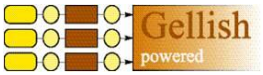


Gellish Modeling Method

Part 2A

Gellish Dictionary/Taxonomy Development Manual

Version 5.3
December 2010

ABSTRACT:

This report documents the methodology for extension of Gellish English, a system independent 'Open Source' language. The definition of Gellish includes its standard relation types, objects and their roles and behaviours, together forming the Gellish Smart Dictionary, including a Dictionary and Taxonomy and Ontology.
Gellish is intended for the storage and exchange of information in a system independent standard way. The Gellish smart dictionary can also be used to customise application systems or in conjunction with Application Protocol ISO 10303-221 or with ISO 15926-2.
The Gellish language definition consists of a set of tables that are compliant with the definition provided in 'Definition of Gellish Databases and Data Exchange Messages'. The Gellish language and the Gellish Data Table are defined on https://www.gellish.net/.
Information expressed in Gellish can be browsed in any Gellish Search Engine and any Gellish enabled application.
The definition of the semantics of the relation types used for the Gellish Smart Dictionary is documented in the Smart Dictionary itself, in particular in the Upper Ontology section (the TOPini part).

KEYWORDS:
artificial language, dictionary, taxonomy, ontology, engineering, standard, classification, class library, thesaurus, keyword, e-business, knowledge base

Document status: Approved
Date: December 2010

This standard is maintained by Gellish.net:
Author: Dr. Ir. Andries van Renssen
- Information and Knowledge Modeling Consultant
Address: Gellish.net
Violiervaart 1
2724VR Zoetermeer, The Netherlands
Tel.: +31-(0)79-3413948
E-mail: andries.vanrenssen@gellish.net

Change history

Version	Date	Modification relative to previous version
5.1	February 2005	Identifiers in tables improved and minor corrections
5.2	February 2006	Clarifications and version management description
5.3	December 2010	Textual improvements

Table of Content

1. Introduction

Gellish is a universal language for data storage, data exchange and data integration. Information, knowledge and requirements that are expressed in Gellish can be combined and integrated without the need for data conversion. The language is computer interpretable and system independent.

Gellish is defined in an electronic Gellish "Smart Dictionary" that can be used by any application system to standardise concept definitions, to use the power of the knowledge and requirements that are expressed in the language. The concepts in the dictionary are arranged as a taxonomy (subtype-supertype hierarchy) which means that facts about concepts are inherited to all their subtypes. The dictionary is further enhanced with the specification of other kinds of facts, so that it can be called an ontology. The dictionary defines concepts, not just words, and includes also concepts thatare denoted by multiple words, which ordinary dictionaries don't. In addition to that the Gellish dictionary defines relation types that can be used to make Gellish expressions (sentences). For example, the relation type <is located in> can be used to express that the Eiffel tower <is located in> Paris. The Gellish language has the semantic expression power to express information about individual things as well as to express knowledge and requirements about kinds of things.
Gellish English is the English variant of Gellish and its electronic dictionary currently contains over 60.000 facts. The dictionary/taxonomy is composed of a number of Domain Dictionaries / Taxonomies (sections) that are integrated, and form specializations of the concepts in the Upper Ontology section.

This document describes how to develop and extent the Domain Dictionaries/Taxonomies and thus how to extent the Gellish English language and how to extent the expression of knowledge about the concepts and thus extent a Gellish English knowledge base. This document also describes how to issue proposals for extension of Gellish English, Gellish Nederlands (Dutch) or any other Gellish variant.

A 'story' in Gellish English consists of a collection of Gellish expressions presented in one or more Gellish Data Tables. The Gellish Data Tables are defined in the document 'Definition of Gellish Databases and Data Exchange Messages' (see www.gellish.net/downloads).

> A correct Gellish English story is a collection of Gellish expressions presented in one or more Gellish Data Tables that only use:
>
> - concepts (individual things and kinds of things / classes) that are selected from a Gellish dictionary/taxonomy,
> - or
>   properly defined subtypes of those kinds of things / classes,
> - or
>   individual things that are properly classified by those kinds of things or their subtypes,
> - whereas the used individual things and kinds of things are related by relations that are classified by kinds of relations (relation types) that are also selected from a Gellish dictionary.

A subtype is properly defined if its definition satisfies the requirements that are expressed in this document.
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.

For guidance on usage of Gellish English see the various parts of the Gellish Modeling Method (see http://shop.gellish.net). Via the Gellish website it Gellish data quality management and certification services are offered.

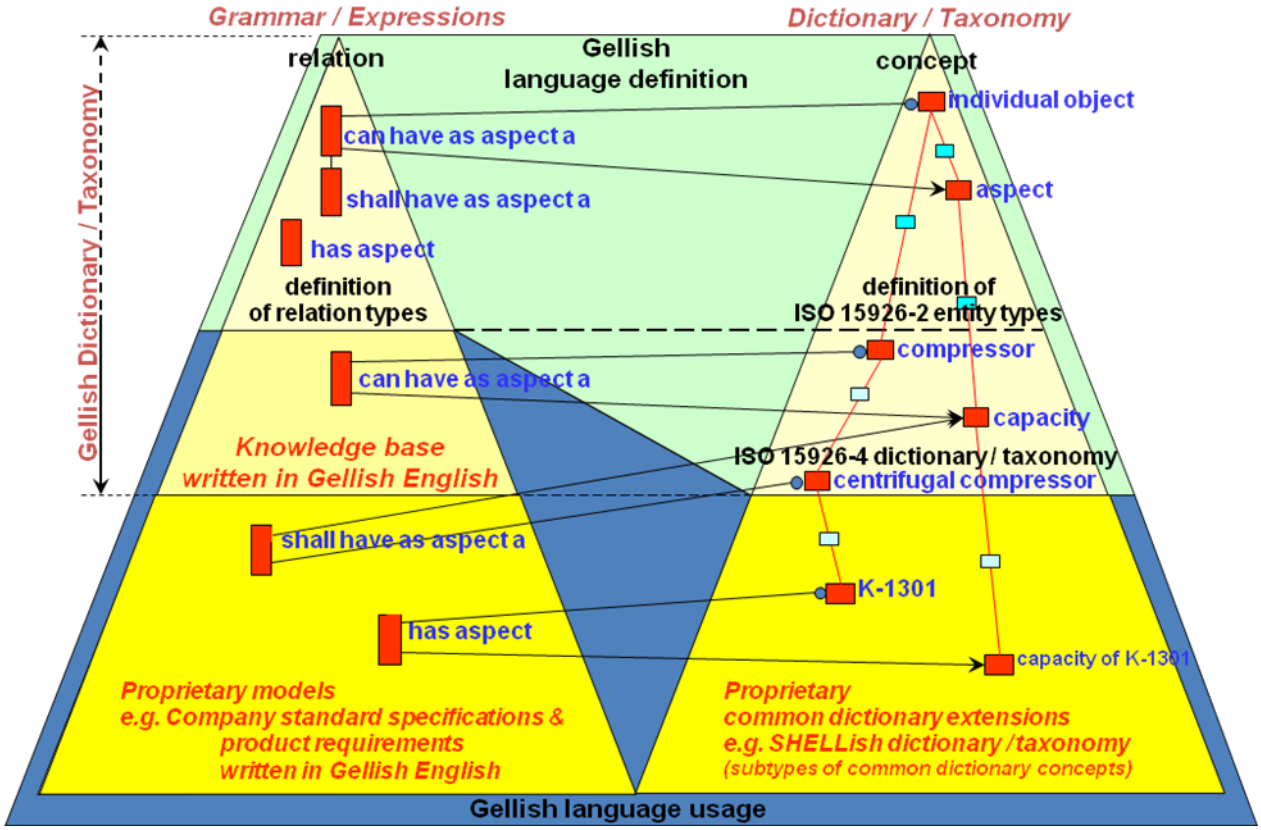
Gellish English is a formal subset of natural English. It is intended to be a common language for electronic information exchange primarily for engineering and business applications. Gellish is an extension and further development of the ISO standard ISO 10303 (STEP) and ISO 15926. A subset of the Gellish Dictionary/Taxonomy is published as ISO 15926-4, a subset of the Gellish relation types will be included in ISO 15926-11. The Gellish Dictionary/Taxonomy includes a lot of the concepts that are standardised in various standards, such as ISO 31, ISO 1000, ISO 1998, ISO 14224, IEC 30, IEC 1131, VDI 3696 and other public sources.

The Gellish language, its dictionary and taxonomy, including also the definition of standard relation types, (together also called the Gellish 'ontology') is intended to support a complete and unambiguous description of facilities, equipment, processes, activities and events, process materials, persons and organizations, and their purpose and behaviour, as well as 'kinds of aspects of objects' and their relationships. The language is intended to standardise the way in which information is stored in computer systems databases and for the way in which information is exchanged between systems.

The definition of those concepts cover the full lifetime of the physical objects, from design and engineering, through installation, operation and maintenance to demolition.

> Note, that the term 'object' in Gellish is a synonym of 'anything', ranging from physical objects to abstractions and aspects of things and also including relationships. In normal practice the term object is often used as a synonym of physical object, but those are not synonyms here.

The following picture illustrates that in Gellish every fact is expressed as a relation between two objects.



The above picture also illustrates:

- Gellish expressions in the upper part illustrate the definition of concepts in the Gellish language, expressions in the lower parts illustrate the usage of the Gellish language for other purposes (expression of knowledge, requirements or information about individual things).
- The definitions of Gellish concepts are themselves expressed in Gellish.

The definition of standard relation types form the Gellish grammar. Those relation types can be used to create expressions, each of which express an atomic fact. The Dictionary/Taxonomy defines the kinds of objects and the kinds of aspects that are required to define product structures with product properties and qualities of any type of object that occurs in various industries, as well as the business processes about them. The Gellish knowledge base contains knowledge about product models and processes, with a focus on facilities and processes in the process industries. The facilities include systems and equipment and their components as well as piping, electrical, instrumentation and control, building, civil and structural components which are applicable in various industries. The dictionary also includes definitions of process materials, construction materials and document types, such as reports and drawings as well as activities, processes, events, persons and organisations.
The objects in the dictionary range from projects and complete facilities up to their detailed constituting components and specifications of them. Furthermore, the definitions cover the full lifetime of facilities, from design and engineering, through fabrication, installation, operation and maintenance to demolition. In addition to data about the hardware, activities, processes the knowledge base also defines the types of roles that objects and organisations can play.

The Gellish language enables the expression of facts and processes regarding any kind of object, especially about design and engineering (of assets), their operation and maintenance with the aim to standardise storage in data bases and exchange of data during data communication. The Gellish language can be used to define public or private extensions of the Gellish dictionary as well as to describe and exchange information about individual company assets in a standard way.

The Gellish dictionary can be used to customise application systems, to create messages for data exchange. If required this can be done in conjunction with the STEP standard Application Protocols of ISO 10303 (and in particular AP221 for Process Plants) and with ISO 15926-2.

Common use of Gellish has as objective to improve commonality and interoperability between Application Protocols and application systems.

Domain experts are invited to participate in *peer groups* to develop the extension of the Domain Dictionary/Taxonomy in their discipline area. Proposals for enhancement and extension of the library can come from any organisation or individual and can be discussed among members of those peer groups. Acceptance or rejection of proposals is the responsibility of groups of domain experts with a final approval by the *Gellish language management team*.

The Gellish language can be used for many purposes, for example:

- Use of the Gellish Smart Dictionary as a standard for the customisation of systems. For that use it is recommended to load the dictionary as the core dictionary of terms in application databases. For example, to customise SAP or CAD systems.
- Use the Gellish Language for the expression of information in data exchange files.
- Use of the Gellish knowledge base, optionally extended with proprietary specifications, for validation of the data quality and completeness in data exchange files or as a template for the creation of data instances.
- Use of Gellish as common language for mapping by translator programs that are intended to convert data from proprietary format into a standard Gellish Data Tables or from a Gellish Data Table into proprietary formats. This supports the possibility to exchange data with systems that store data in proprietary formats.
- Use the Gellish taxonomy as keywords (thesaurus of related terms) in Electronic Document Management Systems (EDMS's) for technical and non-technical documents.
- Use the Gellish language as a starter set to develop your proprietary extensions.

Because of the intended industrial use of Gellish as a standard, a high quality is maintained through stringent quality assurance procedures, although no liability is accepted for any shortcoming in the dictionary/taxonomy or modelling method. Users of the library are advised, for their own benefit, to create extensions and enhancements when they discover any incompleteness or inaccuracy and to propose them for inclusion in the standard Gellish Domain Dictionaries.

The official 'master copy' of the Gellish language definition and knowledge base is documented and distributed electronically, as describe in chapter 6. Versions should bear the date of latest modification of the electronic file as published on the Gellish.net website, otherwise they have no authority.

2. Structure of the Gellish Dictionary/Taxonomy

The Gellish Dictionary/Taxonomy can be regarded as a big tree, or more precisely as a hierarchical network of concepts. The root of the tree is the concept ‘anything’. The tree consists of several branches, each of which is composed of a collection of concepts. Each concept is related by one or more subtype-supertype relations with its direct supertype(s). This means that the concepts together are arranged in a subtype/supertype hierarchy (a taxonomy). The concepts in the hierarchy are also related to each other by various other relations. Those relations are classified by standard relation types.

Examples of collections of concepts are:

- **Kinds of physical objects** that defines the types of equipment and fluids and components of facilities. Furthermore, kinds of **geographic objects, organisations, lifeforms** as well as **documents** and **symbols** on drawings (“ink on paper”).

  The collection of classes of physical objects is sub-divided per discipline.
  Examples of disciplines are: physics, business, engineering, which is subdivided in architectural engineering, systems engineering, process engineering, mechanical engineering, static equipment engineering, rotating equipment engineering, piping engineering, control engineering, electrical engineering, civil engineering, structural engineering, transportation, information engineering, measuring, etc.

- **Kinds of aspects** of physical objects, for example: kinds of properties, qualities, states, process materials and materials of construction and classes for ways of encoding information, such as languages and data formats.
- **Kinds of roles** that objects (can) play in relationships and in occurrences.
- **Kinds of occurrences**, which includes processes, events and activities; basically all verbs.
- **Kinds of information** and **encoding** as recorded on any medium.
- **Kinds of scales (Units of measure)** that can be used to relate properties to numeric values.
- **Kinds of relationships** between ‘objects’. These kinds of relationships determine the semantics of the Gellish language. They form the concepts and meaning used to express facts about ‘objects’.
- **Relations between classes** that describe knowledge that is common for the members of the classes. For example, standard compositions and standard properties of kinds of objects. This includes among others specialisation/generalisation relations, (de)composition and connectability relations and possession of property relations between the kinds of physical objects and their kinds of aspects.

3. The Gellish language

The structure of the Gellish language is based on the basic concepts given in figure 1. The structure of figure 1 described the semantics of an 'atomic' fact. An atomic fact is expressed as a structure of elementary relations. The arrows in figure 1 indicate the direction for reading the sentences that express the elementary facts.
Every (atomic) fact in is expressed in Gellish as a relation between objects, such as object-1 and object-2 that have relation-1 in figure 1. Each of those objects plays a particular role in that relation, indicated as role-1 and role-2 respectively. The objects, the roles and the relation are classified (or conceptualised) by a kind of thing (concept) to provide their meaning. Each concept or kind of thing is defined as a specialization of a wider concept.

For example, the concept of **a specialization relation between classes** is defined as follows:

- Relation-1, called **'is a specialization of',** is classified as a **relation between classes**, in which two objects play a role.
- Object-1 plays a role-1 that is classified as **subtype** in relation-1.
- Object-2 plays a role-2 that is classified as **supertype** in relation-1.
- Object-1 is classified as **class**.
- Object-2 is classified as **class**.

