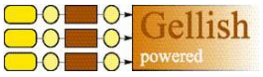


Gellish Modeling Method

Part 4
Managing Facility Information

Creation and Use of a Facility Information Model

dr. ir. Andries van Renssen

Version 3.0

July 2010

Table of content

1 Introduction 4
2 What is a Facility Information Model? 4
3 The Engineering Information Management process 6
4 Exchange of documents and data sets 9
4.1 Request for a work package for consultation or for modification 9
4.1.1 Transmittal for a request for a work package 9
4.1.2 Request for documents and data sets 10
4.1.3 Request for facility data 11
4.2 Exported work package (files) for consultation or modification 11
4.2.1 A directory with exported files 11
4.2.2 A directory with auxiliary files 12
4.3 Delivered work packages with new or modified documents and data sets 13
4.3.1 A directory with the physical files that are delivered. 13
4.3.2 A directory with auxiliary files 14
5 Procedure to create and extend a Facility Information Model 15
6 Step 1, Create a Facility Model composition hierarchy 16
6.1 Make coded names and unit names consistent 16
6.2 Classify equipment and units 17
6.3 Decompose the facility 17
7 Step 2, Create Physical File meta-data 18
7.1 Specify meta-data about the information (the content of the files). 19
7.1.1 Make document titles consistent 19
7.1.2 Qualification of information 20
7.1.3 Aggregate or decompose information 20
7.1.4 Version succession of documents and withdrawal 21
7.1.5 Arrange documents in a table of content 22
7.2 Relate documents to facility components 23
7.3 Relate documents to files at addresses 24
7.3.1 Multi-file documents 24
7.4 Decomposition of documents 25
8 Step 3, Specify data sets with additional Facility Model data 26
8.1 Specify fabricated items and relate them to designed items 26
8.2 Specify activities and processes (functions) of/in facility components 27

8.3 Specify aspects of facility components and products 29
9 Unique identifiers 30
10 Storing Data in a Gellish Universal Database 30
11 Knowledge-aided design (Propose_Facts) 31
12 Verification against requirements (Verify_Facts) 33
13 "Gellish Powered" certification criteria 35
13.1 What is a correct Gellish Facility Information Model 35
13.2 Requirements for Application Software 37
13.2.1 Viewing document files 38
13.2.2 Searching for individual things 39
13.2.3 Independency of sequence of expressions 39
13.2.4 Display of remarks on facts 39
13.2.5 Exclusion of strings in searches 39
14 Application system usage example 40
14.1 Preparation of the application system 40
14.1.1 Search engine installation 40
14.1.2 Loading the database 40
14.2 Search for documents 40
15 References 46
16 Appendix A, Summary of used relation types 46
17 Appendix B, Glossary 47

1 Introduction

This part of the Gellish Information Management Method series is concerned with the management and modeling of information about *individual things*. Other parts of the method deal with subjects such as Dictionaries (Definition Models), Knowledge Models, Requirements Models and Activity Models. They are mostly concerned with models of kinds of things.

This part of the method aims to increase the quality and availability of information (documents and data) about a facility and at the same time to reduce costs of managing data about the facility and its use. The basic vehicle for that is the development and use of an integrated electronic *Facility Information Model.* Such a model is a model of an individual thing that can be either a design of a particular object or it may be real a world object, or both. For example, it can be a design of a complete process plant, a building, a road, a ship or an airplane (which designs are imaginary individual things that only exist in peoples minds), and it can be an object that is fabricated, constructed and installed, operated and maintained.

This document specifies the procedure to create such a Facility Information Model and to modify and extend the information about the facility once the model is created. When a Facility Information Model concerns a plant, a product, a building, etc., then the result can be called a Plant Information Model, a Product Information Model, a Building Information Model, etc.

A Facility Information Model that is created according to the Gellish Modeling Method is documented in a system independent way in one or more standard Gellish Data Tables (see ref 1). This means that the model can be imported in any system that is able to read Gellish Data Tables. The internal data structure of such systems can also be in the form of a Gellish Database, but often it will differ. As the mapping to those internal system data structures is system dependent this document does not discuss any particular software system, but only deals with their data and document content in a system independent format.

Important kinds of systems in which Facility Information Models will most likely be implemented are document-oriented systems, such as Electronic Document Management Systems (EDMS's) that are extended with a part that describes the facility, Content Management Systems (CMS systems) or Enterprise Content Management Systems (ECM systems). Another kind of systems in which Facility Information Models may be implemented are more data oriented systems, such as Product Data Management Systems (PDM systems) and Product Lifecycle Management Systems (PLM systems).

2 What is a Facility Information Model?

Traditionally information about a facility is recorded in documents, first in paper documents, later in electronic documents. In a Facility Information Model a part of the information is expressed as data, which data reflects the facility, its components and its operation and possibly their properties, whereas the other part consists of the remaining documents and data sets in various formats. Each of those documents or data sets is then related to the element in the facility model about which the document or data set contains information.

