things and kinds of things (concepts or classes) that are:

- Either selected from the Gellish dictionary,

or are

- Properly defined subtypes of those concepts,

or are

- Individual things that are properly classified by those kinds of things or their subtypes (via individual classification relations),

whereas the used individual things and kinds of things are related by individual relations that are classified by kinds of relations (relation types) that are also selected from the Gellish dictionary.

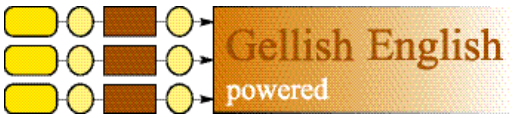
Rule 2: A subtype of a concept is properly defined if the definition satisfies the requirements that are expressed in the document 'Gellish English Dictionary Extension Manual'[2].

Rule 3: An individual thing is properly classified if it has at least one classification relation with a kind of thing that is selected from the Gellish Dictionary or its subtypes.

Rule 4: Any relation shall relate things that play roles of a kind that is required by the kind of relation that classifies the relation.
Example 1: assume that A is a performer of B. This implies that A shall be a physical object, because the 'is a performer of' relation type requires a first role (performer) that can only be played by a physical object. N.B. The second role (performed) is a role that can only be played by an occurrence. So B must be an occurrence.
Example 2: assume that C is a specialization of D. This implies that C is a concept, because the 'is a specialization of' relation requires a first role that is a subtype, which can only be played by a concept (or subtype of concept).

Rule 5: A relation between an aspect and a number may be classified by a scale, in which case the qualitative scale (unit of measure) shall be selected from the Gellish Dictionary or from an extension that is defined in correct Gellish.

Software that is certified by the Gellish Forum as being able to import and/or export data files that are Gellish compliant can be recognized by the following Gellish logo 'Gellish powered' or one of its variants, such as 'Gellish English powered' in the following layout:



[2] See http://sourceforge.net/project/showfiles.php?group_id=28353.

3. Unique identification

3.1. Description of topic

How are things identified and distinguished from each other?

3.2. Guideline

Everything shall have a unique identifier (UID) in Gellish. This identifier may not be changed during the life of the thing. There may not be another thing that has the same identifier, independent of the contexts.

The UID of anything, except a string, is a natural number (integer). A UID for any standard Gellish object is given by the Gellish dictionary manager and is within the range 0 – 30.000.000. To identify objects (or collections of objects) that are unknown in Gellish the range of unique identifiers 1-100 is reserved. Thus, the use of a UID in that range means that the objects are unknown and are requested. The UID's between 101 and 1000 are reserved for testing purposes. Users of the Gellish language may allocate any number above 30.000.000 to identify their proprietary things. If a user wants to avoid overlap with other users, then either the user should ask the Gellish language manager for a reserved range or the user need to make bilateral arrangements with other users with whom he communicates.
A string is a unique sequence of characters in its own right and therefore does not need an additional numeric UID.

3.3. Examples

Standard Gellish things have for example the following UID's:

1225	is classified as a
130206	pump
551564	capacity (mass flow rate)
927838	5.0
570449	dm3/s

User defined things might have for example the following UID's:

40000001	P-6501
40000002	capacity of P-6501
101001001	valve type A
501001002	nominal diameter of valve type A
1501001002	fact 1501001002

4. Classification and specialization

4.1. Description of topic

How is defined what individual things are and how are new concepts defined? How is an entity type or an attribute type in an existing data model represented in Gellish and how is an instance represented?

4.2. Guideline

Every individual thing shall be classified by an explicit relation either with a concept that is selected from the Gellish dictionary or with a user defined concept that is explicitly defined by a user as a proper subtype of a concept that is selected from the Gellish dictionary. Multiple classification is allowed in Gellish.

Every (new) concept shall be defined by a specialization relation with a concept that is either selected from the Gellish dictionary or that is a user defined concept that is defined as a specialization of a concept that is selected from the Gellish dictionary. A concept may be a subtype of more than one supertype and may be a supertype any number of subtypes.
Any qualitative and quantitative concept shall be defined by a qualification relation with a (conceptual) concept. Note that a qualification relation is a specialization of a specialization relation.