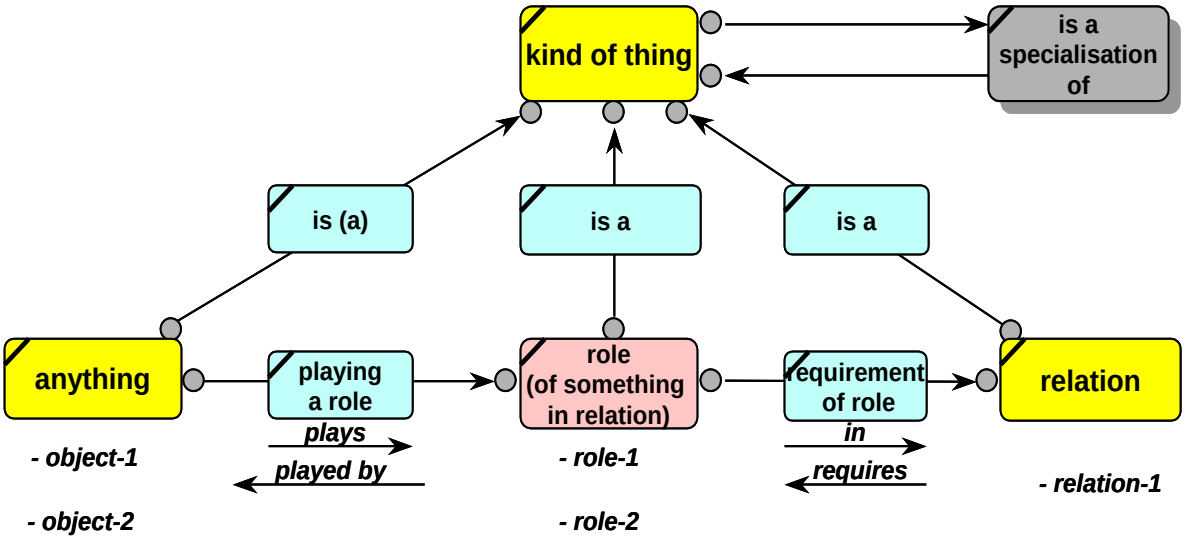


Figure 1, Gellish language - Definition of an object

The definitions of these concepts are given in the Upper Ontology part of the Gellish Dictionary (see the TOPini section).
Paraphrased definitions of concepts that are used in this document are:

- An **'Individual physical object'** is anything that has a real or imaginary individual existence. Anything such as 'P-1', 'John', 'the Eiffel tower', 'the earth', etc., that is recognised as having an individual identity is an individual physical object.

- **'Class'** or **'kind of thing'** is the common nature of all things of that kind.
  Anything that is recognised as being a common nature of a number of things is by definition a kind of thing.
  For example, each of the concepts 'physical object', 'aspect', 'activity', 'person', 'organisation', etc. is a 'kind of thing'. The concept called **'individual object'** is the most generalized supertype of the concepts that classify individual things, such as these examples illustrate.

- A **'specialization'** relation between two kinds expresses the fact that the members of the subtype kind of thing have more aspects in common than the members of the supertype kind of thing.
  For example: a specialization relation between 'air cooler' and 'heat exchanger' expresses the fact that an air cooler is a special kind of heat exchanger.
  A specialisation of a kind of individual thing shall be based on an aspect of individual things that are members of the class.

- A **Unique object id (UID)** in Gellish is a natural number that represents the identity of one 'object'. It is used to uniquely identify an object within a certain context. Every object in Gellish has a UID that is unique in the context of Gellish, except for strings of text (names, definitions, etc.), which shall be unique within an explicit discipline context (the subject area).

- An **'identifier'** in Gellish is a string of text that is used to describe the identification aspect of an object. All classes and individual things in Gellish have at least an English name as identifier. Anything in Gellish has a name that is unique in at least the explicitly indicated subject area.

- A **'definition'** is a string of text that is associated with an 'object' just as a name.

- A **'synonym'** relation indicates that two strings that identify or define an object are related to the same object. The strings may act as a name of the object in different contexts.

- A **Gellish** relation name is a name for a relation type that is close to natural language. It is intended to support the readability of the relations in data exchange messages.
  Most types of relations have also other synonyms especially those that are names of entities in various data model such as in ISO 10303-221 and ISO 15926-2.
  For example:

Name of relation type in Gellish	**Name of relation type in data model terminology**
is part of	assembly of individual things
can be part of a	conceptual assembly of individual things
has as role	role of object
can have a role as a	conceptual role of object
is a synonym of is an abbreviation of is a code for	description of object by encoded information
is a specialization of	specialization of class

Relations can be distinguished in the following kinds:

- **Relations between individual objects.** They represent a description of the reality.
  In general, the names of these relations start in English with "is …" or "has …".
- **Relations between classes.** They generally represent generalised knowledge.
  In general, the names of these relations start in English with "can …" or "is by definition…" (the relation "is a specialization of" is an exception).

- **Relations between individual things and classes**, typically classification relations. They define of which kind an individual thing is. This means that the knowledge about the kind of thing that is contained in the library is applicable to the individual object.

Figure 2 illustrates how knowledge about objects is documented as Gellish expressions. Any instance of each of the 15 relation types in figure 2 is an expression of an atomic fact. The whole Gellish language contains the definition of a few hundred standard relation types.

Instances of the relations with the purple objects describe knowledge about individual objects. Instances of the relations between yellow objects describe general knowledge. The first kind of relations mainly consist of day to day business objects, such as actual plants, activities and business transactions as is documented in operational applications and databases. The second kind of relations mainly consist of knowledge about kinds of objects as is documented in the Gellish knowledge base.

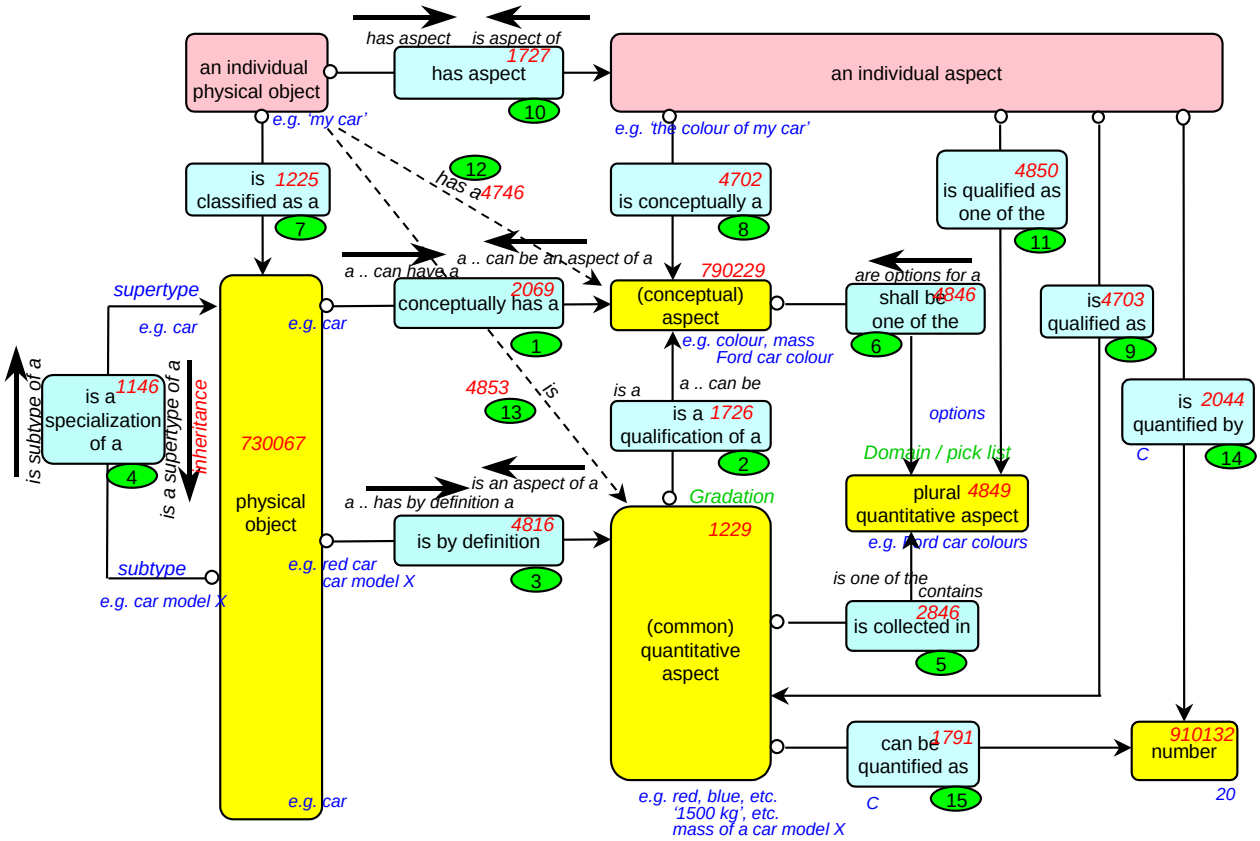


Figure 2, Aspects of objects and of kinds of objects

Knowledge about a kind of thing includes generally known facts such as the fact that a kind of thing conceptually has certain aspects, although the particular qualified value of those aspects may be free.
For example, the fact that "a material has a mass" and the fact that "a car has a colour" (1) are factual knowledge, expressed as a relation between a kind of thing and a conceptual aspect, without specifying the particular qualitative aspects of mass or colour.
More specialized classes (4) have aspects that might be qualified. For example, a 'red car' is by definition 'red' (3). Common valued properties, such as 'red', are qualifications of concepts of property (e.g. 'red' is a 'colour' (2)). Some more specialized classes may have property values that include only a subset of the set of available options that belong to the concept. For example, an RGB signal can only carry the colours red, green and blue (6). Such a domain or pick list is defined by including some property values in a collection of property value classes (5).
Classes are used to classify individual objects (7). Concepts of properties are used to classify individual aspect (8). Common valued properties are used to qualify individual properties (9) of individual objects (10). Other aspects than properties are treated in a similar way.

In Gellish expressions this becomes:

Gellish expressions		Inverse expressions
1.	a car *can have a* colour	a colour *can be an aspect of a* car
2.	red *is a* colour	a colour *can be* red
3.	a red car *is* red	red *is an aspect of a* red car
4.	a red car *is a* car	a car *can be a* red car
5.	red *is one of the* car colours	car colours *contains* red
6.	a Ford car colour *shall be one of the* Ford car colours	car colours *are options for a* car
7.	'my car' *is classified as a* red car	a red car *is a classification of* 'my car'
8.	'C-1' *is conceptually a* colour	colour *is a conceptualization of* 'C-1'
9.	'C-1' *is qualified as* red	red *can be a qualification of* 'C-1'
10.	'my car' *has aspect* 'C-1'	'C-1' *is an aspect of* 'my car'
11.	'C-1' *is qualified as one of the* car colours	car colours *includes the qualifier of* 'C-1'
12.	'my car' *has a* colour	a colour *can be an aspect of* 'my car'
13.	'my car *is* red	red *is an aspect of* 'my car'
14.	'T-1' *is quantified by* 20 *on scale* C	20 *on scale* C *is a quantification of* 'T-1'
15.	'20 C' *can be quantified as* 20 *on scale* C	20 *on scale* C *can be a quantification of* '20C'

Other terms for types of relations used in Gellish and their synonym names are for example:

Relation type name in Gellish	Relation type name in a data model
can have as aspect a	conceptual possession of kind of aspect by object
has aspect	possession of aspect by object
is classified as	classification of individual object
can be a role of a	common role for members of a class
can be a role for	common role for object in activity
is role of	role for individual object
can be involved in a	common involvement of object in activity
can be performer of a	common capability of object to perform kind of activity

Examples of the use of these kinds of relations are:

centrifugal pump	is a specialization of	pump
centrifugal pump	is defined as a	pump which applies the centrifugal principle
impeller	can be part of a	centrifugal pump
driven end bearing	can be a role of a	bearing
bearing driven end	is a synonym of	driven end bearing
impeller	can have as aspect a	diameter

Many objects in facilities are composed of items that conform to '**standard parts**' that are defined in private or public **standards** published by standards bodies, such as ISO, ASTM, DIN, BSI, etc. These standard parts in fact define types of parts and therefore they are also defined as (specialised) classes. The Gellish Smart Dictionary includes the content of some of those standards in a separate section.

Company specific **catalogue items** or other internal company standards are in fact further specialisations of classes that are defined in Gellish. To be proper Gellish, these company specific product types shall be defined as specialisations of existing Gellish concepts, using the same conventions as is required to define new classes in Gellish.
Gellish intends to provide a sufficient set of definitions of kinds of objects and kinds of properties and other aspects to enable companies to define their catalogue items.
Gellish is also suitable for electronic storage and exchange of those standards and catalogue items, including the product structure of those items. Expression of those items in Gellish would allow customers to electronically search and select items that satisfy their requirements and to copy the details of components electronically from those catalogues into their design and maintenance systems, provided the proper access authority is given to them.

Objects can play roles in activities. Gellish contains definitions of kinds of physical objects, as well as kinds of roles (way of being involved in occurrences) and kinds of occurrences (activities, processes and events).
Kinds of roles that kinds of objects can play, as well as kinds of roles recognised for kinds of occurrences are also included in the library.

The model for that is given in figure 3.

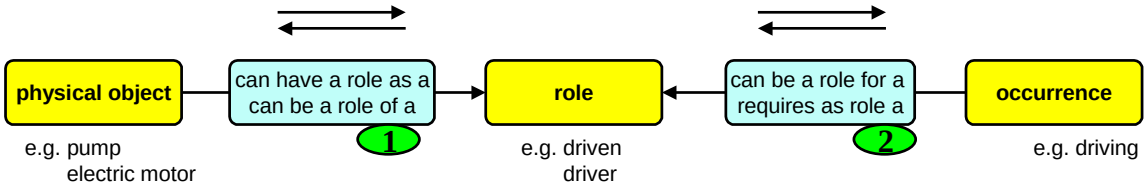


Figure 3, Roles of Objects in Activities

Other examples of roles of physical objects in activities are:

batch of fluid	can have a role as a	feed
batch of fluid	can have a role as a	top product
batch of fluid	can have a role as a	bottom product
distillation	is a specialization of	process
feed	is a specialization of	input
top product	is a specialization of	output
bottom product	is a specialization of	output
feed	can be a role for	distillation
top product	can be a role for	distillation
bottom product	can be a role for	distillation
batch of fluid	can have a role as	pumped
pumped	can be a role for	pumping (to pump)

4. Usage of Gellish

Gellish can be used for example to customise or configure existing (legacy) design systems or logistics systems, such as CAD systems, SAP, etc. Sometimes users want to make only a subset of the Gellish Dictionary visible, in which case it is recommended to keep the whole library in the background in order to maintain the inheritance of aspects. This use is than limited to the use of Gellish as a standard **dictionary**, **thesaurus** or **taxonomy**.

The library can also be used in conjunction with other standards. This holds in particular for parts of ISO 10303 (STEP). Extensions of Gellish can be added as and when required. The procedure for that is described in this manual.

Gellish can also be used to standardise the data structure for storage of data in new systems. Then it is used as a **standard data model**.

Gellish can be used as a standard for **data exchange,** in particular in combination with the Gellish Database Format. For example to exchange data about facilities designed during investment projects and used during procurement, construction, operation and maintenance. It can also be used for the exchange of business messages, such as enquiries, orders and confirmation messages. In such cases Gellish is used to define the **common terminology** and the common semantics (meaning) of the 'sentences' in the Gellish **language.** This means that it can be used by the projects or organisations to describe data about their assets, catalogues and internal standards.