This is illustrated in

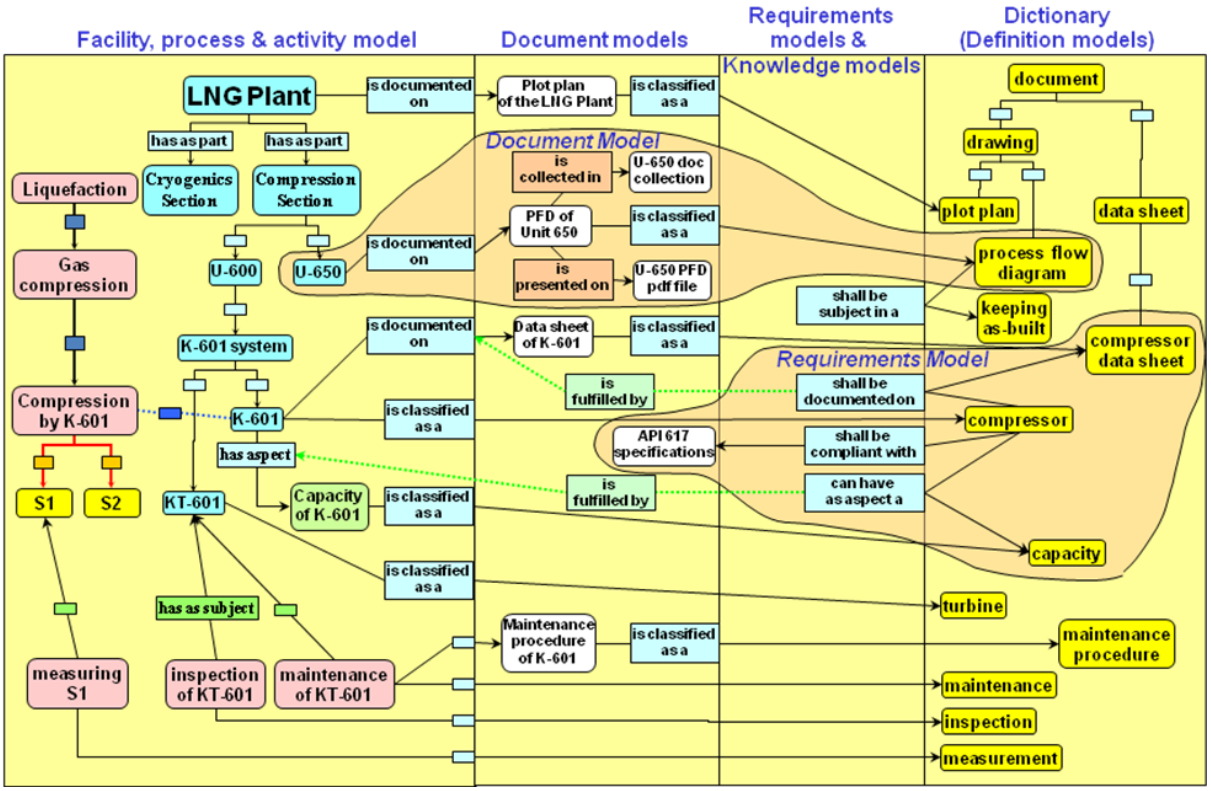


Figure 1.