4.3. Explanation

An entity type or an attribute type in a conventional data model is equivalent to a concept in Gellish. An instance that is an individual thing is represented in Gellish as a user defined individual thing (that shall be classified by a Gellish concept or by one of its subtypes). An instance that is a qualitative or quantitative concept is represented in Gellish as a qualitative or quantitative concept.
The (implicit) instantiation relation in conventional models is represented explicitly in Gellish by either a classification relation or by a specialization or a qualification relation, depending whether the instance is an individual thing, a (conceptual) concept or a qualitative or quantitative concept.

The consistent application of this rule has as a consequence that the Gellish concepts are all arranged in one consistent specialization hierarchy (= subtype/supertype hierarchy).

The application of this rule has also as consequence that aspects of concepts are inherited by a subtype concept from (all) its supertype concepts. This includes the inheritance of kinds of roles that it can play in relations of a kind. This means that it can be verified whether any relation is of a type that is semantically correct as being allowed according to the definition of that relation type and consistent with other relation types.
For example, if P-5601 is classified as a pump, then it follows from the classification relation that P-6501 is an individual pump, and from the specialization hierarchy above pump it follows that it is an individual physical object and thus only relation types are allowed that are defined for an individual physical object (or for one of its supertypes).

4.4. Examples

P-6501	is classified as a	pump	
capacity of P-6501	is classified as a	capacity (mass flow rate)	
P-6502	is classified as a	cycle-pump	
bicycle pump	is a specialization of	pump	that is intended to inflate tires of bicycles.
583.4 dm3/s	is a qualification of	volume flow rate	
RGB 60, 40,10	is a qualification of	colour	

5. The difference between assembly and collection

5.1. Description of topic

Single things and pluralities need to be distinguished. What is the criterion to determine whether something is a single thing or a plural thing? And when a thing is a part of or an element of a bigger whole, how can it be avoided that it is counted more than once in a bill of materials?

5.2. Guideline

Collection:
A group of items that have no fixed position relative to each other and has no specific role towards the whole forms a collection.
The relation type used to describe the relation between the element and the collection is: 'is an element of'. The inverse relation type is called: 'is a collection of individuals for'.

Assembly:
A group of items that have a fixed position relative to each other and forms an integral part of the whole forms an assembly.
The relation type used to describe the relation between the part and the assembly is: 'is a part of'. The inverse relation type is called: 'is an assembly of'.

An item may be element of more than one collection and/or the same item may be part of more than one assembly. This means that relations are added, but the item is not duplicated. This avoids duplications in bills of material.

Union:
Multiple collections can be united into a collection that is a 'union' of the united collections. A union does not contain duplicates that may appear in the united collections. It uses the relation type 'is united in'.

Note: A collection typically can have as aspects: number of items, (total) mass, (total) price, but not a length of the collection.

5.3. Examples

A collection of individual things:

Left hand object name	Relation type id	Relation type name	Right hand object name
collection of bolts B	1225	is classified as a	collection of items
collection of bolts B	4843	each of which is classified as a	bolt
bolt-1	1227	is an element of	collection of bolts B
bolt-2	1227	is an element of	collection of bolts B
collection of bolts B	2853	is a union of the elements of	collection of nuts and bolts C
collection of nuts N	2853	is a union of the elements of	collection of nuts and bolts C

A collection of classes (kinds of things):

collection of RGB base colours	1224	is a particular	collection of classes
red	4730	is an element in collection of classes	collection of RGB colours
green	4730	is an element in collection of classes	collection of RGB colours
blue	4730	is an element in collection of classes	collection of RGB colours

An assembly of individual things:

wheel-1	1190	is a part of	my car
motor-1	1190	is a part of	my car

A conceptual assembly (possible relation between members of classes):

wheel	1191	can be a part of a	car
motor	1191	can be a part of a	car

6. Aspects of parts with product decomposition

6.1. Description of topic

How should aspects of parts of products be specified when the decomposition of the product is also specified?