The use of Gellish will have additional benefits for databases that make an explicit distinction between information about an individual object and information about a kind of object (e.g. between data about item P-10 and data about the kind 'pump'). Similarly, there are more benefits possible for systems that make a distinction between individual properties and kinds of properties, etc. This distinction is also made in ISO 10303-221 and in ISO 15926-2. For such databases it is easier to use Gellish as an 'initial fill' of instances of classes that are loaded into the database before an application that operates on that database is made operational.

Another use of Gellish is as **common reference for mapping** of data by translator programs that are intended to convert data from proprietary format into standard data exchange files or from such files into proprietary formats, so that the data can be stored in proprietary systems and can be mailed and imported or exported as standardised files.

The Gellish Smart Dictionary can also be used as **intelligent keywords** (thesaurus terms) in Electronic Document Management Systems (EDMS's) for documents and files such as engineering drawings, reports and equipment documentation. Because of the consistent specialisation hierarchy of Gellish and the explicit other relations the search of documents can be significantly enhanced. For example, Gellish makes it possible not only to search on the given keyword itself, but also on all the subtypes of a given keyword.

Furthermore Gellish can be used to define **standard parts** and **catalogue items** as further specialisations of the existing classes. For example, an electrical cable manufacturer can extent Gellish with a new class called a low voltage electric cable, which is defined by a number of aspects. The various types of low voltage electric cables that he can deliver (according to his catalogue) are further specialisations of this low voltage cable. These extensions will not become part of Gellish, but they can be delivered as a Gellish extension to his potential customers, who can include both Gellish and these extensions in their system and use his own software or a standard Gellish Browser to find also the manufacturer's catalogue items lower in the hierarchy.

Whenever **extensions** to the language terms or expressions (relation types) are required, then it is recommended to define 'private' project or organisation specific extensions according to the same methodology as described in this document. Private extensions that are of general interest and that are fed back as proposals for extension of Gellish are highly appreciated.

4.1 Example usage: Validation of designs

Classification of an individual object is simply done by defining an atomic relation between the individual object and the appropriate class in Gellish (typically this is a relation between their unique identifiers, their UID's).
For example, P-101 is an individual physical object (a specific instance). It can be classified as 'centrifugal pump', where 'centrifugal pump' is a standard kind of physical object in Gellish. The standard class is usually not just a concept with a definition, but usually it is a concept associated with related concepts, which relations belong the definition of the common aspects of "well formed" members of that class, including the degrees of freedom for the related aspects, defined among others by their 'cardinalities' (allowed minimum and maximum number of occurrences).
This is illustrated in figure 4.

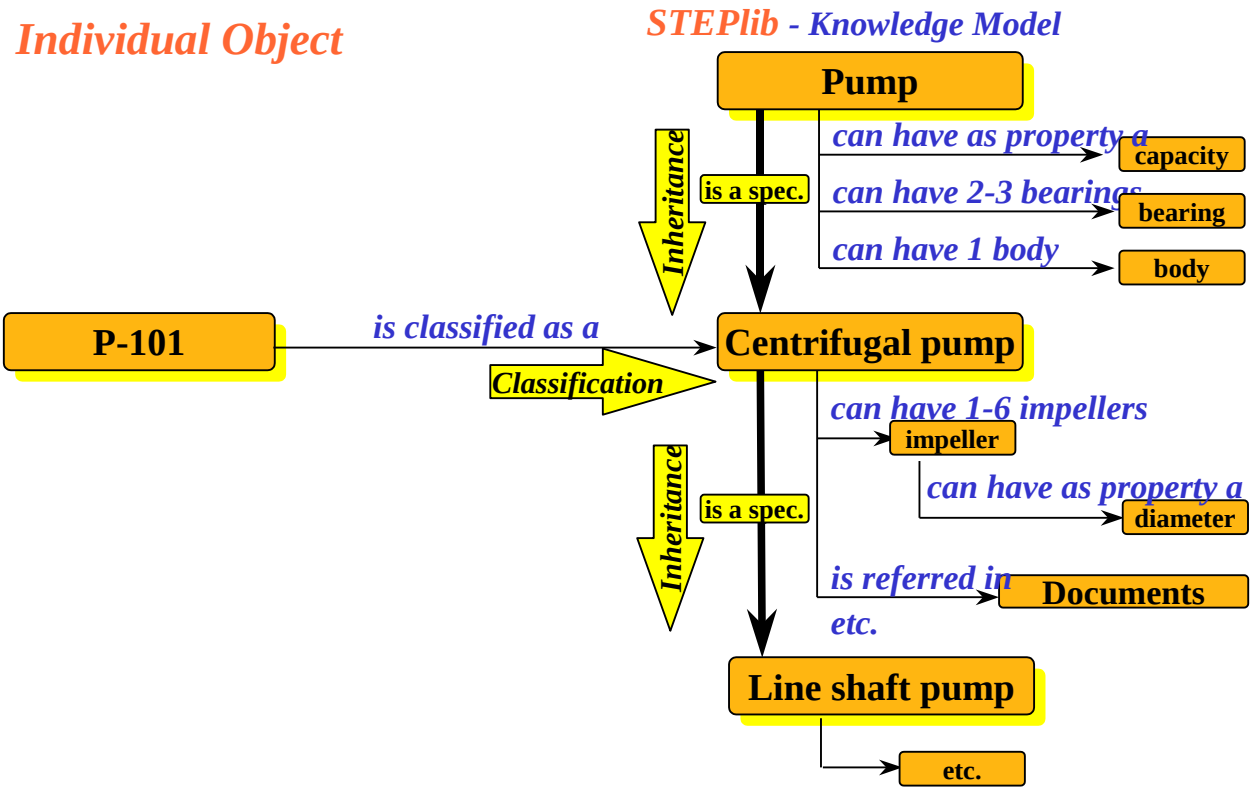


Figure 4, Explicit classification of an individual object

The explicit distinction between an individual thing and a class makes it possible to hold information and knowledge about a class and its behaviour separate from information about individual objects. This means that all objects that belong to a class (are classified as members) are assumed to have properties and behaviour that are common to the members of that class.

This can be used to verify as design as is illustrated in figure 5: **Verification of a design.**

If an object is classified by a person to belong to a class and if the properties of that individual object are known, then those properties can be validated by computer software against the properties of the class to judge whether the object is a proper member of the class.
The verification of the quality and quantity of aspects of an individual designed object is illustrated in figure 5.

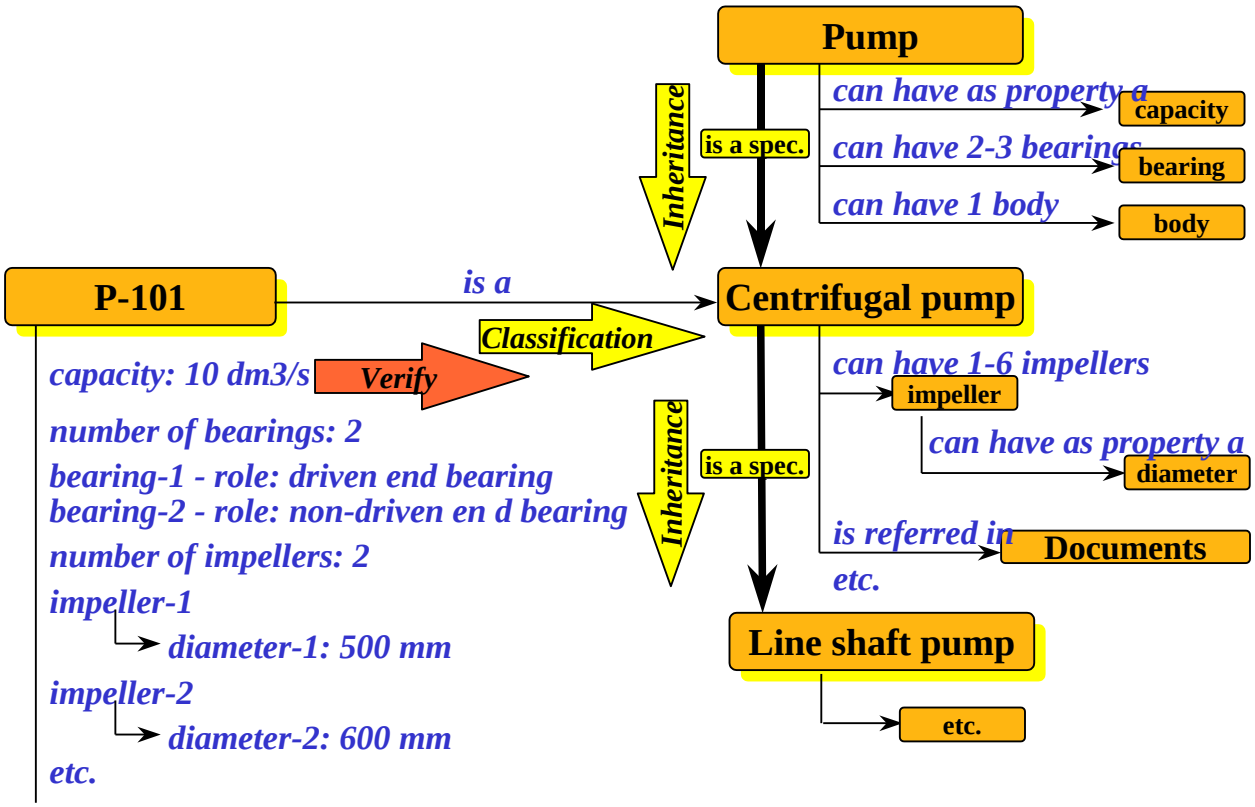


Figure 5, Verification of a design

4.2 Example usage: Creation of a template by allocation of properties to an individual thing.

If an object is classified by a person to belong to a class, but the properties of the individual object are not yet known (as is typically the case when starting a design), then the properties of the class can be ‘allocated’ to the individual thing.
This procedure is a flexible equivalent for the definition of templates, because when the knowledge about the kind of objects is defined, extended or enhanced, then software that uses the knowledge can automatically derive extended and improved ‘templates’ from the class definition.
It should be noted that in addition to the aspects that are common for the members of the class, the individual objects can have their own additional properties that are beyond the definition of the class and that are not shared with other individual things that belong to the class.
This allocation of aspects to individual objects is illustrated in figure 6.

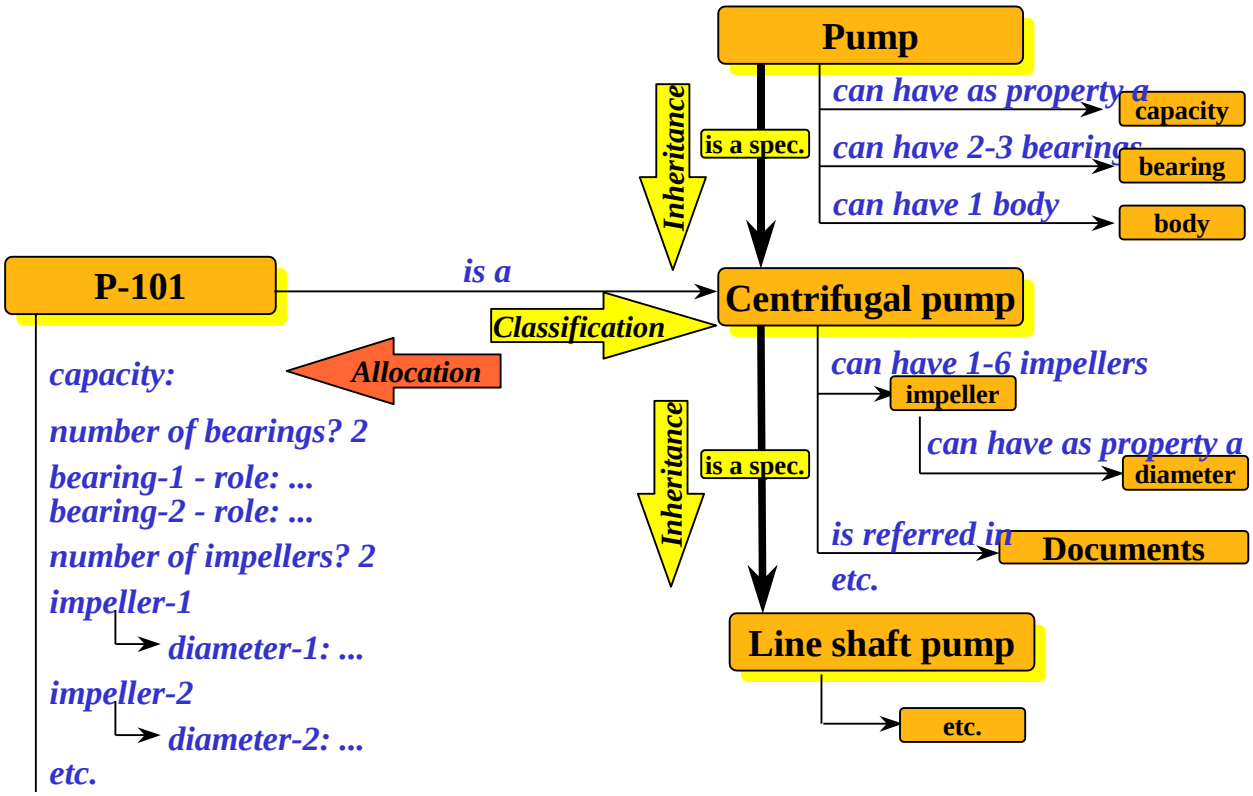


Figure 6, Allocation of properties to an individual object

4.3 Automatic classification

Another possible use of a class is to automatically classify objects whose aspects are observed. This might be applied in laboratories to classify samples. However, in practice usually a sample is tested against the specifications that define a particular product class. This means that in practice automatic classification will hardly be used, but verification of compliance to the class criteria will usually be done instead.

4.4 Example usage: a product model of an individual object

In most of the above usages the data in Gellish is used to classify individual designed physical objects, fabricated physical objects and typical design and to classify their roles and properties and the processes in which they are involved. In other cases it is used to specialise classes by definition of standard parts or catalogue items
Such a definition of individual objects is called the definition of a product model for the object. The main relation used to define what an item is, is the classification relation. This should preferably be done by recording an explicit classification relation between each individual item and an appropriate class in the library.

An example of the way in which this can be done is illustrated in table 1. That example is the definition of nozzle K1A and manhole A-1, both of which are part of vessel V-6060 in unit 6000. Every line in the table describes a fact by defining a relation between to things. The columns 'name of relation type' contain standard types of relations from the Gellish Dictionary and the columns 'UID of right hand object' and 'Name of right hand object' contain UID's of concepts and standard names of concepts from the Gellish Dictionary. The other text in table 1 is user defined and created as part of the design process for V-6060.
The tables 1a and 1b are (parts of) a standard Gellish Data Table , which can be used to define an individual thing of any kind.