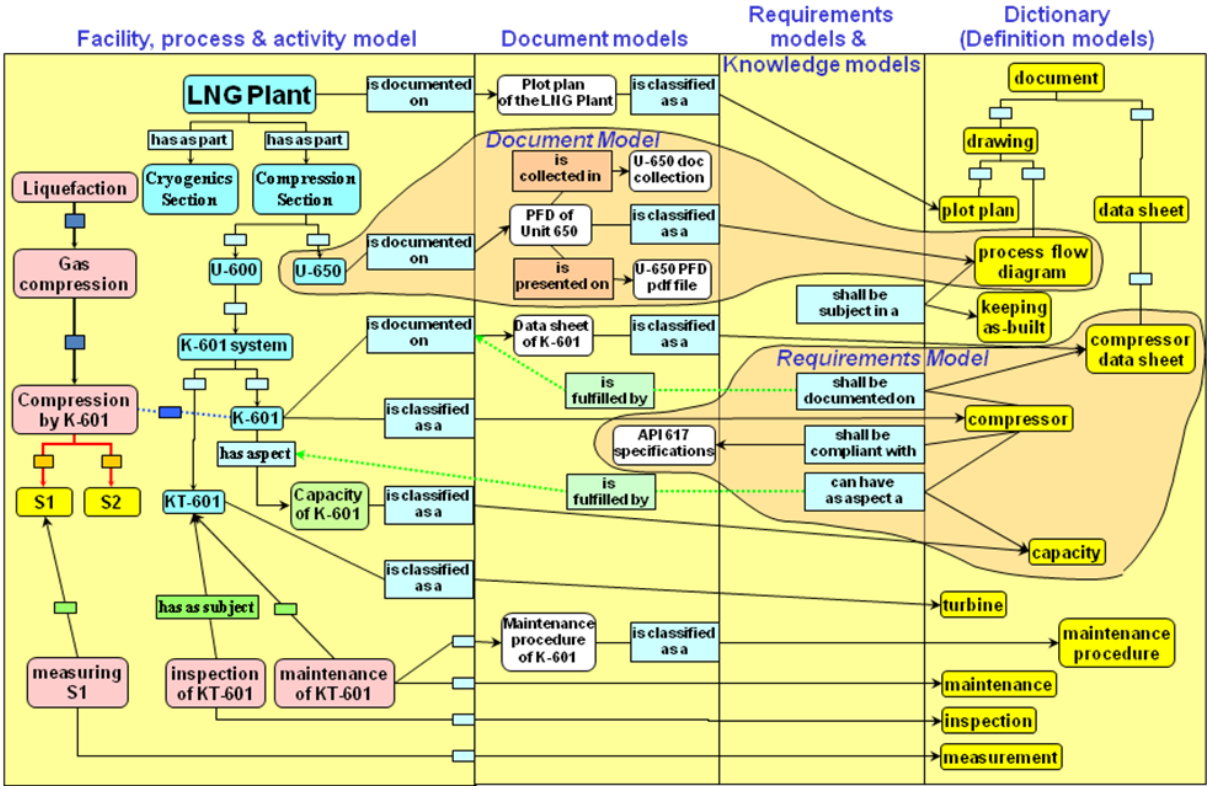


Figure 1, A facility information model of a process plant

In

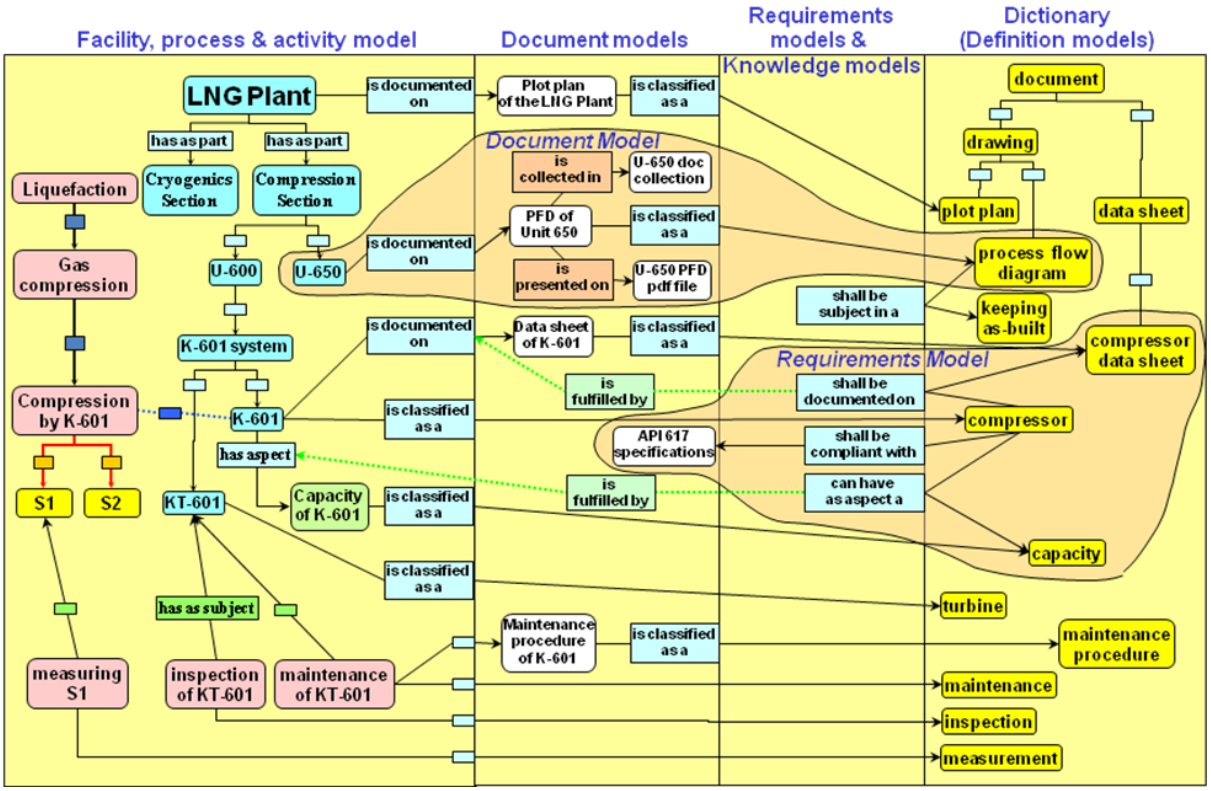


Figure 1 the left hand part includes the facility breakdown structure, together with the processes that take place in the facility and the activities on its components, as well as the properties of those components. The second part includes the document repository with electronic files and models of documents about the facility and its operation and maintenance. Those two parts form the core of the Facility Information Model. The small rectangles in the lines illustrate the relations between the components in the model, whereas different colours indicate different types of relations. These relations include not only relations between the components in the facility model, but also relations between the components and the processes, activities, properties and documents. Furthermore, each facility model component as well as each document is classified by a classification relation with a kind of thing in the Gellish English Dictionary on the right hand side of the figure (note that not all relations are drawn). That dictionary is therefore also an integral part of every Facility Information Model.