6.2. Guideline

Specify an individual thing as being a part of the whole product and allocate the aspect to the individual thing that has the role as part of the whole.

6.3. Examples

wheel-1	1190	is a part of	my car
wheel-1	1727	has aspect	diameter-1

7. Aspects of parts without product decomposition

7.1. Description of topic

How should aspects of parts of products be specified without decomposing the product?

7.2. Guideline

Define a role of the aspect and allocate the role of the aspect to the whole product.

7.3. Explanation

The preferred method is: Define a part and allocate the aspect to the part of the product (see the previous guideline).

In conventional system products are often not decomposed and as a consequence aspects of parts are then recorded as aspects of the whole product. This requires that an aspect of a part is distinguished from others by specifying that the aspect is possessed by the part. This is usually done by adding the name of the part to the name of the aspect. In Gellish such a 'possessed aspect' is a role of an aspect.

7.4. Examples

1. Specify that a bearing of a pump can have a diameter.

A bearing can possess a 'diameter', but in conventional systems the 'bearing diameter' is often recorded as being a possessed aspect of a pump.

The Gellish expressions for such a specification are:

diameter of a bearing	5317	is by definition a role of an aspect of a	bearing
diameter of a bearing	5343	is defined as a possible role of a	diameter
diameter of a bearing	1146	is a specialization of	possessed aspect
diameter of a bearing	5258	can be a role of an aspect of a part of a	pump

It is recommended to ensure that the associated knowledge is also defined. For example, the knowledge that a bearing can have a diameter. This should be done as follows:

bearing	2069	can have as aspect a	diameter

In the Upper Ontology (TOPini) it is already defined that:

possessed aspect	1146	is a specialization of	role

2. Specify the diameter of a driven end bearing of a pump

Driven end bearing is by definition a role of a bearing. So in this case the possessed aspect 'diameter of a driven end bearing' is not specified to be an aspect of a bearing, but of a role of a bearing. Therefore, the Gellish expressions for such a specification are:

diameter of a driven end bearing	is by definition a role of a	diameter
diameter of a driven end bearing	is a specialization of	possessed aspect
diameter of a driven end bearing	is by definition a role of an aspect of a	driven end bearing
pump	can have a role of a part with as possessed aspect a	diameter of a driven end bearing

3. Specify the type of bearing.

type of bearing	is a specialization of	subtype

7.5. Validation method

The relation type definitions specify the nature of the role players that are related. This shall be consistent with the specialization hierarchy of the related objects.
For example, the relation type ‘can have a part with as possessed aspect a’ requires that the first role player is a physical object and that the second role player is a role. Therefore, it should be verified whether ‘pump’ is a subtype of ‘physical object’ and whether ‘diameter of a bearing’ is a subtype of ‘role’.

8. Distinction between subtypes of aspects and roles of aspects

What is the difference between a subtype of an aspect and a role of an aspect? For example: is an internal diameter, a boiling point, etc. a subtype of property or a role of a property?

8.1. Guideline

When a possessor physical object is included in the definition of an aspect/role, then it is a role, else it is an aspect. In other words, if something is by definition possessed by a particular kind of physical object, then it is a role of aspect (= possessed aspect).

8.2. Explanation

Adjectives in names of aspects result in subtypes of aspects. Physical objects in names of aspects result in roles of aspects. The used aspects and physical objects shall have been defined in the dictionary.

8.3. Example

internal diameter	is a specialization of	diameter	(being an aspect)
pipe diameter	is a specialization of	role of aspect	of a diameter when being possessed by a pipe
pipe diameter	is by definition a role of an aspect of a	pipe	
pipe diameter	can be a role of a	diameter	
pipe internal diameter	can be a role of a	internal diameter	

Note: the aspect ‘diameter’ and the physical object ‘pipe’ are already defined in the Gellish dictionary.

9. Can properties have properties?

Properties of properties are not allowed, although properties can be correlated with other properties

9.1. Guideline

Aspects cannot have aspects, except that they can have roles.