UID of left hand object	Name of left hand object	UID of fact	Name of relation type	UID of right hand object (Gellish UID)	Name of right hand object (kind of physical object from Gellish Dictionary)
1	unit 6000	1101	is classified as a	160104	process unit
2	V-6060	1102	is classified as a	520243	vessel
3	head-1	1103	is classified as a	520229	torispherical head
3	head-1	1208	is qualified as	551534	seamless
4	K1A	1104	is classified as a	10298	nozzle
4	K1A	1213	has a role as	13135	flush connection
5	flange-1	1105	is classified as a	340160	flange
6	reducer-1	1106	is classified as a	10018	concentric reducer
7	neck-1	1107	is classified as a	520164	pipe neck
8	collar-1	1108	is classified as a	520128	insulation collar
9	K1A-flanged end	1109	is classified as a	551542	nozzle projection
10	A-1	1110	is classified as a	520146	manhole
11	flange of A1	1111	is classified as a	340160	flange
12	cover of A1	1112	is classified as a	520331	cover
13	neck of A1	1113	is classified as a	520164	neck
14	collar of A1	1114	is classified as a	520128	insulation collar
15	davit of A1	1115	is classified as a	520076	davit
16	bolt for cover A1	1116	is classified as a	340081	stud bolt
17	nut for cover A1	1117	is classified as a	340055	nut
18	gasket for cover A1	1118	is classified as a	340039	gasket
12	cover of A1	1119	is classified as a	520331	cover
19	shell-1	1120	is classified as a	520204	shell
10	A-1	1121	is classified as a	520146	manhole
20	weld-1	1122	is classified as a	340091	weld

Table 1a, Gellish Data Table with classifications of physical objects - Vessel V-6060

UID of left hand object	Name of left hand object	UID of fact	Name of relation type	UID of right hand object	Name of right hand object
1	unit 6000	2101	is a part of	?	?
2	V-6060	2102	is a part of	1	unit 6000
3	head-1	2103	is a part of	2	V-6060
4	K1A	2104	is a part of	2	V-6060
5	flange-1	2105	is a part of	4	K1A
6	reducer-1	2106	is a part of	4	K1A
7	neck-1	2107	is a part of	4	K1A
8	collar-1	2108	is a part of	4	K1A
9	K1A-flanged end	2109	is a part of	5	K1A
10	A-1	2110	is a part of	8	V-6060
11	flange of A1	2111	is a part of	10	A-1
12	cover of A1	2112	is a part of	10	A-1
13	neck of A1	2113	is a part of	10	A-1
14	collar of A1	2114	is a part of	10	A-1
15	davit of A1	2115	is a part of	10	A-1
16	bolt for cover A1	2116	is connection material in	2119	*connection 2119*
17	nut for cover A1	2117	is connection material in	2119	*connection 2119*
18	gasket for cover A1	2118	is connection material in	2119	*connection 2119*
12	cover of A1	2119	is connected with	11	flange-A1
19	shell-1	2120	is a part of	2	V-6060
10	A-1	2121	is connected with	19	shell-1

20	weld-1	2122	is used in	2121	*connection A1-shell-1*

Table 1b, Decomposition and connection of physical objects - Vessel V-6060

The properties and other aspects of individual items can be described in a similar way, by using the standard kinds of properties and kinds of documents from the Gellish Dictionary. This is illustrated in table 2 for head-1 and nozzle K1A of vessel V-6060 from table 1.

Left hand object id	Left hand object name	Fact id	Relation type name	Right hand object id	Right hand object name	UID of UoM	UoM
3	head-1	201	has aspect	101	d-1		
3	head-1	202	has aspect	102	r-1		
3	head-1	203	has aspect	103	r-2		
3	head-1	204	has aspect	104	d-2		
3	head-1	205	has aspect	105	d-3		
3	head-1	206	has aspect	106	h-1		
3	head-1	207	is described via	108	st-1		
4	K1A	209	has aspect	110	nd-1		
4	K1A	210	has aspect	111	rat-1		
5	flange-1	211	is described via	115	st-2		
5	flange-1	212	has aspect	116	nd-2		
101	d-1	213	is classified as a	550206	external diameter		
102	r-1	214	is classified as a	550325	crown radius		
103	r-2	215	is classified as a	550361	knuckle radius		
104	d-2	216	is classified as a	550454	thickness before forming		
105	d-3	217	is classified as a	550967	minimum thickness after forming		
106	h-1	218	is classified as a	550326	cylindrical height		
108	st-1	219	is classified as a	910181	fabrication standard		
110	nd-1	220	is classified as a	551563	nominal diameter		
111	rat-1	221	is classified as a	551396	rating		
115	st-2	222	is classified as a	910181	fabrication standard		
116	nd-2	223	is classified as a	551563	nominal diameter		
101	d-1	224	is quantified as	922467	4120	570423	mm
102	r-1	225	is quantified as	922385	3300	570423	mm
103	r-2	226	is quantified as	921827	635	570423	mm
104	d-2	227	is quantified as	920265	12	570423	mm
105	d-3	228	is quantified as	920359	10.3	570423	mm
106	h-1	229	is quantified as	920293	40	570423	mm
108	st-1	230	is qualified as	121	DIN 28013		
110	nd-1	231	is qualified as	584945	DN 40 x 20		
111	rat-1	232	is qualified as	586703	150 psig		
115	st-2	233	is qualified as	586534	ASME B16.5 specification		
116	nd-2	234	is qualified as	585014	3/4"		

Table 2, Properties and Definitions of Items - Parts of vessel V-6060

5. Definition and extension of Gellish concepts

5.1 What is a class?

Basic concept of the Gellish language are the concepts ‘individual thing’, ‘kind of thing’ (‘class’) and ‘classification relation’. Classification relations are used to classify individual things by classes. Therefore, we first discuss: what is the definition of a ‘kind of thing’ (or ‘class’)

A class or kind of thing is something that is the common nature of a number of things. In other words, it is a category of which the members commonly have a number of similarities, such as aspects or composition.

For example, the class ‘centrifugal pump’ is defined as: a pump that is intended to increase the pressure of a liquid by means of a rotating impeller.

The definition can also be described as follows (a formal definition is included in the Gellish Dictionary):

- a class is a commonality between things, rather than an individual thing.
A class has an explicit or implied basis for inclusion or exclusion from being member of the class.

In other words, a class has a definition that implicitly or explicitly defines constraints that form a basis to determine whether a thing belongs to or does not belong to that class.
These criteria for inclusion or exclusion can relate to all aspects of things, including their behaviour and their composition.

A class (kind of thing) can classify a *single thing*, but it can also classify a *plural thing*. The difference is that the elements in a collection (a plural thing) don't have a particular position relative to each other, whereas the part of a single assembly do have a particular position relative to each other, typically expressed by a connection relation.
Single objects does not need to be a *composite object (assembly)* but can also be a *single component object*.
For example, a flange is a single component object and a pump is a composite object, consisting of many parts.
A composite object can be either an **assembled object** or an **arranged object**. The difference is that the elements in an arranged object don't have connection associations with each other, whereas in an assembly the parts are normally connected. For example, a valve is an assembled object, a village is an arrangement of houses and a stock of valves is a collection of things.

N.B. There is a difference between a **class** and a **typical individual** thing. Typical individual things are individual things that have a particular status of ‘being typical’ (for others). They are normally defined for the purpose of reference or for copying of their specifications when an equivalent specific individual is created. For example, a typical individual object can be a prototype individual, that can be intended as a “template” or “master copy” for the derivation of one or more specific individual objects, such as a (typical) symbol in a symbol library. A typical individual object is defined in a similar way as a specific individual. On the other hand, a class is an abstraction that cannot have a specific location, nor can it have a connection to a specific individual object.

5.2 Addition of a new concept (class) to Gellish

A new concept, which is referred to by a new unique Gellish identifier (UID), is properly defined and added to the Gellish language by completion of a single line in a Gellish Database. That line shall contain the following information (that expresses a main fact and its auxiliary facts):

1. A main fact expressed by an explicit specialization relation (‘is specialization of’) between the defined concept and its (existing) direct supertype concept (class), which expresses that the new concept is a subtype of the existing concept.
   The consistent application of this way of definition causes that the concepts are arranged in a hierarchical network. That network is not a pure tree, but a network, due to the fact that a concept can be a subtype of more than one supertype concept.
2. A textual description that is related to the class and that is a text string that expresses in which respect the subtype is specialized and by what aspects it is distinguished from its ‘sister’ subtypes.
3. A (preferred) name, being an alphanumeric string that expresses how the concept is called in a particular language and within a particular language community (such as a discipline).
4. A language context that specifies in which language the name (and the definition) is expressed.
5. A language community that provides the context in which this name of the concept is specified as the preferred name and in which context this name is unique (within this context there are no homonyms).
6. Some other auxiliary facts that complete a line in a Gellish Database table.

For example, a cooling system is a system that is intended to be suitable to cool some kinds of things. The concept ‘cooling system’ with this definition can be added to the Gellish Smart Dictionary by specification of a line in a Gellish Database table of which the core is as follows:

UID-1	Left hand object name	Fact ID	Relation type ID	Relation type name	UID-2	Right hand object name	Description
1	cooling system	2	1146	is a specialization of	3	system	intended to be suitable to cool some kinds of things.

The absolute supertype of all classes, which is the top of the hierarchy, is the concept **‘anything’ (or ‘object’)**. Close to this top the following classes (concepts) are defined, each of which is further specialised as far as required by the application domain:

- **physical object**, which is the sub-top of kinds of objects that satisfy the laws of physics. This includes, solid items, fluids, waves, spaces, geographic objects, signals, organisations, etc.
  A special subtype of physical object is a class that classifies objects that are alive, called **lifeform**. This includes organisms and persons, being physical objects that are alive**.**
- **occurrence (activity**), which is the sub-top of all kinds of events and happenings that occur, but also of static occurrences such as support, storage, etc.
- **role,** which is the sub-top of kinds of roles of objects in activities and roles of objects in relations or associations (facts),
- **aspect**, being further specialized into **characteristic** and **property** (physical quantities), **quality, state, phase** and **‘substance’** and all their detailed subtypes.
  These aspects are intended to characterise physical objects and activities.
  Another specialization of aspect is the **encoding aspect**. This is a sub-top of classes according to which information can be encoded, such as languages, kinds of identifiers and all kind of electronic coding formats. It concerns the ‘what’ of the information, usually described as text, whereas the ‘how’ it is recorded or presented is a physical object (often ink on paper or light on a screen).
- **relation** (‘is related to’), being the sub-top that is specialized into a large number of standard relation types that define the semantics of the Gellish grammar. Examples of specializations of relation are: classification relation, assembly relation , specialization of class, possession of aspect, etc.

- **information,** which represents ‘meaning’ irrespective of the way the meaning is expressed. For example, irrespective of the language, the encoding system and the format used.

Figure 7 illustrates the top of the Gellish Smart Dictionary or specialisation hierarchy with examples of further subtypes of the above classes (the formal hierarchy is slightly different).

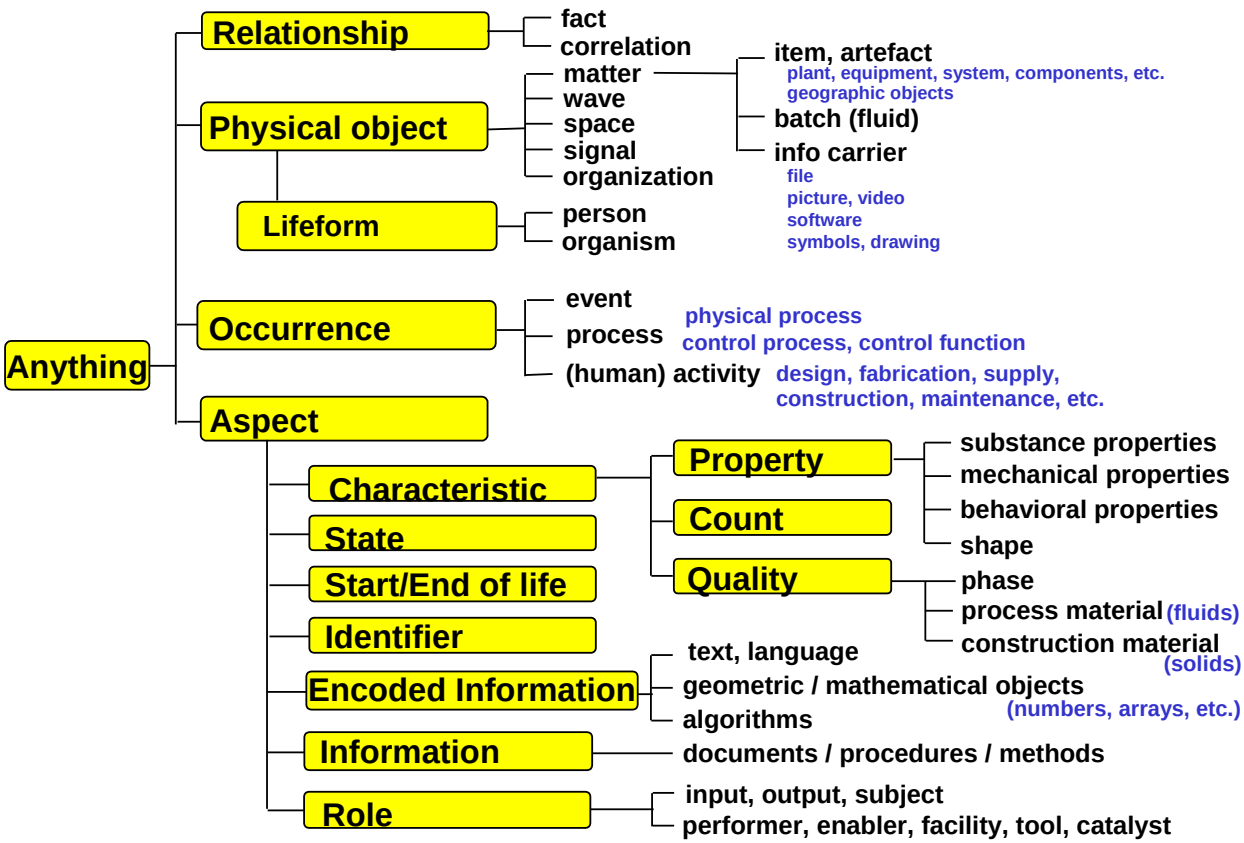


Figure 7, Structure of the Gellish Smart Dictionary

5.2.1 Collections of classes

The Gellish language is suitable to define any kind of class as is illustrated in Figure 8.

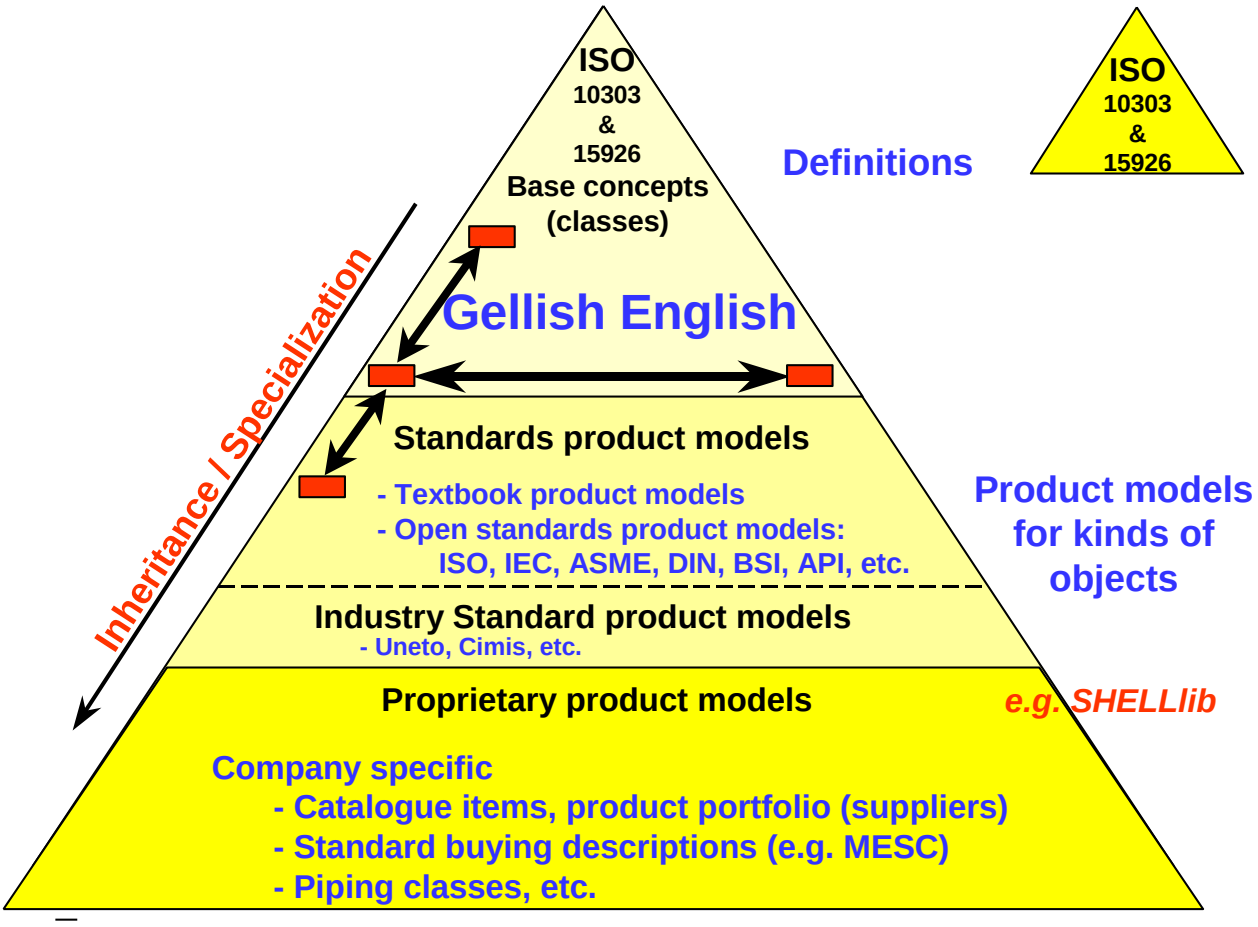


Figure 8, Collections of classes

Figure 8 illustrates that there is a collection of classes that can be regarded as **base concepts** (or foundation classes) and a collection of classes that are **standardised product models,** which have **qualified aspects**. Furthermore there are proprietary classes that are typically owned by companies and are known as company specific **catalogue items** or **proprietary product models**.