Note that the 'shall have…' and 'can have…'relations in the third box illustrate that it is possible to integrate requirements specifications and knowledge in the model as well. These requirements and knowledge or standard specifications are expressed as relations between kinds of things, whereas the definitions of those kinds of things are included in the Dictionary. The method to model specifications of requirements and knowledge is described in part 3 of the Gellish Modeling Method and is outside the scope of this document.

So, a Facility Information Model is the integration of a set of data and documents that model and describes the facility, its operation and maintenance, whereas each component and each document is classified by a concept (a class) that is defined in the Gellish English Dictionary.

A Facility Information Model can be implemented in various ways. The essence is that the user of a system by which the data and documents are accessed should experience it as one integrated system. Nevertheless, the system may be constructed in a way that the documents

are stored in a simple directory or in a way that they are stored in a separate document management system and the data are stored in one or more databases.

3 The Engineering Information Management process

The Engineering Information Management business process aims to ensure that complete and consistent information about a facility is readily available at the right time to the right persons.

Projects and Plant Changes not only create new documents and data sets, but they also modify documents and data sets that are already included in a Facility Information Model.

The process to modify and enhance an existing Facility Information Model requires that the existing documents and data sets are used as a basis for the modifications and that it is recorded which changes are made relative to the previous situation.

The process of Engineering Information Management (which is also called Asset Information Management (AIM)) therefore includes the creation of new documents and data sets as well as the revision process of documents and data sets and their content.

The Engineering Information Management business process is illustrated in Figure 2.

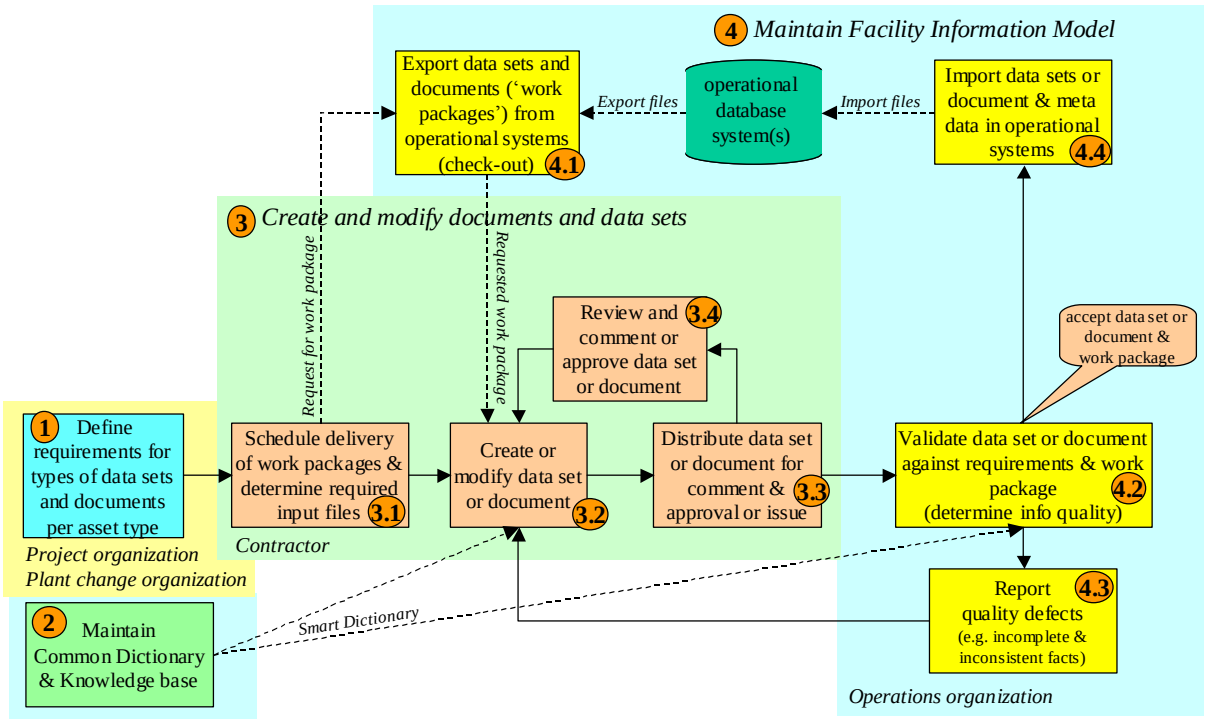


Figure 2, Engineering Information Management business process

In outline, the process covers the management of the quality of the four parts of the Facility Information Model creation and maintenance and therefore includes the following sub-processes:

1. Specify *Requirements* for types of data sets and types of documents.
   This part 4 of the Gellish Modeling Method assumes that requirements are already defined according to the method that is described in part 3.

2. Maintain an electronic *Common Dictionary and Knowledge base*.
   This part 4 of the Gellish Modeling Method assumes that a Smart Dictionary is available, and is created according the method that is described in part 2.
3. Create and modify *Documents and Data sets, including also data sets with meta-data*.
4. Create or maintain the *Facility Information Model and its tools*.