Two aspects for which one is thought to be an aspect of the other should both be modeled to be an aspect of the possessing totality. It may be specified however that the aspects are correlated (this correlation causes that people might think of properties of properties). A **correlation** between properties is specified through a correlation relation between the aspects.

9.2. Examples

A temperature limit that is the end of a temperature range. Both have to be specified to be possessed by the same possessor. Then the limit can be defined to be correlated with the range by a relation that indicates for example that the limit is a lower limit of the range.

thermometer	can have as aspect a	temperature range	
thermometer	can have as aspect a	upper temperature limit	
upper temperature limit	is upper boundary of	temperature range	

10. Collections of aspects

Aspects, especially qualitative aspects are often specified to be elements of collections of aspects. Typically such collections are intended to be used as pick-lists or lists of allowed values. Special collections are: ordered lists (pairs, triples, etc.) which can be homogeneous or inhomogeneous.

10.1. Guideline

Allowed nominal diameters for a welding neck pipe flanges according to DIN or according to ASME.

11. Correlations between aspects

11.1. Examples

1. Colours defined in the RGB system

Red, green and blue are qualitative aspects of colour.

Subtypes of red are for example: Magenta-1 = (R=76, G=12, B=14), Magenta-2 = (R=86, G=22, B=24).

Magenta-1		is a qualification of	colour
Magenta-2		is a qualification of	colour
Magenta-1-RGB		is a specialization of	correlation
Magenta-1		is a result of	Magenta-1-RGB
R=76 (76,0,0)		is a coefficient in	Magenta-1-RGB
G=12 (0,12,0)		is a coefficient in	Magenta-1-RGB
B=14 (0,0,14)		is a coefficient in	Magenta-1-RGB

This is a particular case of the general equation:

y = f (a, b, c), where f(a, b, c) is a function or correlation in which a, b, and c are the parameters. Thus the general correlation is expressed in Gellish as follows:

y		is a result of	Magenta-1-RGB
R=76 (76,0,0)		is a coefficient in	Magenta-1-RGB
G=12 (0,12,0)		is a coefficient in	Magenta-1-RGB
B=14 (0,0,14)		is a coefficient in	Magenta-1-RGB

2. Heelheid, Schoonheid en veiligheid.

3. Total width of an assembly is the sum of the widths of some parts.

12. Dictionary and Taxonomy versus knowledge model

12.1. Description of topic

How does a manufacturer's model with its detailed specification (constraints) relate to a concept in a taxonomy model?

Which concepts (including relation types) are used for such descriptions (in these models).

12.2. Guideline

A detailed model, such as a manufacturer's model (or model and size) is a further subtype of a concept in the taxonomy. To distinguish this (sub)typing from the ordinary specialization relation the synonym 'is a type of' can be used. The model will inherit aspects from the higher-level concepts, but in can have additional aspect.

12.3. Explanation

If a thing is defined in a dictionary and included in a taxonomy hierarchy and specified in a knowledge model, then there is still only one thing. It means that the same thing occurs in three different contexts. Therefore, the thing has one and the same identifier (UID).

Views can be distinguished by selecting the context(s) that are relevant and on top of that an application can specify further selections. For example a knowledge model view can be further constrained by specification that only the relations of the type 'can have as aspect a' should be made visible.

The three contexts use the following relation types:

dictionary: 4685 is defined by

taxonomy: 1146 is a specialization of (= is a subtype of)

knowledge model: see table below.

12.4. Examples

Pump in a dictionary, pump in a taxonomy and pump in a knowledge model.

Context	UID	Left hand object name		Relation type name	Right hand object name
dictionary	370,130	pump	4,685	is defined by	rotating equipment item that is capable to increase a pressure and a flow rate of a fluid.
taxonomy	370,130	pump	1,146	is a specialization of	rotating equipment item
knowledge model	370,130	pump		can have as part a	bearing
knowledge model	370,130	pump		can have as aspect a	capacity (volume flow rate)
taxonomy	1,000,001	pump model A	1,146	is a type of	pump
knowledge model	2,000,001	pump model B		can have as aspect a	model A capacity
knowledge model	1,000,002	model A capacity		can be qualified as	10 dm3/s

13. Functions and locations of physical objects in assemblies

13.1. Description of topic

How to describe that a thing has a particular function or location as part in an assembly. This applies to kinds of things as well as individual things.