The **base concepts** form the basic Gellish Smart Dictionary. They are classes with names such as pipe, pump, floor, etc., without a specification of their aspects. Base concepts includes also classes which names are composed of more than one word (at least in English), such as centrifugal pump, long radius bend, etc. The base concepts also include all the standard Gellish relation types, such as classification ('is classified as a'), composition, collection, connection and possession of aspect relations.

Base concept are usually not specialised on the basis of a material of construction, because all possible combinations between each kind of physical object and each possible kind of substance would lead to a combinatorial explosion of the number of classes.
For example, 'stainless steel pipe' is not a recorded base concept, although it may be recorded as a standardised subtype (e.g. because there is an ASME standard about it).

Note: an individual item can always be classified as pipe together with a material of construction as follows:

- object	#1	is classified as a	pipe
- object	#1	is made of	stainless steel

The category of **standardised product models** actually does not consist of just classes, but they are 'structures' or related expressions that express facts about classes. This category is formed by the classes with relations that describe the objects that are standardised by standardisation bodies, such as ISO, IEC, ANSI, DIN, BSI, etc.
These standardised product models are defined by much more types of associations such as the possession of specific property values, material of construction, fabrication standards, etc. Therefore, their definition requires a much deeper understanding of the Gellish language. The

necessary information for that is not given in this guide, but should be obtained from the full Gellish language definition, or from other documentation, such as the Gellish Application Handbook.

These classes for standardised items will only be included in the Gellish Smart Dictionary / Knowledge Base when they are defined and published by standards bodies. For example, standard pipes as defined by ANSI.

Note: an instance of such a class is a specialisation of pipe and can be documented in Gellish through the explicit associations that define its aspects, such as its nominal diameter, schedule, fabrication standard, etc.
For example, a pipe with a nominal diameter of 6", should be defined as follows:

- object	#1	'6" pipe'	is a specialisation of		pipe
- object	#1		has aspect	#2	ND 6"
- object	#2		is classified as		nominal diameter
- object	#2		is quantified as		6 inch

In other words, the name '6" pipe' is only a name for such a class. A computer cannot derive property values from such a name

Company specific product models, also when they are 'standardised' within the company are modelled according to the same methodology as the classes defined by standardisation bodies. However, company specific classes are proprietary classes and will not be included in standard Gellish. They can be developed, maintained and managed by companies as 'private extensions' to Gellish. Those classes can be exchanged with business partners in the context of projects or of individual exchanges of data on the basis of those additional classes.

The Gellish language is intended also to support the description of such proprietary product models. Data exchange would be simplified significantly when **standardised objects** and (**buyers and suppliers**) **catalogue items** would become available as Gellish Databases.
However, the definition of those product models is the responsibility of **standards bodies** and of individual companies. The Gellish Knowledge Base will only contain a limited collection of standardised product models.

The Gellish language has some special subtypes of 'class', called '**conceptual class**' and '**qualitative class**'. They classify classes and not individual things. A conceptual class is a class that is not qualified or quantified. A qualitative class is a class that is qualified or quantified. For example, the concepts 'colour' and 'temperature' are unqualified conceptual classes, whereas the concepts 'red' and '37 degree C' are qualitative classes.

Another kind of class is called '**plural class**' or '**collection of classes**'. This class and its subtypes are intended to classify collections. For example, 'ASME classes' is a particular collection of classes that is not defined on the basis an aspect of individual items, not of the classes themselves, but only on the basis of the fact that the ASME organisation has defined the class.
Collections of classes are useful for administrative reasons to collect classes on the basis of their origin or custodian or on the basis of a prime discipline that defines or primarily uses them.

5.2.2 Kinds of relations

Among the concepts in all categories there are not only specialization relations, but also many other kinds of relations. For example, for a kind of physical object there are different relation types that are used to defined what its properties are, what its normal composition is, which roles it can play, etc.
Therefore, the Gellish Smart Dictionary includes a hierarchy of kinds of relations, also called relation types.

Kinds of relations include:

- Relation types that relate classes. They express knowledge about possible relations between individual objects. This includes the definition of the semantics of Gellish, as well as 'product models' that express social, business, technological and scientific knowledge.

- Relation types that can be used to classify relations between individual objects. They express real of imaginary facts about individual objects.
- Relation types that can be used to classify relations between individual objects and classes. They express knowledge about individual objects. For example, the fact that an individual object belongs to a certain class or the fact that it can be used for a kind of purpose.

The application of these standard relation types has as a result that the Gellish knowledge base contains a network of relations between concepts that documents knowledge about the concepts. Each concept acts as a node in the network of relations. A 'cloud' of relations around such a node in the network defines a "knowledge model" that specifies the aspects that are normally true for a 'well formed' object that is classified by the concept (class). Figure 9 illustrates such a knowledge model by specifying knowledge about a vessel.

Figure 9, Example of a knowledge model with relations between concepts (classes)

5.2.3 Kinds of Physical objects

To design facilities in detail and to operate and maintain them, it is required to include concepts for all relevant kinds of physical objects that are used during the lifecycle of those facilities and their components.
Physical objects are objects that obey the laws of physics. Kinds of physical objects (also called kinds of physical objects) intent to define what nature of those things are, taking into account the physical aspects of the objects. For artefacts those aspects are the ones that should make them suitable to fulfil their intended role.
Every kind of physical object should be defined on the basis of one or more aspects that the members of that class have in common. A criterion for a proper kind of physical object is, that even if such an object would not be used for its intended role (e.g. when it is stored in a warehouse), then still it can be concluded from its aspects that it is a member of that class.

For example, an eccentric reducer is a kind of physical object that is a specialisation of the class "reducer", and is defined on the basis of a shape aspect.

The types of aspects that are most frequently used to distinguish kinds of physical objects are:

1. Intended role or function - e.g. heat exchanger
   This means that the object is intended to be suitable for an intended activity and that that implies consequences for the physical aspects of the object.
2. Characteristic - e.g. vertical submergible pump

3. Material of construction - e.g. stainless steel pipe
   However, it is a general rule for Gellish that physical objects specialisations according to material of construction are not included (except for special cases such as “glass fiber”). Kinds of material of construction or ‘substance’ are kinds of aspects that are defined in a separate hierarchy as specialisations of instances of the entity called atomic and sub atomic structure.
4. Shape - e.g. sphere (spherical vessel), tree
5. Composition - e.g. shell and tube heat exchanger
6. Principle - e.g. pitot tube (which implies shape)

Examples of types of aspects that are used but that indicate a role rather than a physical aspect are:

1. Location - e.g. driven end bearing
2. Condition - e.g. waste

The resulting classes usually receive their names from one or more of those aspects. However, a class usually implies the definition of more than only one aspect. A class may derive its name from only one aspect, but its definition usually includes more than one aspect which makes it distinct from its parent class. Furthermore, it may derive its name from an intended role or function, but the physical aspect that really makes the distinction is the design consequence of that.

For example: the class 'sphere' (in the meaning of ‘spherical vessel’) derives its name from its shape, although it also implies an intended activity of 'storage'.

Example 2: the class 'heat exchanger' derives its name from its intended role in an activity, although to execute that activity certain shape and properties are required such as a surface area and a thermal conductivity.

Example 3: the class 'shell and tube heat exchanger' derives its name from a combination of an intended role and a composition aspect.

The fact that a kind of physical object *derives its name* from a specific aspect has not necessarily an impact on what the class **is**. The aspect from which a class derives its name may even be language dependent. For example, a 'gear' in English is intended to gear, but it is called a 'tandwiel' in Dutch, which is a classification according to shape or composition (dented wheel).

Many kinds of physical objects are designed and/or fabricated suitable for a particular kind of role and therefore those classes are named according to that **intended** role. In order to be suitable for that role it should be recognised that this intended role implies consequences for the physical aspects of the objects, such as their shape. For example, assemblies of equipment such as process units are usually dedicated designs for a specific role. This does not imply that the class 'process unit' only defines activity or functional aspects, because it has shape consequences that makes the thing suitable to fulfil that role.

Kinds of physical objects often have names that are derived from their intended role. For example a pump derives its name from its intended role to pump (of made for the role of "pumper"). Nevertheless it is clear that a pump is a kind of physical object. The same holds for a blower, which is a type of compressor.
However, "pure" roles are independent of the type of physical object that performs the activity. If English knew the word "pumper" then it would have been a kind of role, because even a person could be in that role.
The problem in natural language is that we often use the same name for a kind of physical object and a kind of role. The term blower is a good example. It is not something that can blow (e.g. a person) , but it is a special type of machine that is made to blow.
The general rule is in those cases that we define the kind of physical object "blower" and we only define a separate kind of role "blower" if there are clear examples in industry that other types of equipment can play that role, without being named after that role.

Kinds of physical objects should not be defined on the basis of the role the object incidentally plays. Such a class typically is a **kind of role** (see below). A handy check is to think of the object being located in a warehouse, out of its normal context. Then you should still be able to conclude about its classification in a kind of physical object, but you cannot draw conclusions about a classification according to its kind of role.

The following kinds often belong to this category:

- **components**, i.e. classes that normally have a name that ends on 'component'. They are normally classes of roles that indicate that the item normally is part of a piece of equipment. For example, equipment component, control system component.
  Exceptions are classes where the term 'component' in its name can be replaced by 'item'. Such as in pneumatic component, which is a synonym of pneumatic item. The common aspect is that the item is pneumatic, not that it is a part of another item.
- **material**, i.e. classes that normally have a name that ends on 'material'. For example, connection material, insulation material, welding material. They are normally also kinds of roles, that indicate that the item normally can be used for a type of purpose.
  Exceptions are classes that do share a common aspect. Normally the term material in their name indicate that they collections. Because Gellish defines only singles, the term material in their name is superfluous. For example, plate material, which is synonym of plate.
  The class 'material' itself is also an exception, because it defines a kind of physical object whose members have a specific quantity of mass.
- **equipment,** i.e. classes that normally have a name that ends on equipment. If the term equipment is not preceded by an aspect of the equipment, but by a purpose, then the class is often also a kind of role.
  For example, connection equipment, protection equipment, fire fighting equipment.
  Examples of proper kinds of physical objects which are defined on the basis of a common aspect of its members are: rotating equipment, electrical equipment.

The number of roles that physical objects **can** play may be unbounded. For example, a screw driver can play a role as hammer. These 'possible roles' of objects (in activities) should be distinguished from the 'fit for purpose' classification which defines what a thing ***is***.
A proper kind of physical object classify objects that have *a common **intended** role which has design consequences*, i.e. what a class ***is***, not objects with common incidental possible roles or actual roles.

The following list indicates the kinds of kinds of physical objects that are included in the Gellish Dictionary:

1. Plants, Systems and generic items, including spaces
2. Rotating equipment and their components
3. Heat generation and heat transfer equipment and their components
4. Static equipment, especially Vessels and their components,
   including columns, reactors and storage items
5. Instrumentation and control systems and their components, including IT
6. Electrical systems and their components
7. Piping systems and theirs components
8. Valves and their components, both in-line and control valves
9. Protection systems and materials
10. Connection materials
11. Transport and Movement systems and their components,
    including loading facility components
12. Civil and steel structures, including buildings
13. Heating, Ventilation and Air Conditioning system components
14. Solids Handling systems and their components
15. Geographic items, such as oil fields, wells and their components, area's

16. Computers and telecommunication items
17. Waves, such as nuclear radiation, radio waves, light, sound, etc.
18. Signals, such as control signals, digital signals, etc.
19. Document files, such as drawings, data sheets, reports, messages and manuals.
20. Software.
21. Symbols and their components (annotation elements).

5.2.4 Kinds of Roles

Some classes are *only* roles. If an object does not have such a role at a certain moment, then it is not possible to classify it as a member of such a class.
For example, a 'driven end bearing' is just a normal bearing, which is classified according to its location. But that location is 'non-permanent'. It is not a permanent aspect of the bearing. When the bearing is in a warehouse it is not possible to determine this type of class on basis of the aspects of the bearing. Conversely a “thrust bearing” is a valid kind of physical object because it is specifically designed to be able to withstand axial loads as well as radial loads.

This means that those roles are played by objects with a name which differs from the name of the kind of role. In those cases there are no objects that are especially designed or fabricated for that specific purpose.
Such classes are associated by a 'can be a role of' association with another kind of object and they are also classified as 'role'.

For example,
- driven end bearing can be a role of a bearing
- driven end bearing is a specialization of role (of a bearing when located at a driven end of a shaft).

Note 1: It might be expected that 'fire fighting pump' is a role only. It is interesting to note that this 'intended use' appears in practice to have mechanical consequences and thus this class name indicates functional aspects as well as mechanical aspects and therefore is a kind of physical object.

Note 2: There are exceptional cases where a kind of physical object should be distinguished from a kind of role with identical names. For example, a front wheel is just a role of a wheel in the context of a car, but a front wheel is a specialization of wheel in the context of a tractor. Similarly: a temperature transmitter is a specialization of material, because there are specifically designed transmitters for that role. But there are also general purpose transmitters which can play the role of temperature transmitter.
This can be recorded as follows:
- kind of physical object temperature transmitter (item) is a specialization of transmitter
- kind of role temperature transmitter (role) can be a role of a transmitter

It is trivial that a material can play the role for which it was designed. Therefore it is not recorded for example that temperature transmitter (item) can be a role of a temperature transmitter (role).
Such possible roles are presupposed.

So, kinds of roles include
- kinds of functions, being roles (involvements) of objects in activities.
- kinds of roles in relations, including in correlations.

5.2.5 Identification of new classes and their hierarchy

The easiest way to determine relevant classes is to decompose a plant, system, piece of equipment or other physical object into all types (classes) of components.
For example: a pump can be decomposed into a *bearing*, a *shaft*, etc. This identifies the classes *bearing, shaft, etc.* and it identifies common composition associations between the whole (class) and its parts classes, which can be recorded as:

- bearing *can be part of a* pump.