Training and support on the use of procedures and tools and the optimization of those tools are assumed to be included in the above-mentioned sub-processes.

This part 4 of the Gellish Modeling Method describes how to create and maintain data sets, documents and meta-data, such that they can be managed and easily retrieved via an integrated and consistent Facility Information Model. This includes guidelines for the following elements of the Engineering Information Management sub-process 3 and 4:

3.1 Determine the required input files.
This sub-process includes that the party who will create of modify documents or data sets (typically a contractor) shall issue a request for delivery of existing documents and data sets (a work package). This includes a specification of the existing documents and data sets about the facility that are required as input files for the creation and/or modification of documents and data sets. The required files may be either for information or for modification. This shall result in a *request for existing documents and data sets (a work package),* which shall include a request for a range of identifiers ("document and drawing numbers") for new document and drawings, when applicable. A request for existing documents and data sets shall have the form of a data set as is described below.

4.1 Export of work packages.
This sub-process is only applicable in cases where documents or data sets already exist.
The export includes the creation of a collection of files with electronic copies of documents and data sets (*exported work packages*) with information about the current situation. The files are extracted from the Facility Information Model as implemented in the operational systems. The files are determined on the basis of a data set that contains a request for existing documents and data sets as issued by the party that will consult, create and/or modify documents and data sets and therefore needs them as a basis for its work. This sub-process implies that exported files are arranged in work packages that shall be prepared and supplied to the requester.

A data set with transmittal data shall be delivered to the modifying party. After verification of correct arrival the receiver shall return an amended copy, which shall include a confirmation of correct arrival.

The sub-process shall include the recording of check-outs, which means that it shall be recorded by the operational organization which documents and data sets are delivered to a modifying party for modification. This can be a single check-out or it can include multiple check-outs for the same document or data set.

a) Check-out data.
   The exported documents and data sets shall be checked-out. Check-out means that it is recorded that the checked-out document or data set is recorded to be a subject in a modification activity by a particular party that intents to modify the document or data set, whereas that party is provided with a copy of it.

b) Multiple check-out data (Concurrent engineering).
If another party also wants to modify a checked-out document or data set, then that document or data set is registered as being also a subject in another modification activity by that other party. Such a multiple check-out shall be accompanied by a warning to both parties that concurrent engineering is taking place on different copies, and that consolidation of the two modifications should take place after completion of one of the modifications. It shall also be agreed who will do the consolidation.

3.2 Creation of work packages.

The party that creates or modifies documents and data sets shall not only create the documents and data sets themselves, but shall also create data sets with information about the documents and data sets as well as data sets with information about the new and changed objects in the facility model. This means that for every created file the delivering party shall also prepare a contribution to the data set with meta-data and with facility data.

3.3 and 3.4 Distribute for review or approval or issue documents and data sets.

Each file with a (composed) document or data set, the relevant facts from the file with meta-data and the relevant part of the file with facility model changes shall be distributed for review or approval together with a review transmittal data set that specifies what is expected from the reviewer/approver. The review and approval process implies that a data set is available that specifies per document or data set type which persons shall review such a file and what actions are expected. The comments or reviewers shall be acted upon and their approvals shall be collected.

When the files are approved they shall be issued to the operations organization, together with a delivery transmittal data set.
So in this sub-process there are two transmittal data sets that play a role:

a) A data set with review/approval transmittal data.
A data set that specifies for its receiver which files he or she receives and what actions are expected from him or her.

b) A data set with delivery transmittal data.
A data set with transmittal data shall be delivered to the operational organization. The transmittal data set specifies the delivered files with documents and data sets, the file with meta-data and the file with facility model changes. The operational organization shall issue a return transmittal data set to the delivering party to confirm correct arrival and acceptance of the package mentioned in the transmittal.

4.2 and 4.3. Quality verification of documents and data sets and meta-data.

The operations organization that receives a transmittal with the auxiliary directories with files shall verify the file contents and the information about the files (document titles, etc.). Any quality defects shall be registered and mitigation action shall be taken to improve the quality of the delivered information.

4.4 Import of documents and data sets and meta-data in the Facility Information Model.

When the information passed the quality verification process the files will be imported in the operational systems and the computer interpretable data sets will be uploaded as data in the appropriate databases.

4 Exchange of documents and data sets

This chapter describes the exchange 'messages' and the data sets with data about the documents and data sets that are exchanged during the above-mentioned sub-processes.

4.1 Request for a work package for consultation or for modification

Typically a design office or an engineering contractor who needs existing documentation as a basis for the execution of a project or plant change will issue a request for delivery of a work package, being a collection of existing documents and data sets. Some of those documents and data sets are intended to be consulted only, others will be modified and replaced by newer versions.

The process to request and create the work package is illustrated in Figure 3.