13.2. Guideline

13.3. Explanation

The function or location of a thing is represented by a role in an occurrence (activity) or the role in a relation.

13.4. Examples

The concepts of the roles of the four wheels of a car need to be present or need to be added to the Gellish Dictionary. When those concepts are available it can be specified that a car shall have four wheels in those four different roles.

The concepts 'front' wheel' and 'left front wheel' are defines as follows:

front wheel	1,146	is a specialization of	role
left front wheel	1,146	is a specialization of	front wheel
front wheel	4,714	can be a role of a	wheel

The requirement that a car shall have a left front wheel is specified as follows:

car	5,237	shall have a part with as role a	left front wheel

Note: in principle the relation type 4,989 'shall have as part a' could be used and interpreted correctly, because from the definition of left front wheel it can be concluded that that is a role of a wheel.

An individual left front wheel is specified as follows:

w-1	1,190	is a part of	c-1
w-1	1,225	is classified as a	wheel
w-1	1,588	has a role as a	left front wheel

14. Type of thing: Requirements to classify things

14.1. Description of topic

On standard forms there is often a field with the name ‘type’. How should that requirement be specified? Is this an aspect of the thing?

14.2. Guideline

Use the relation type ‘shall be classified as a subtype of’ or ‘shall have a part that is classified as a subtype of’ to specify the requirement to classify the item or a part of the item by a subtype of the specified concept.

14.3. Explanation

Fields on standard forms, such as on data sheets, not only specify an object type, but they specify primarily a requirement for a kind of fact. In other words they specify a requirement for a relation of a particular type. Furthermore, the fields do not always specify aspects of the thing about which the sheet provides information. In this case a type is not an aspect of an individual item, but is its classification. Thus the term ‘type’ specifies a requirement for a classification of something by a subtype of a particular concept.

14.4. Examples

Assume that on a ‘pump data sheet’ there is a field with the name ‘type’ and another field with the name ‘bearing type’. The first filed expresses apparently a requirement to fill in the name of a subtype of the concept ‘pump’. A proper value would be ‘centrifugal pump’. The second field expresses apparently a requirement to fill in the name of a subtype of the concept ‘bearing’, whereas that such a bearing is apparently a part of the pump. An example of proper value is ‘ball bearing’.
These requirements are expressed in Gellish as follows:

pump	4,994	shall be classified by a subtype of	pump
pump	5,392	shall have a part that shall be classified by a subtype of	bearing

Note: The latter relation type is a short-cut for the full specification that a pump shall have a bearing and that a bearing shall be classified by a subtype of bearing. This short-cut relation type is available to enable the specification of such a requirement in case no decomposition is supported or wanted.

15. How to add relation types

15.1. Description of topic

If we cannot find a suitable relation type, how can we add a new one?

15.2. Guideline

Identify a relation type that is more generic than the required relation type and verify the suitability of its subtypes.
If not successful, then verify whether the role that the related objects play do appear in the Gellish dictionary. If that is the case, then determine whether there is a suitable relation type that requires that role (or its supertype or subtypes).

If still no suitable relation type is found, then add the required relation type as an extension of the Gellish relation types according to the Guidelines for the Extension of the Gellish language.
This also includes providing feed back to the Gellish Forum with the proposal to include the extension in standard Gellish.

15.3. Explanation

The relation types in Gellish are arranged in a hierarchical network of subtype-supertype relations, just as any other concept in the taxonomy. This also holds for the role types that are required by the relation type and for the object types which members can play such roles. Therefore, these three elements of a relation type definition should be properly defined, so that software can verify the semantic correctness of Gellish expressions.