The next step is to identify sub-types and supertypes of those components. For example, the sub-types *ball bearing, roller bearing*, etc. and the supertype *rotating equipment component.* These result in specialisation associations which can be recorded as:

- *ball bearing* is a specialization of *bearing*,
- *roller bearing* is a specialization of *bearing*,
- *bearing* is a specialization of *rotating equipment component*

So, objects can be classified and object **classes** can be classified themselves into higher level classes. (See figure 1, entity: 'specialisation of kind of object').
For example the object class 'axial centrifugal pump' is a class (type) of centrifugal pump, which on its turn is a specialization of pump, which is a specialization of equipment.
Such specialisation of classes result in **specialisation hierarchies**. Such a specialisation hierarchies are required in order to distinguish between data that is valid for all types of pumps from data that is valid only for a specific type of pumps, e.g. only for axial centrifugal pumps. Specialisation hierarchies are important a.o. for selection and comparison of performance characteristics of similar equipment and equipment components.

The Gellish language allows that any class can be specialized in an unlimited number of more specialized classes as and when required.

5.2.6 Compositions and connections between members of classes

Members of kinds of objects can be **composed of members of other kinds of** objects. These form common composition associations between existing classes. The do not define additional classes. For example, the fact that an impeller can be part of a centrifugal pump is true for the whole kind of centrifugal pumps. In the Gellish knowledge base also possible composition associations between classes are recorded. This results in composition hierarchies that specify the knowledge how members of a class are normally decomposed. For example:

- impeller can be part of a centrifugal pump
- driven equipment can be part of a rotating equipment system

Such a normal composition can be constrained by so called (simultaneous) cardinality constraints. For example, the number of impellers that is simultaneously part of a centrifugal pump normally varies from 1 to n, but an impeller can be simultaneously part of only one centrifugal pump or an impeller can be not a part of a centrifugal pump at all, for example when being in stock (so it has a range from 0 to 1). When these cardinality constraints are made explicit, the definition becomes:

1,n	impeller	can be part of	0,1	centrifugal pump
1,n	driven equipment	can be part of	0,1	rotating equipment system

Kinds of objects can also be **connected to other classes**. For example, a nozzle can be connected to a vessel. These form common connection associations between existing classes, similar to the possible connections.

5.3 Kinds of Properties, Qualities and Encoding aspects

The Gellish Smart Dictionary also contains 'kinds of properties', 'kinds of quality', 'kinds of encoded information' and allowed units of measure. The knowledge base also contains associations between those classes and the kinds of objects.
Properties are quantifiable characteristics, such as pressure and diameter.
Qualities are qualifiable characteristics, such as: open, fire proof, corrosive, etc.
Encoding aspects are the information bearing aspect of physical text, sound that is stored as a bit pattern in a computer, which is encoded according to some coding rules, such as a natural language, a coding standard such as ASCII, Unicode or HTML as used by the Internet or an application specific encoding, such as .DOC as used by MS-Word.

A quantified property can either be a specific possessed property or a common property that can be used for comparison of possessed properties. These should be distinguished from the concept for a type of property.
For example, a specific possessed property classified as diameter, is equal to 3000 mm, which is a common value of a concept of property called distance.
Gellish also include ***allowed*** **common properties (with values)** for kinds of properties, common qualities or standard kinds of encoded information, such as the characters of the alphabet. These allowed values are sometimes called domain values.

The library contains associations between kinds of properties, qualities and encoded information and the kinds of objects for which they are recognised to exist or for which they are definitional. For example, the library contains an association that says 'diameter' is a common kind of property for 'vessel'. These are defined as associations between a kind of property and a kind of physical object.

So, kinds of properties and their associations to kinds of objects include:
- mechanical characteristics, such as diameter, hardness,
- shapes, such as cylindrical shape, etc.
- fluid properties, such as density, viscosity, etc.,
- duty characteristics, such as capacity,
- design limits, such as design pressure,
- units of measure,
- states, such as phases (liquid, vapour, etc.), open, closed, toxic, corroded, etc.

Kinds of encoding aspects and their associations to kinds of objects include:
- languages, standard data formats,
- procedures, methods and algorithms, etc.

5.4 Kinds of Aspects of Physical objects

Kinds of physical objects are always defined on the basis of one or more aspects that their members have in common. The foundation classes in Gellish that form the top of a much bigger class hierarchy were initially defined only by being a subtype of a more general class and a further textual definition that describes its distinguishing aspects. More and more these aspects will be modelled explicitly in Gellish.
For example, a centrifugal pump is a specialisation of a pump on basis of the fact that is applies the centrifugal principle. So 'centrifugal' is an aspect that qualifies centrifugal pumps.

Specialised kinds of physical objects can often be defined completely by the fact that they possess one or more aspects that are the basis for inclusion or exclusion of members in the class. This may mean that they are defined sufficiently by the definition of those aspects, without the need for an additional textual definition. So, specialised classes should be defined by the explicit definition of their possessed aspects. They might not even need a name, but only an identifier that is unique in a specific context. This is especially the case when the name of the class can be derived from a concatenation of its aspects.
For example, a special type of flange (= a kind of flange) defined by a specific buying description might be identified only by a company specific code, such as a KKS code. It does not need a name and it can be selected from a search on one or more of its aspects, including its parent classes.

To select a 'standardised object' an engineer typically starts with a few **properties** of the specific object that he is designing, based on the requirements of the process or activity that should be performed. Then a 'standardised object' (= a specialised kind of physical object) is selected which has **properties** that fulfil the required properties sufficiently. Then the specific object can be classified to be of one of that standard class.
However, it should be realised that classification of an object may result in a situation that one or more of the properties of the specific object don't get a value assigned to it, but classification means that its properties are within the constraints on the property of the class.

The foundation classes in Gellish do not include classes that are defined by values of their properties. For example, a 6 inch valve will not be included as a foundation class. There are a few exceptions on this rule: whenever classes are generally recognised in (engineering) practice, then they should be included in Gellish, even though they might be defined completely by their explicit aspects and even if their textual definition might be left away.
On the other hand, the definition of such classes should be limited to classes that are already in use in normal engineering practice, because theoretically it is possible to think of an unlimited number of those classes.

For example: 'single stage compressor' is a relevant subtype of compressor, because it is generally recognised in industry, even though a specific physical object can be described also without that class, such as by classification as 'compressor' with a number of stages = 1.

The aspects of physical objects also include quantification of their magnitude and qualification of their shape. Therefore kinds of mathematical and geometric items are included, such as:

- geometric models and their components,
- mathematical models.

5.5 Kinds of Substances - Process materials & Materials of construction

Physical objects are made of types of matter. The atomic and subatomic structure of matter is often referred to as its 'type of material' or its substance. Kinds of these define the types of atoms and molecules and the way in which they relate in a mixture.
For example, an impeller is made of stainless steel. Then the class 'stainless steel' only refers to such a structural aspect. This fact is to be distinguished from the statement that an impeller is made of a specific batch of material. (The structural aspect of) the batch can also be classified as stainless steel.
Thus 'stainless steel' is an examples of a kind of substance (usually in the role of **material of construction** for a physical object). This means that objects that are made of some kind of substance will have an association with that kind of substance.
For example: physical objects *can be made of* stainless steel

Therefor, 'material of construction' is not defined as a characteristic of an object, but as a classification association between an aspect of an object and a 'kind of substance'.
Process material like construction material, is also a role of a material. So, oil, water, butane, etc. are other examples of instances of 'kind of substance'. They have their own hierarchy:

crude oil	is a specialization of	oil
stainless steel	is a specialization of	steel

Process materials, can be classified also according to their state or phase. For example: material can be in a fluid state, which state can be specialised further into liquid, vapour, solid, plasma, etc. or its state can be crystalline, polymorph, etc.

So, kinds of substance include:
- solid, such as steel, plastics, concrete etc. mainly used as material of construction
- fluids, such as oil, chemicals, water, air, etc. mainly used as process material
- physical elements and their constituents, such as H, He, etc., electrons, nuclides, etc.

5.6 Kinds of Signals, Waves and Energy

Signals and waves are subtypes of physical object. Energy is defined as a **property.** They have a classification hierarchy which starts as follows:

signal	is a specialization of	physical object
electrical signal	is a specialization of	signal
radio signal	is a specialization of	signal
wave	is a specialization of	physical object
light	is a specialization of	wave
energy	is a specialization of	property
etc.		

Signals are modulated streams of physical objects, from which the modulation properties are representing the carried information.

Waves are physical objects, which can move in matter or can be present in the form of radiation. Waves represent (or 'contain') **energy**, such as heat, which can be contained by a material (and is seen as a property of the material). Energy can be extracted from a material and transported through another material.
For example, **heat** extracted from a fluid can be transported through the surface of a heat exchanger and transferred into another fluid.
Electro-magnetic waves are physical objects that can move in vacuum, but can also move through another physical object, for example in the form of **electricity** which moves through a conductor. Electrical energy can also be possessed by a physical object, such as electrical charge which is stored in an accumulator (such energy is also seen as a property of that physical object). It can also be extracted from the physical object and transported through a conductor such as in a wire.
For example, electricity and light waves are used to carry a physical representation of **encoded information** (e.g. a message) for control purposes, but they can also be used to transport energy.

5.7 Kinds of Activities, Processes and Functions

Kinds of activities and processes include:
- human activities and business processes,
- industrial processes and unit operations, such as distillation, heat transfer, etc.,
- natural processes, such as corrosion,
- control processes (functions) and electrical processes, such as conversion of signals,
- events

5.7.1 Kinds of Occurrences - Activities and Events

Occurrences are things that take time and may change over time. Activities are things that happen or occur. So an activity is a subtype of occurrence. In language terms, kinds of occurrences are expressed by the verbs in the language. In Gellish the active form , the passive form and the noun form are seen as synonyms. For example, to act, acting and action are all three names of one kind of activity.
Gellish contains kinds of activities, irrespective of the question whether they are expected to be executed by people (human activities) or whether they are physical and/or (bio)chemical processes. Some of these kinds of activities are formalised versions of the kinds of activities that can be found in the 'Process Plant Engineering Activity Model' (AP221 annex E).
So the kinds of activities cover the whole life cycle of process plants, from research to demolition, including the activities of the organisation and the physical phenomena that take place in the plant during that time.

Kinds of activities in Gellish do not include the subject of an activity, nor the performer or enabler. For example, 'design', 'maintain' and 'repair' are kinds of activities, but design a pump and repair a pump are not recorded as subclasses. If this would be done it would lead to a combinatorial explosion of the number of classes because of all possible combinations.

A specific activity can be described as follows:

activity	#1	is classified as	repair	
physical object	#2	is classified as	pump	
physical object	#2	has a role as	subject	in activity #1

5.7.2 Kinds of Processes, Process control and Functions

Anything happening over time is called an activity in AP221. Therefore **processes** such as *distillation* or *fabrication* are defined as subtypes of activity. Also **process control** processes, such as *control, measure or actuate* are subtypes of activity. These processes are therefore included in the class library in the same way as kinds of activities.

Examples of physical processes included as kinds of activities are: distillation, reaction, catalytic cracking, heat transfer, transport, etc.
Also 'passive' activities and events are included as kinds of activities. For example, 'contain', 'happen', 'trip', 'crack', etc.

The term **function** is used in practice in several ways. Sometimes a function is an algorithm or mathematical equation which is an association between the parameters or inputs and outputs. For example the VDI/VDE 3696 ‘control functions’. In other cases a function is a role, such as the role (function) of a person. Finally, the term ‘function’ can also be used to indicate an activity or process (to function). For example a functional decomposition usually means a decomposition of the activities that should be performed. Therefore in Gellish kinds of functions should be recorded as either subtypes of activity, role, or relation (mathematical function).

5.8 Templates and Product models (Forms and Data Sheets)

Information is in general expressed as a collection of aspects possessed by objects. Such information is in general required for, or resulting from an activity. It can be recorded in a file or in a document, such as a data sheet, a bill of material, material balance, etc. Those collections of aspects possessed by objects contain a selection of relevant **facts** collected from a specific perspective, called a context. It is always only a sub-set of all the facts that exist about an object. The combination of a complete set of facts about an object is called a **product model**. Such a required or resulting set of facts can also be defined in the form of a **query.**
It is also possible to define a ‘fill in the blanks’ **template** for the creation of new product models for either an individual thing or a kind of individual thing. A template can be compared with an '**empty form**' with pre-defined text that define the meaning of the empty fields.

For example, in AP221 terms a centrifugal pump data/requisition sheet defines a set of atomic facts about a pump unit. Each atomic fact consists of an association between objects, such as pump components, properties, numeric values, etc., including classification associations with kinds of properties and kinds of pump components, fluids, activities, etc.
On the other hand such a template implicitly defines, so to say, the **query** (e.g. on a database) that selects the data required to order a pump and thus to print such a data sheet.
Application systems need the definition of those templates in order to operate properly.

Gellish contains the definition of common properties for kinds of physical objects for those templates. For example, Gellish contains facts such as:

a container can have property internal volume,

which association is inherited by the subtypes of that physical object, so that a 'pressure vessel' and a 'tank' also can have an internal volume.

The definition of templates can make use of those facts, but the templates themselves are currently not (yet) part of Gellish, although data requirements for some specific 'exchange scenarios' may be incorporated.
An **exchange scenario** defines a set of data that can or should be included in a data exchange, which is part of a standardised kind of activity.

For example, 'order a pump' can be done in a standard way, whereas that standard way can be defined by the definition of a kind of activity. Such an activity requires a specific set of data that is sufficient to specify a required pump. The content of such a specification may be more or less detailed, depending on the company that defines the activity. Therefore, if this is standardised, the standard can and will define only a minimum set. The maximum set includes all characteristics that can be associated with a pump and its components.

5.9 Synonyms, Homonyms and Unique identifiers

Names of classes are associated with a class via an identification (naming) association. This allows for multiple names for the same class, i.e. synonyms. It also means that a class itself is in principle sufficiently identified by a unique identifier and can even be without a name. Therefore, it is intended that all classes in Gellish will have a global unique identifier. This unique id consists of a part that is identical for all classes (ISO10303-221.CLASS), followed by a unique number (00001, etc.).
These unique identifiers will provide among others the capability to use homonyms (the same name for different classes), whereas homonyms are distinguished by relating the '**context** for reference', which is the context in which a homonym is primarily applicable to the relation between the object and its name. Homonyms are distinguished in Gellish by the context usually referred to as the "subject area" in which the term is used.

The prime names of classes in Gellish are the 'preferred names' in the English language. Synonyms in Gellish are 'non-preferred alternatives'. The names in Gellish are intended to be as much as possible context independent.

Gellish also provides names in various languages and can contain coding according to various 'naming systems' or 'coding conventions'.
Examples of standard kinds of *encoded information* are: French, German, Dutch, Kanji, etc.
Gellish is primarily documented in English, but users are invited to send translations of the current class names to the library manager, they will be considered for future inclusion in Gellish.

5.10 Other Standard data

The Gellish Dictionary also includes:

- Scales, including Units of Measure and Currencies.
- Reference values and ranges for properties and qualities (domain values).

- Standardised types of components.
- Specific Geographic Items, such as Countries, Area's and Oceans.
- Specific individual organisations.

6. Individual Things

Apart from kinds of things, there are also '**individual things'.** Those individual things can be used for example as reference objects. They can be distinguished according to their role in **specific objects** and **typical objects**.
Examples of specific objects in Gellish are countries and organisations.
Examples of typical objects that are included in Gellish are standard symbols and typical designs, etc., intended for derivation of specific objects.

Typical objects usually appear in object libraries. Gellish can contain definitions of these typical objects and their associations, such as decomposition, classification, etc.

7. How to define a kind of thing

A kind of thing (a concept) should only be included in the Gellish dictionary when it defines that a collection of things have common characteristics and/or have a common behaviour.