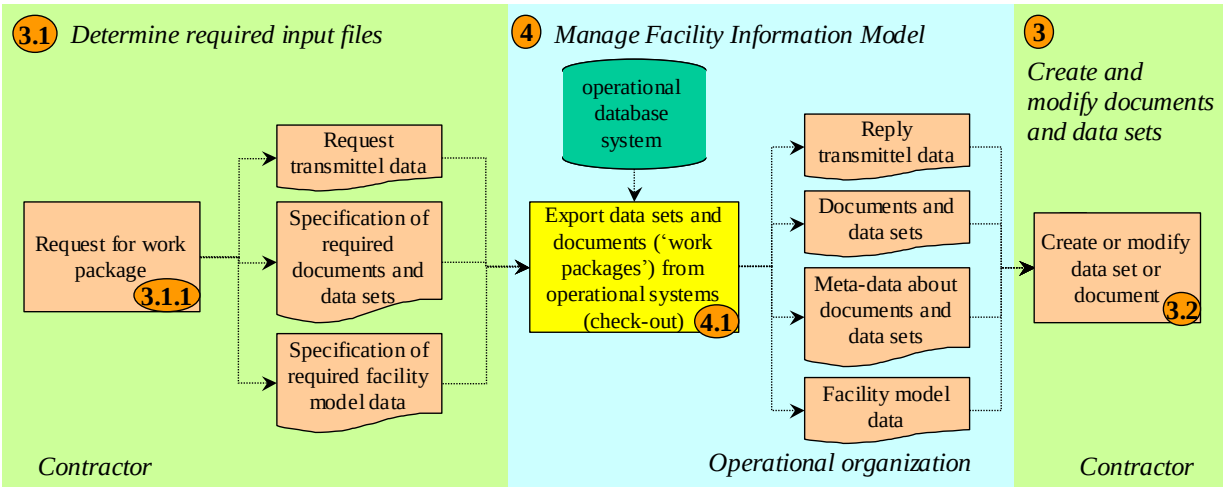


Figure 3, Request for and delivery of a work package

Sub-process 3.1 illustrates that the determination of the required documents and data sets results in a request in the form of a Gellish Data Exchange Message that consists of three sections that are describes in the next paragraphs. Each of the three sections will consist of a collection of lines in one Gellish Data Table that is sent to the organization that is responsible for the delivery of the existing information. The request initiates a query for retrieval and export of the existing information that results in four sections. Those sections are discussed in the following paragraphs. The results are then available for being consulted and modified and as a basis for the creation of additional documents and data sets.

The various sections are discussed in the following paragraphs.

4.1.1 Transmittal for a request for a work package

The first section is a transmittal data set, which consists of a kind of envelop or header data that accompanies and describes the originator and addressee. The core of such a data set is typically expressed in a Gellish Data Table as follows:

UID of left hand object	Name of left hand object	Communicative intent	UID of Fact	UID of Relation type	Name of relation type	UID of right hand object	Name of right hand object
600	request for documents-1		700	4,760	has as subject	601	collection of documents-1
600	request for documents-1		701	1,225	has as originator	602	Fred
600	request for documents-1	request for delivery	702	1,225	has as addressee	603	John
600	request for documents-1		703	1,225	is classified as a	194,601	request for a delivery
601	collection of documents-1		704	1,225	is classified as a	490,036	collection of documents

These transmittal data can be completed with explicit information about the location of delivery, the date of delivery, the conditions, etc.

Each of these facts belongs to a collection of facts about a particular transmittal, such as 'facts about transmittal-1' (recorded in column 50 and 68 of a Gellish Data Table).

4.1.2 Request for documents and data sets

The content of a request shall consist of the following information:

For each requested document or data set:

1. The Document Identifier of the required document or data set.
2. The purpose for which the information is required. This is either for consultation or for modification. If a document or data set is requested for consultation only, then the document is not checked-out.
3. The party that is responsible for the modification and the delivery of the modified version.
4. The expected delivery date of the modified version.

For example, assume that drawing T-123 need to be modified and drawing T-124 is required for consultation only. Then the request for delivery of T-123 and T-124 is specified in a system independent way in a Gellish Data Table as follows.

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
611	T-123	711	4,760	is subject in	612	modification of T-123
611	T-123	712	1,227	is an element of	601	collection of documents-1
612	modification of T-123	713	1,225	is classified as a	194,367	modification
612	modification of T-123	714	4,761	is performed by	613	process engineer X
612	modification of T-123	715	5,124	has as delivery date	20,081,231	31Dec2008
614	T-124	716	4,760	is subject in	615	consultation of T-124
614	T-124	717	1,227	is an element of	601	collection of documents-1
615	consultation of T-124	718	1,225	is classified as a	193,480	consultation

The above table column 'Communicative intent' indicates the purpose with which the facts 701 and 705 are communicated. When a field in that column is empty, then the default value is that it is communicated as a statement (for information).

The table also illustrates that the modification activity and the consultation activity is explicitly created. This enables a linking with the activity scheduling. The classification as consultation implies that the document is not checked-out is should not be returned. Therefore, neither a performer, nor a delivery date is required.

Note that the UID of the delivery date reflects the date in the format yyyymmdd (20081231).

4.1.3 Request for facility data

A request can also be for data about a facility component or activity. Such a request consists of a Gellish Query, for example a general request for 'all facts' that are known about a particular object. This can be done using the Retrieve_Facts function and the standard query 'All_Facts' that are included in the Gellish Universal Database API (see Ref. 3). It can also be an ad hoc request for specific information. Such a specific request is also expressed as a Gellish Query.