15.4. Examples

Assume that you want to express that object A is indirectly connected to some physical object.
First it is verified whether a suitable relation type is available by searching on the string elements 'is' and 'connect'. A good browser will find various alternatives, such as 'is connected to', is logically connected to', etc. (the STEPlib Browser of Mi2 is software that is very well suited to support such a search). Assume that these options are not acceptable to you.
A search on 'connected' leads also to 'is connected to' and does not provide other options.

Then a new relation type 'is indirectly connected' can be added as a subtype of the found supertype (connection), whereas the Gellish synonym and inverse synonym can be added as follows:

indirect connection	1,146	is a specialization of	connection	that is …
is indirectly connected to	1,981	is a synonym of	indirect connection	
is indirectly connected with	1,986	is an inverse of	indirect connection	

The above definition implies that an indirect connection inherits two roles, connector and connected, from its supertype, as is defined in the Gellish upper ontology (TOPini). That upper ontology also specifies that these roles can be played by any physical object. Now assume that it is wanted to constrain the allowed role players of the first role to only objects of the type A. Then it is necessary to define that the subtype relation type requires as first role a subtype of 'connected', possibly called 'indirectly connected', whereas such a role can only be played by an object of type A.
This is specified as follows:

indirect connection	1,146	is a specialization of	connection
indirect connection	4,731	requires as first role a	indirectly connected
indirectly connected	1,146	is a specialization of	connected
object type A	4,733	can have a role as a	indirectly connected

A similar subtype role and role player can be defined for the second role.

16. Appendix: List of issues (to be converted into guidelines)

The following issues are requests for guidelines. The capabilities of Gellish are present, but guidelines are required for their proper usage.

16.1. How to distinguish between information about a thing and a thing itself.

How to make a distinction between a 'cloud' of information about a thing and the 'thing itself'? In other words, how to distinguish between a network about a node and the node itself?

16.2. How to specify domains of allowed values

Domain of allowed values of an aspect; especially reduced domains in case of aspects in roles; constraints can be indicated by a minimum, a maximum value, but also by an enumeration, etc. A domain can be within a wider domain (constraints).

16.3. How to specify the discriminating aspects of a subtype and its values.

How to specify a discriminator and its value? This should enable semantic searching.

16.4. How to distinguish between conceptual and qualitative physical objects

How can we distinguish between conceptual physical objects in a taxonomy and specified physical objects as present in product libraries. What is their relation?

Should we distinguish between intrinsic (type determining) aspects and parts on one hand and extrinsic aspects on the other hand (or always roles of aspects)? In other words are objects that have qualitative aspects special ones?

16.5. How to distinguish between classification and instantiation

When to use classification, instantiation, as defined in POM, PLM/PTM and PCM; in general the problem of instantiation of an object and the use of templates (PLM or PTM or Template)

16.6. How should the consistency between implicit and explicit definitions be maintained.

How to ensure consistency of implicit (PDM) and explicit definitions (PCM). For example, if something is defined explicitly by a value of a discriminating aspect, should then the implicit (free text) definition be deleted?

16.7. How to use roles of aspects and roles of physical objects

How to use roles and relations between physical objects and roles. For example: 'input terminal voltage' and its relation to terminal and to voltage and to the possessing physical object.

16.8. When to use qualification and when quantification?

```
is qualified as     (alleen gebruiken voor niet-zuivere getalwaarden)
is quantified as     (alleen gebruiken bij zuivere getalwaarden)
```

16.9. What is the difference between 'has as aspect' and 'has as property'

When to use " has aspect" or "has as property" in view of the fact that ISO 15926 does not use the concept 'aspect'. Is aspect properly positioned in the taxonomy?

16.10. Correlations and their link with 'equation solvers'

16.11. How to describe tables

16.12. How to describe placement

16.13. Inverse relations

A fact can be expressed by a relation that is classified by a relation type indicated by a phrase or its inverse phrase. Only one of the two phrases is required.

The synonym is preferred.

16.14. Cardinalities

'Shall be…' relations need cardinalities to indicate whether the inverse is nevertheless optional.

16.15. Homonyms

Homonyms should be defined in different language communities as their contexts