Every kind of thing shall be defined by the following information (in line with ISO 1087):

1. A name
   A name shall comply with the rules for names of kinds of things, as defined in the next paragraph.
2. Optionally one or more synonyms.
   Note that synonyms are really interchangeable. However, for the clarity of the visualisation of the hierarchy, it is recommended not to use the synonyms in a visualisation of the hierarchy, but to use only the 'preferred term'. (Note that in Gellish the term 'preferred term' has no formal status, because preferences are context dependent).
3. A specialisation relation with an existing supertype kind of thing:
   … is a specialization of …

4. Optionally one or more relations with other kinds of things, such as
   … can have as aspect a …
   … can be a part of a …
   … can be a role of a …
   … can be connected to a …
   especially what its discriminating aspects are. For example:
   … is by definition qualified as …
   … is by definition a part of a …

   For example:
   The subtypes of a pump are distinguished by the discriminating aspect ‘operating principle’.
   The operating principle of the subtype ‘centrifugal pump’ is by definition qualified as ‘centrifugal’.
   This textual definition can be modelled as follows:

- pump	can have as aspect a	operating principle
- centrifugal pump	is a specialization of	pump
- centrifugal	is a qualification of	operating principle
- centrifugal pump	is be definition qualified as	centrifugal
- pump	can have as aspect a	orientation
- horizontal pump	is a specialization of	pump
- horizontal	is a qualification of	orientation
- horizontal pump	is be definition qualified as	horizontal

   Note, the first and fifth fact of this example can be derived from the other facts. They are logically implied. Software should be able to draw such conclusions.

   A textual definition can be automatically generated from such facts. For this example as follows:

   A centrifugal pump is defined as: a pump of which its operating principle is centrifugal.
   and a horizontal pump is defined as: a pump of which its orientation is horizontal.

   In case of multi-parent classes it is possible that a class inherits two discriminating aspects while it does not have an additional discriminating aspect.
   For example:

- a horizontal centrifugal pump	is a specialization of	horizontal pump
- a horizontal centrifugal pump	is a specialization of	centrifugal pump

   In fact this class does not need a textual nor a modelled definition. However, for clarity two definitions might be added as follows:

   - a horizontal centrifugal pump is defined as: a horizontal pump of which its operating principle is centrifugal
   - a horizontal centrifugal pump is defined as: a centrifugal pump of which its orientation is horizontal

5. A textual definition.
   A free textual definition should build on the definition of its direct supertype (see item 3 above), and preferably start with ‘that consists of’ or with ‘intended to’ (which means: ‘that is intended to be suitable to’) or ‘of which its’ (followed by the discriminating aspect (see item 4 above). This textual definition shall describe in which aspects the defined subtype is a specialisation of its direct supertype and in what respect it is distinguished it from its ‘sister’ kinds of things.
   This means that the additional textual definition should preferably be read as follows:
   … is a specialization of … that consists of *[etc.]* or
   … is a specialization of … intended to *[etc.]*.
   When the definition is given this way the ‘full sentence definition’ will be generated automatically by the concatenation of the association with the direct supertype and the additional textual definition.

6. Further definition associations (optional)
   Optionally one or more relations with a specific property value, algorithm or other kind of thing. These additional relations provide an explicit definition of the thing by qualifying an aspect. Such an explicit definition should be specified as follows:
   … is by definition …(property value-1)
   For example: a plastic vessel is by definition made of plastic.

Private extensions to the Gellish language are recommended for proprietary concepts. However, it is recommended to feedback any proposed extensions of kinds of things that belong to the public domain knowledge.
This implies that company specific kinds of things can be defined. The definition of such private extensions can also be exchanged in Gellish and can subsequently be used in other Gellish expressions. However, those private extensions should also be defined according to the rules described above. Otherwise they will not form a consistent set and do not inherit any information from higher level concepts.

This capability is especially of value for supplier specific catalogues which may contain inventions and proprietary company specific concepts. However, it is strongly recommended to minimise the definition of private concepts in order to minimise the diversity of Gellish dialects. If private extensions are nevertheless defined, then they should be made subtypes of existing standard kinds of things in Gellish.

For example, the class 'centrifugal pump' is part of the STEP AP221 Class Library.
The catalogue item 'Sundyne pump model 311' should be defined as follows:
- Sundyne pump model 311 is classified as Sundyne pump
- Sundyne pump is a specialization of 'centrifugal pump'.

So, when a company has exchanged the company specific class 'Sundyne pump' and its classification association: 'Sundyne pump' is a specialization of 'centrifugal pump' together with its specific properties, then the receiving party can classify a specific pump as a 'Sundyne pump' even without referring to a specific model.

7.1 Rules for names of kinds of things

A class name should satisfy the following rules:

1. **Artificial class names** shall be avoided, but natural language and normal engineering class names should be used instead.
   For example, 'piping_segment_cmpnt' is not used in normal engineering practice and thus it should not be used.

2. **Class names** shall be **in single**, not plural.
   For example, *vessel* and not *vessel**s*** or *fitting* and not *fitting**s***.

3. **No abbreviations** nor **underscores** shall be used and names and definitions shall be in **mixed case**, which means generally **lowercase** with exceptions for generally accepted abbreviations. For example: *centrifugal pump* and not *centrif._pump.* Widely used abbreviations and/or uppercase may be included, especially in *synonyms* (see rule 5 below). For example: PC. Note, this rule holds for English, whereas in other languages different rules may apply. For example, in German all nouns start with an uppercase character.

4. **Multi word class names** shall be in the **normal word sequence** for the applied language. For example, 'centrifugal pump' in English and not 'pump centrifugal'.

5. **Synonyms** are allowed and may be accompanied by a context in which the synonym is valid. A synonym is an 'alternative' which should be associated with the 'reference class name' (via an 'is a synonym of' association). The reference class name should be used in all other associations.
   For example, *ship* might be seen as a synonym of *sea going vessel*.
   Synonyms within a language context are e.g. synonyms in French, or German.

6. **Concatenated classes** are allowed, but shall be classified according to their constitutive components.
   For example, the class *single stage centrifugal compressor* is allowed, but should be classified as *single stage compressor* and as *centrifugal compressor*.

7. All classes shall be accompanied with **at least one** classification or (de)composition **association**.

8. All classes inherit the aspects of their parents in their hierarchy. For example all equipment and equipment components are a **subtype of 'item'** and thus they inherit the associations which an item has.
   An item is defined as any component which is a solid material and thus has a particular shape. All items can have the following properties:
   mass, volume, number of items, cost, economic value, marginal value.
   So all those subtypes inherit these properties. This needs not be specified for every class again. For example, a *bearing* is an item and therefore it inherits the fact that it can have a mass and a volume.
   All kinds of physical objects also inherit all possible associations that may be made to physical objects as defined in the data model. For example, all classes can have a name (tag), a description, a material of construction, a shape (geometry), etc.

9. All classes can be used to classify single items or **collections**. Therefore, no separate classes are required for collections.
   For example: the class centrifugal pump can be used to classify P-101 or can be used to classify a collection of pumps in a warehouse. The object type should then be defined as "plural physical object". The size of the collection can be indicated via the count 'number of items'. It is possible to classify a collection as centrifugal pump. Alternatively the elements in the collection can be classified as a centrifugal pump.

10. Class definitions shall build on the definition of their direct supertype and only add the aspect(s) in which it distinguishes itself from that supertype and its neighbours.
    For example:
    ball bearing is a specialization of *bearing* **which contains balls**.
    As a result a definition may seem to be simplistic. However that is not the case. You should realise that the name of the direct supertype in fact stands for the definition of that direct supertype which has a direct supertype again with a definition, etc. etc. until the top of the classification hierarchy is reached. This chain gives a very rich total definition.

Furthermore, the additional textual definition (the bold part) shall preferably start with an element that is derived from the aspect which is used for classification.
For example,
'intended for' for classification according to intended activity,
'which consists of' for classification according to composition, etc.

11. **When to use a dash or a space or no-space?**
For example, should there be a dash (‘-‘) in methylethylketon as follows methyl-ethyl-keton and should we write, slip-on flange, electro-magnetic and soft-start spool?
In Gellish the ‘normal’ natural language conventions are used. I believe that the normal convention for the example is to concatenate the entire word, as in methylethylketon.
In case of doubt it is possible to define a synonym with another spelling for the same concept.

12. **When to use a dash or a slash?** For example in AC-DC or AC/DC.
In Gellish the rule is to avoid abbreviations and codes. The ‘normal’ natural language conventions in this case is that the dash (-) means ‘to’ (AC to DC) and the slash (/) means ‘or’ (AC or DC). Therefore we recommend to have the preferred term spelled out. So use ‘to’ and ‘or’ in full text, but to add strings with a dash or a slash as synonyms.

13. **When to use an s and a z?**
Gellish follows the rules of natural language. So for Gellish UK English this means in most cases an s is used. However, it is valuable to add a synonym with a ‘z’, because in British English, there are two conventions, one of which uses a z for words like "organize", "piroritize" etc. and the other of which uses an s for these same words.
In American (US English) many words take a z, which in British English take an s.

7.2 Rules for relation between kinds of things

There are a number of different **types of relations** that can exist between kinds of things.

7.2.1 Specialization relations

A specialization relation defines a 'subtype - supertype' relation between two kinds of things. The kind of thing that is the subtype is a specialization of the supertype.
For example: a pump *is a specialization of* equipment item.
Note that this relation can also be interpreted in the inverse direction. Then it defines that the supertype is a generalization of the subtype.
For example: equipment item **has as subtype** pump, or in other words: equipment item is a generalization of pump.

Specialization relations can specialize a kind of thing according to its inherent (persistent) aspects or according to the roles that its members can play. Therefore we distinguish between kinds of phenomena (physical objects and aspects) and kinds of roles.

Rules:
- multiple specialization and multiple generalization is allowed. This means that a kind of thing may have several subtypes and also several supertypes.

1. **Specialisation relations** are expressed by the phrase '**is a specialization of**'.
This describes the relation between a kind of thing that has a role as subtype with another kind of thing that has a role as supertype in the relation.
The example above: 'a pump *is a specialization of* equipment item' is an example of such a specialisation relation.

2. **Possible role player relation**, expressed by the phrase '**can be a role of a**'.
   This relation relates a kind of role and a kind of thing of which members can play such a role. For example, fire-fighting pump *can be a role of* a pump.
   The inverse phrase for this relation expresses that a member of the kind of thing can play a role that is classified by the kind of role. For example: a pump *'can play a role of*' fire fighting pump.

7.2.2 Composition relations

Because of the difference between an assembly and a collection (see above) we also make a distinction between an *assembly relation* between a part and a whole assembly and a *collection relation* between an element and a whole collection.

1. **Assembly relation**, expressed by the phrase '**can be a part of a**'.
   Many kinds of things can have **compositions**. This means that the knowledge can be expressed how (members of) kinds of things can be composed of (members of) other kinds of things.
   The Gellish knowledge base contains a large number of such possible assemblies, expressed in the form of (conceptual) assembly relations between kinds of things.

   An assembly relation describes a 'part - whole' relation, defining that members of one kind of thing can be a 'part of' members of another kind of thing, being the whole (the composite or assembly).
   For example, each of the following three kinds of things, a nozzle, a head and a shell *can be a part of a* vessel.
   Note that this relation type also defines its inverse, being the fact that a whole assembly can be decomposed into parts of these kinds. For example, a vessel '**can be composed of**' a.o. nozzles, heads and shells.
   Rules:
   - an item may be part of more than one assembly at the same time.
2. **Collection relation**, expressed by the term '**can be element of**'.
   Apart from single things, we also recognise **collections of things,** which may be collections of identical things or collections of different things. For example, a collection of pumps in a warehouse.
   Collections are plural things. They are things in their own right (just as assemblies). This means that catalogues, libraries, bills of material, etc. are objects themselves, that can have collection relations with their constitutive components.
   Gellish enable to express that each element of a collection is classified by a particular kind of thing, whereas the whole collection is classified only on a high abstraction level. For example, 'my stamp collection' is classified as a collection AND for my stamp collection holds the relation that 'each element is classified as a' stamp. This has as advantage that the dictionary does not contain separate names for all kinds of collections, but that a specific collection is classified as collection AND that each element is classified. To support this the dictionary contains classes like 'collection', 'catalogue', 'bill of material', etc.
   For example, there is no need for a kind of thing called 'pumps' next to pump, because a pump collection should be classified as 'collection' (or one of its subtypes) and should have a separate relation that says 'each element is classified as a' pump.

   A conceptual collection relation expresses a relation between an element of a kind and a collection of a kind. This means that members of one kind of thing *'can be an element of*' members of another kind of thing, being a collection.

For example:
- a schematic symbol ‘can be an element of a' symbol library,
- a piping component ‘can be an element of a’ piping class (library),
- a spare part ‘can be an element of a' stock in a warehouse and
- a material ‘can be an element of a' catalogue.

The inverse phrase expresses the possible decomposition of a collection of such a kind into individual elements of the indicated kind.
For example: a piping class *'can include a'* valve and a fitting.

7.2.3 Geometric Representation of Objects and Characteristics

Gellish distinguishes between a kind of 'annotation element' and a kind of thing that it can represent. For example, a valve symbol can represent a valve.

Furthermore, Gellish distinguishes between the symbol which is an assembly of lines, circles, etc. and their geometric aspect (the shape of the symbol and its parts),
In the Gellish dictionary there are therefore kinds of symbols (and possibly typical individual symbols) and relations between the kinds of symbols and the kind of things that they can represent.

The shape of kinds of symbols in a symbol library, is often company specific, although various standards institutes have published catalogues with 'standard symbols'.
Gellish can be used to define such catalogues with geometry (shape), but the dictionary is not intended to contain the precise dimensions of such standard symbols.
For example, the concept 'centrifugal pump symbol' and 'globe valve symbol' as well as their associations with the kinds of things that they can represent will be part of the dictionary, but the exact geometry of those symbols is not defined in the dictionary.

7.3 Multiple languages

Any fact may be expressed in different languages. However the fact id’s of identical facts shall be identical.
It is also allowed to add translation relations that express how the name of a concept in one language is translated to another language.
For example:

Language	**UID of left hand object**	**Name of left hand object**	**UID of fact**	**UID of relation type**	**Name of relation type**	**UID of right hand object**	**Name of right hand object**
English	1	network	4	1146	is a specialization of	3	system
Nederlands	1	netwerk	4	1146	is een specialisatie van	3	systeem
English	1	network	5	4691	is a translation of	1	netwerk

It is also allowed to define the reverse translation as well, as follows:

Nederlands	1	netwerk	6	4691	is een vertaling van	1	network

Two facts are identical when the following UID’s are identical:
- the left hand object id,
- the right hand object id,
- the relation type id,
- the id of the validity of fact.

Two identical facts expressed in different languages implicitly define that the names of the related concepts are each other’s translations (synonyms). For example, fact 4 in the above table implicitly defines that network (English) is a translation of network (Nederlands).

If an explicit translation relation is present for things that also appear in identical facts (expressed in different languages) then software should verify whether the names of the concepts in the translation relation is consistent with the names used in the corresponding identical facts.

It is recommended to define translations via a translation of the specialization relation, because it shows the taxonomy (subtype – supertype hierarchy) also in the other language.

8. Gellish language definition management

8.1 Rules for acceptance of proposals

The Gellish language is regularly improved and extended. The dictionary and knowledge base is developed and its quality is ensured by groups of domain experts, organised in 'peer groups' per discipline area. Any person or organization can contribute by issuing proposals according to the procedure below.
Because the Gellish language definition is maintained in electronic form it is recommended always to work from the latest electronic version as published by the Gellish project on SourceForge. It is also recommended to minimize private local modifications, but to propose modifications and enhancements for the centrally maintained version. This will allow compatibility with new issues of the library.

The following rules for management of the Gellish language definition apply:

1. Gellish is **Open Source** and in the **public domain**. So are anybody's contributions.

2. Use of Gellish in commercial systems is at everybody's own **responsibility and risk**. It should be noticed that the dictionary is subject to regular updates.