A request for facts about a particular individual object can be expressed as a relation with an unknown object. Such unknown objects have their own UID in the query. For that purpose the UID's 1-100 are reserved and names 'what', 'which', etc. can be used as names for the unknown objects.

For example, the very general relation type <is related to > can be used to query about any kind of relation with an object, because all relation types that are a subtype of that general one will be selected as well. The resulting generic query is expressed as follows:

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	UID of collection of facts	Name of collection of facts
621	P-1201	721	4,658	is related to	1	what		

Such a Gellish Query can in principle be processed automatically by agent software that implemented the Gellish Universal Database API and that operates on a central or distributed Gellish Universal Database.

4.2 Exported work package (files) for consultation or modification

A work package is typically created by the organization that manages the master copies of the documents and data sets for a facility. An exported work package shall consist of collections of files, arranged in two directories as described below.

The files that are described below contain data that are exported from application systems in which the data already exist. Therefore, it is not described here how such data is created. The description below only indicated how such data should be presented in Gellish Data Tables or in the form of a Gellish Message.

4.2.1 A directory with exported files

A directory with exported files shall contain the files with copies of the documents and data sets with information about the current situation, arranged in one or more work packages.

The directory shall have the name of the collection of files concatenated with the date of preparation.
For example, the directory 'Unit 1300 documents 30Sep2008' may contain a number of drawings and other documents about unit 1300 as exported on the specified date.

4.2.2 A directory with auxiliary files

The directory typically contains three auxiliary files:

1. A file with export transmittal data
2. A file with meta-data
3. A file with facility model data

The directory shall have the name of the collection of files, extended with a dash embedded in spaces (‘ - ‘) and the term auxiliary files, before the date.
For example, the auxiliary directory of the above-mentioned directory should be called: ‘Unit 1300 documents - auxiliary files 30Sep2008’.

The three files contain the following three data sets:

1. **A data set with export transmittal data.**

   A data set with transmittal data about the exported files shall contain the following data:

UID of left hand object	Name of left hand object	Communicative intention	UID of Fact	UID of Relation type	Name of relation type	UID of right hand object	Name of right hand object
631	delivery of work package-1		731	4,760	has as subject	632	work package-1
631	delivery of work package-1		732	1,225	has as originator	603	John
631	delivery of work package-1	information	733	1,225	has as addressee	602	Fred
631	delivery of work package-1		734	5,660	is a response to	600	request for documents-1
631	delivery of work package-1		735	1,225	is classified as a	193,350	delivery
632	work package-1		736	1,225	is classified as a	492,018	collection of electronic data files

2. **A data set with meta-data.**

   This file shall contain data about the files in the directory with exported files and data about the documents and data sets in those files. The data set shall also specify which documents and data sets are arranged in (multi-file) composed documents and which (composed) documents and data sets are the elements in the work packages.

   For example: Assume that document T-123 was requested for delivery. Then it might be that it appears that T-123 is a composed document that consists of documents, T-123 sheet 1 through 4, whereas some of them have multiple versions. This means that four files will be delivered with the latest version of each document. A detailed description of the process to determine which files should be supplied is beyond the current scope of this document.

   The resulting data set, with in this example only one file, is as follows:

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
108,000,600	T-123 sheet 1 of 4 - rev. A	208,301,124	4,996	is presented on	108,300,308	BRPFD09.dwg
108,000,600	T-123 sheet 1 of 4 - rev. A	208,000,600	1,225	is classified as a	490,183	P&ID
108,300,308	BRPFD09.dwg	208,302,224	1,225	is classified as a	490,533	electronic data file
108,300,308	BRPFD09.dwg	731	1,227	is an element of	108,300,567	C:\LNG\Files
108,300,567	C:\LNG\Files	732	1,225	is classified as a	492,017	directory
108,300,308	BRPFD09.dwg	733	1,227	is an element of	625	work package-1

3. **A data set with the facility model data.**

   This data set is the response to the request for data about a facility component or activity. The data set contains the results of the query that is specified in the request.

   For example, the query for data about P-1201 may result for example in a data set with the following facts:

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	UID of UoM	Unit of measure
610	P-1201	720	1,225	is classified as a	130,058	centrifugal pump		
610	P-1201	721	1,190	is a part of	622	U-1200		
610	P-1201	722	1,727	has as aspect	623	capacity of P-1201		
623	capacity of P-1201	723	1,225	is classified as a	550,318	capacity (volume flow rate)		
623	capacity of P-1201	724	5,025	has on scale a value equal to	920,774	26.5	570,449	dm3/s

4.3 Delivered work packages with new or modified documents and data sets

A delivered work package with a collection of files that contains new and/or modified documents and data sets shall be arranged in two directories. The two directories shall contain the following files:

4.3.1 A directory with the physical files that are delivered.

This directory shall contain the delivered files with documents and data sets. The directory shall have the name of the collection of files concatenated with the date of preparation.
For example, the delivery directory ‘Unit 1300 documents 30Sep2008’ may contain a number of drawings and other documents about unit 1300 as prepared on the specified date.