3. **Proposals** for additions, deletions or other enhancements shall be provided as follows:

   - Use the spreadsheet as distributed by the Gellish Manager.

   - Proposals can be issued via a peer group member or directly to the Library Manager. A proposal shall satisfy the quality criteria as described in this guide before being considered by the peer groups.

   - All changes shall be introduced via a new row in the spreadsheet, either by insertion or by copying of an existing row,

   - Fill in the fields in the new row as it is or should be in your opinion or preferably as the common opinion of the peer group members. Filling in the following fields is obligatory:
     - Fact id, Unique id = *blank* for new line or *duplicate of id* of the line that is proposed to be modified.
     - Status = 'proposed', or 'propdel', which means: ‘proposed to be deleted’.
     - Originator of proposal = *your name*.
     - Date of start of life = *current date* for a new line.
     - Date of latest change = *current date* for new or for modification of existing lines.
     - Source / reference = *source of proposal (in case of an authoritative reference)*.
     - Remarks = option to argue why the proposal is made.
     - Context (optional) = The name of a context in which the fact is applicable, 'required' or 'relevant'.

   - **Select only your proposals** from the spreadsheet, via sorting on your name and date of latest change; copy your proposals into a separate spreadsheet and return that spreadsheet to the Gellish Manager or ask a peer group member to support and forward your proposal.

4. The Gellish expert peer groups will **review** the proposals in their subject area and provide technical advise on their acceptance or rejection.

5. The **Gellish Forum peer groups** have the authority to accept proposals or to reject proposals for inclusion in the library. Final approval is given by the Gellish Language Manager under supervision of the Gellish Forum.
   The authority on inclusion of classes in the official ISO 10303 or ISO 15926 standards remains with ISO TC184/SC4.

6. The **Gellish Manager** or his delegates are the only authorised persons to change the status of the proposals into **accepted, replaced, deleted, deleted class, deleted association, inherited, ignore or issue**.
   The quality and completeness of the library is the result of volunteered co-operation of various parties. However no one can be formally held accountable for the quality of the library content. No compliance to the library can be claimed, until formal publication of the standard and until official tests are performed as part of the ISO STEP test procedures.

7. The Gellish Manager will issue regular **updates** of the Gellish language definition. It will be ensured that the future ISO 15926-4 standard will be a subset of the engineering concepts of Gellish, whereas Gellish will also include concepts from other ISO, IEC, ASME, and many other standards.

8.2 Return proposals to the Gellish Manager.

With reference to the rules of class library management as described in Section 7, a spreadsheet which contains the proposals in the prescribed format have to be returned to the class library manager. This can be arranged by sending either a file via Internet or by sending a diskette by mail.

The e-mail address for sending attached files to the Gellish Manager is

andries.vanrenssen@shell.com

8.3 Version management

The Gellish manager will ensure that regularly back-up copies are made and that consistent releases of Gellish will be published regularly on the world wide web.

A new version will consist of the following content:

- One or more files, each with a particular release date.
- Each file will consist of a first sheet with a Gellish Database table that has in its first line a date of the latest change within the file.
- Each line in the Gellish Database table has a date of latest change. From that date it can be retrieved which lines were changed since a previous version.
- There may be a sheet called ‘Deleted’ with a Gellish Database table that contains lines that are removed from the first Gellish Database table. Those lines may have the status ‘replaced’, together with a UID of a successor of fact. This indicates that the replaced line is succeeded by the line indicated by the UID of the successor.

9. How to obtain and view Gellish

9.1 Gellish downloading from Internet

A version of the Gellish Dictionary/Taxonomy, can be down loaded from the Gellish website.

The library is available as a Gellish Database (either as a collection of Excel spreadsheets or in other form).
All files may be downloaded from the SourceForge website at no cost under open source license conditions.

To obtain a copy of the whole or a part of Gellish and its documentation:

Visit the www.gellish.net website.

- select the Download area
- download one or more of the files from the Gellish Dictionary/Taxonomy package.

Note: There are various other files available on that website, for example with Documentation or Examples, such as:

- Definition of Gellish Databases and Data Exchange Messages
- Gellish API specification

9.2 Gellish enabled Browsers & Databases

Gellish information can be read and browsed via Gellish enabled Browsers or Database systems. An example is the Gellish Search Engine application, which is able to read Gellish Databases in the form of XLS files (MS Excel files) or a Gellish Database in its proprietary CLB format.

A summary of the capabilities of a typical Gellish enabled application is given below:

1. Read and view Gellish Databases.
   This includes the provision of a search OPTION and a navigation option to find things and their related things.
2. Generate a Gellish Database for selected information.
3. Search for documents and data via a Facility Information Model and launch document viewers.

9.3 Tree hierarchy from a Gellish Database

For Gellish Dictionary/Taxonomy Developers there is a Visual Basic module for Excel workbooks available. This module can generate a tree hierarchy view in a second spreadsheet in the workbook. This module can be initiated as follows:

1. Load the module.
2. Load the target spreadsheet from which a partial hierarchy is required.
3. Ensure that the target spreadsheet contains a sheet called ‘partial hierarchy’.
4. Press “control+I” to initiate the module.
5. The module will then present the name of the source sheet. Confirm that name or cancel.
6. Enter the class name that is to be used for the top of a part of the tree hierarchy to be generated.

The result is presented in ‘partial hierarchy’ as an indented list.
- Classes presented in black have the status ‘accepted’,

- Classes presented in red have the status ‘proposed’ and
- Classes that are stroked through have the status ‘deleted’.

9.4 Gellish application services

The Gellish@Work organisation can provide support or certification of the application or extension of the Gellish language. Further information is available via www.gellish.net. Various companies can also provide services on the application of the Gellish language.

10. Appendix A. The Gellish Dictionary/Taxonomy content

10.1 Gellish Data Tables

The Gellish language definition and knowledge base is available as a collection of files with **Gellish Data Tables**.
The Excel spreadsheet table version of those files contains also worksheets with example partial hierarchies, generated from the source table.

The Gellish Data Table, Database and Messages are documented in the document "Definition of Gellish Databases and Data Exchange Messages" available from website https://www.gellish.net. The relation between the Gellish Database and some data models is also explained in that document. This includes an outline of the mapping to the ISO 10303-221 (AP221) and the ISO 15926-2 data model.

10.2 Subject areas

At the date of publishing of this version of this Gellish Language Development Manual, the Dictionary/Taxonomy is available as a set of Gellish Databases in Excel spreadsheets. On request it can be delivered as XML files. Each spreadsheet file contains one or more of the subject area sections as listed below. The list is an indication only as it is subject to change, due to the regular updates of the Gellish knowledge base content.

10.2.1 TOP of the specialization hierarchy

TOPini Kinds of Relations, Roles in relations and Generic concepts.

10.2.2 Associations with Kinds of Physical Objects (Solid items)

Class definitions, specialisations, compositions and roles of solid items.

civil	kinds of	civil, Architectural (buidings) and Structural items
connections	kinds of	connections & Connection materials
electrical	kinds of	electrical items
e-machines	kinds of	electric machines and their components
instr. & control	kinds of	instrumentation, control and IT items
transportation	kinds of	transport and (un)loading equipment items
geographic	kinds of	geographical items
piping	kinds of	piping items
facilities	kinds of	plants, process units and systems
protection	kinds of	protection material
rotating equipment	kinds of	rotating equipment items and their components
solids handling	kinds of	solids handling equipment items
valves	kinds of	valves and valve components
static equipment	kinds of	static equipment items - vessels, reactors, tanks, etc.

10.2.3 Associations with Kinds of Substances

Class definitions, specialisations, compositions and roles of substances.

steel	kinds of Steel
solids	kinds of Non-Steel Solids
fluids	kinds of Fluids - Process Material, Waves & Signals
elements	kinds of Elementary particles and atoms - Elements and their constituents

10.2.4 Associations of Kinds of Occurrences (Activities, Events & Processes)

Class definitions, specialisations, compositions of occurrences.

activities	kinds of Occurrences	- Human Activities, Events and Processes
functions	kinds of Functions	- Especially Process Control Functions

10.2.5 Associations of Kinds of Roles & Roles of Objects in Activities

Class definitions and specialisations of roles and recognised roles of objects in the context of occurrences.

roles	kinds of Roles (of objects in relations and in activities)

10.2.6 Associations of Kinds of Information & documents

Class definitions and specialisations and roles of information carriers and presented information as well as the presented objects.

documents	kinds of Documents - Reports, Drawings & Other Media
information	kinds of Information (irrespective of the way of expression)
annotation	kinds of Annotation elements - Lines, Area's & Symbols and the relation with the intended presented Information or presented Physical Objects.

10.2.7 Associations of Kinds of Aspects, including Properties and Encoding aspects

Class definitions and specialisations of properties, qualities, identifiers and information.

encoding	kinds of Encoding, incl. names, numbers and definitions (text)
properties	kinds of Properties and Qualities

10.2.8 Units of Measure and scales

Definitions of scales and units of measure (UoM's) for classes of properties and their conversion.

UoM	Unit of Measure - Definitions and relations to kinds of Property

10.2.9 Association of Kinds of Organisms

Class definitions, specialisations, compositions and roles of lifeforms.

lifeform	kinds of Organisations, Persons and other Lifeforms (organisms)

10.3 Examples

10.3.1 Example of an Indented specialization hierarchy for Rotating Equipment

rotating equipment
- blower
  - air blower
  - axial blower
  - centrifugal blower
  - straight-lobe compressor
- centrifuge
  - bowl centrifuge
  - filtering centrifuge
  - oil conditioner
  - scroll centrifuge
- compressor
  - dynamic compressor
    - axial compressor
    - centrifugal compressor
      - high speed centrifugal compressor

- integrally geared centrifugal compressor
- regular API compressor
- mixed flow compressor
- turbocharger
- positive displacement compressor
- reciprocating compressor
- diaphragm compressor
- piston compressor
- labyrinth compressor
- rotary compressor
- liquid ring compressor
- vacuum pump
- screw compressor
- straight-lobe compressor
- vane compressor
- worm compressor
- thermal compressor
- vacuum compressor
- wet gas compressor

dynamic mixer

- axial turbine mixer
- bottom entry mixer
- propeller mixer
- radial turbine mixer
- tank mixer

engine

- combustion engine
- external combustion engine
- stirling engine
- internal combustion engine
- diesel engine
- head recovery diesel engine
- IDI diesel engine
- marine diesel engine
- stationary diesel engine
- gas engine
- petrol engine
- jet engine
- motocycle engine
- rotary engine
- spark ignition engine
- steam engine
- two stroke engine

expander

- cryogenic expander
- turbo expander

extruder

fan

- axial fan
- forced draft fan
- natural draft fan
- radial fan

motor

- air motor
- reciprocating air motor
- rotary air motor
- hydraulic motor
- hydraulic reciprocating motor
- hydraulic rotary motor
- outboard motor
- pneumatic motor
- servomotor

pump

etc.

10.3.2 Example of an indented Composition hierarchy of ‘Rotating Equipment’

- rotating equipment
  - bearing assembly
    - bearing
      - bearing bushing
      - locknut bearing
    - bearing cap
    - bearing circlip ring
    - bearing cover
    - bearing deflector
    - bearing housing
    - bearing housing radial
    - bearing housing thrust
    - bearing locknut
    - bearing lubrication ring
  - bearing bracket
  - casing drain
  - casing drain connector
  - casing nut
  - casing stud
  - casing vent
  - casing vent connector
  - casing wear ring
  - constant level oiler
  - cover
  - drain
  - grease nipple
  - heating/cooling jacket
  - hood
  - intermediate bearing
  - lifting lug
  - radial bearing
  - radial bearing bracket
  - seal
  - suction strainer
  - thrust bearing bracket
  - transmission system

11. Appendix B. Questions and Answers

11.1 Classes versus Characteristics (see par 5.4)

Originator: third Epistle Class Library workshop
Description of question:
Should we create a subtype kind of physical object (kind of object) when there is an alternative to define a 'kind of characteristic' for such a kind of object which can define the same information.
Solution:
The two ways of modelling are not each other's opposite, not does one exclude the other, because a subtype should be defined as having such a characteristic (value) by definition. Subtype kinds of physical objects should only be defined when it is widely recognised that there are common characteristics for all members of such a subtype, so that all the members of the subtype have that characteristic by definition.
The fact that characteristics of objects can be used to characterise them should not be of influence to the decision whether or not to define a kind of object (class).

As a general guideline it is recommended:

1. To distinguish clearly between classes of roles (sometimes called 'functional classes') and classes of physical objects. Kinds of roles don't define common mechanical aspects of the items that play the role. For example, 'driven end bearing' is a role, because the same bearing can be used as a non-driven end bearing.
2. To define only those kinds of roles (functional classes) that are already commonly used in the engineering practice.
   For example a 'driven end bearing' is not different from a 'back end bearing'. Nevertheless it may be useful to classify bearings according to their location, e.g. in order to capture differences in maintenance requirements. So the class 'driven end bearing' is a proper kind of role, but not a proper kind of physical object.
3. To define all kinds of characteristics for kinds of objects that are required to characterise an objects in that class. It is accepted that this may lead to duplicate information, which shall be kept consistent (which can be automated).
   For example, the class 'single stage pump' is a valid class. It implies that the class has a characteristic called 'number of stages' and that for the class 'single stage pump' the value of 'number of stages' = 1.

11.2 Characteristics required for Activities (see par. 5.7)

Originator: third Epistle Class Library workshop
Description:
Should we record characteristics of objects that are required for activities.
Solution:

Yes, all kinds of characteristics that are recognised as relevant for kinds of objects are valuable to record. Within ‘views’ it is possible to identify the characteristics that are relevant within the context of such a view.
The question which kinds of characteristics are relevant in which context of activity is complicated. In many cases the characteristic appears to be possessed by an object that is involved in the activity and not by the activity itself. The strategy for Gellish is to define characteristics of activities only for explicitly defined qualified activities, such as "exchange scenario's". An example of such an exchange scenario is the activity: **'order an item'.**

11.3 Requirements for detailed kinds of Systems

Originator: Christopher Sanders - M.W. Kellogg

Description:

Is there a need to identify and define all kinds of systems?

Solution:

The strategy is to define only kinds of systems when they appear to be used and valuable in practice, especially when knowledge or requirements are available that are specific for such a kind of system. The domain called *Facilities & Systems* contains a lot of kinds of systems. The amount of facts (knowledge) about those kinds of systems will grow especially their composition and intended role.

Note that such knowledge and requirements is typically not included in the public domain Gellish Dictionary / Taxonomy / Ontology, but may be included in proprietary systems and standards.
Definition of subtypes of systems become more valuable when those subtypes have defined standard configurations, such as the kind of system “API 614 compliant lubrication oil system”. Manufacturers may define further detailed standard product types by defining subsystems that are other assemblies of the same kinds of components with the same set of kinds of characteristics as the higher level ones, but different values.

For example, a 'methanol injection system' is a specialization of 'chemical injection system'.
The class 'chemical injection system' is a suitable class for a wide variety of chemical injection systems, irrespective of the substance that is injected (methanol or other). Often it is sufficient to express the fact that a specific system, say S-1 ‘is classified as a’ chemical injection system (with the description:) that is intended to inject methanol (or: that is a methanol injection system).

The ‘facilities and systems’ peer group will decide which kinds of systems will be standardised.

12. References

1. ISO 1087-1:2000, Terminology work - Vocabulary - part 1: Theory and application.
2. *Gellish Modeling Method – Part 2, Creation and Use of Domain Taxonomies*
3. *Gellish Modeling Method – Part 10, Gellish Dictionary/Taxonomy Development Manual*

   (see http://shop.gellish.net/)
4.