There are two categories of physical files:

- Physical files that contain data sets that are interpretable by the system that implements the Facility Information Model, especially physical files with facts that are expressed in Gellish Data Tables. The content of these files can be uploaded in the databases of operational application systems.

- Physical files that contain documents or data sets that are only interpretable by dedicated software and for which such software has to be launched in order to visualize the content for a user. For example, pdf files, and files in native file formats of AutoCAD, MS Word, etc.

4.3.2 A directory with auxiliary files

The directory typically contains three auxiliary files:

1. A file with meta-data
2. A file with facility model data
3. A file with delivery transmittal data

The directory shall have the name of the collection of files, extended with a dash embedded in spaces (‘ - ‘) and the term auxiliary files, before the date.
For example, the auxiliary directory of the above-mentioned directory should be called: ‘Unit 1300 documents - auxiliary files 30Sep2008’.

The three auxiliary files contain the following information:

1. A file that contains a data set with delivery transmittal data.

   A data set with transmittal data is a data set that is intended to accompany a collection of one or more documents and data sets and that explains what the collection is about, why it should receive the addressees consideration, and what the addressee should do with it. It often contains important dates or deadlines that the addressee should be made aware of.

2. A file that contains a data set with meta-data.

   This data set includes information (facts) about each physical file in the other directory (thus including also the inventory of files that are delivered) and information about the documents and data set that are contained in those files. The data set shall also specify which documents and data sets are a part of composed documents. It also specifies which (composed) documents are elements of work packages. The delivered work packages may be deliveries of modified partial or complete export work packages. This data set shall indicate also which files are new and which files are revisions of earlier files.

   The data set with meta-data shall also contain a specification of the format in which the information in the physical file is encoded. This determines whether the physical file is computer interpretable and thus whether its content can be imported in the database of a system that implements the Facility Information Model.

3. A file that contains a data set with the facility model changes.
   The modifying party shall provide a data set that specifies additions, deletions and modifications of facility components and their relations, relative to the previous state of the model in the context of the delivered documents and data sets. This includes among others data sets with computer interpretable equipment lists that specify the changes relative to the previous versions of the lists. In particular a facility decomposition hierarchy and data about its components.

5 Procedure to create and extend a Facility Information Model

The scope of a Facility Information Model will depend on the available material, the phase in the (project) lifecycle of the facilities and the application purpose. The creation of the facility information model may start for example with source material such as:

- Equipment lists, including information about related equipment/assemblies and documents about the equipment.
- Document files and document indexes.
- Process Flow Diagrams, Piping and Instrumentation Diagrams, Plot Plans, etc.

The process to creates or extends an electronic Facility Information Model shall be preceded by loading the electronic common dictionary as far as available and loading the available requirements for documents and data sets.

The process to create or extent a facility model itself then consists of the following steps:

1 Create or extent a Facility Model decomposition hierarchy (also called an asset breakdown structure).
This means: create a data set that specifies the facility decomposition structure.
This includes the following actions:

1.1 Make coded names (tag names) and unit names consistent and compliant with a standardized convention.

1.2 Classify equipment and units
Classify the site, the plant(s), process units, systems and main equipment.

1.3 Decompose the facility
Specify for each component of which assembly (or assemblies) it is a part.
Add equipment systems and include them in the decomposition hierarchy.

2 Create Physical File meta-data.
This means: specify a data set with meta-data for each physical file (paper, electronic or other medium). It includes the following actions:

2.1 *Specify meta-data for the documents (the content of the files).*

2.1.1 Make document titles consistent

2.1.2 Classify documents
Classify each document by standard document types.

2.1.3 Decompose documents
For some documents (e.g. binders or multi-page drawings?) a decomposition shall be specified.

2.1.4 Version succession of documents
Documents that are succeeded by new versions shall be related to their successor.

2.2 *Relate documents to equipment, area's or processes.*
Specify for each equipment item, area or process on which document it appears, or for each document about which equipment items, area or process it contains information.

2.3 *Relate documents to files at addresses.*
Specify for each document on which a physical medium it is presented, either as an electronic data file or on paper, microfilm or any other medium.

3 *Specify data sets with additional Facility Model data*

3.1 *Specify fabricated items and relate them to designed items*
Specify for each installed item for which design item (tagged item) it is installed and identify the manufacturer's models that are applied.

3.2 *Specify activities and processes (functions) of/in facility components*
Specify which processes (functions) are or will be performed in or by the facility and its components and which activities are performed by people that operate or maintain the facility and its components.

3.3 *Specify aspects (properties and qualities) of facility components and products*
Specify the dimensions and other qualitative and quantitative aspects of the facility and its components and possibly of the activities, processes and products (fluids or solids).

6 Step 1, Create a Facility Model composition hierarchy

The first step to create a Facility Information Model is to create a Gellish Database (one or more 'data more 'data sets' or Gellish Data Tables) that specifies how the facility is composed. Such a facility facility composition hierarchy is sometimes called an asset breakdown structure). Such a structure is structure is illustrated by the blue part in the left hand section of

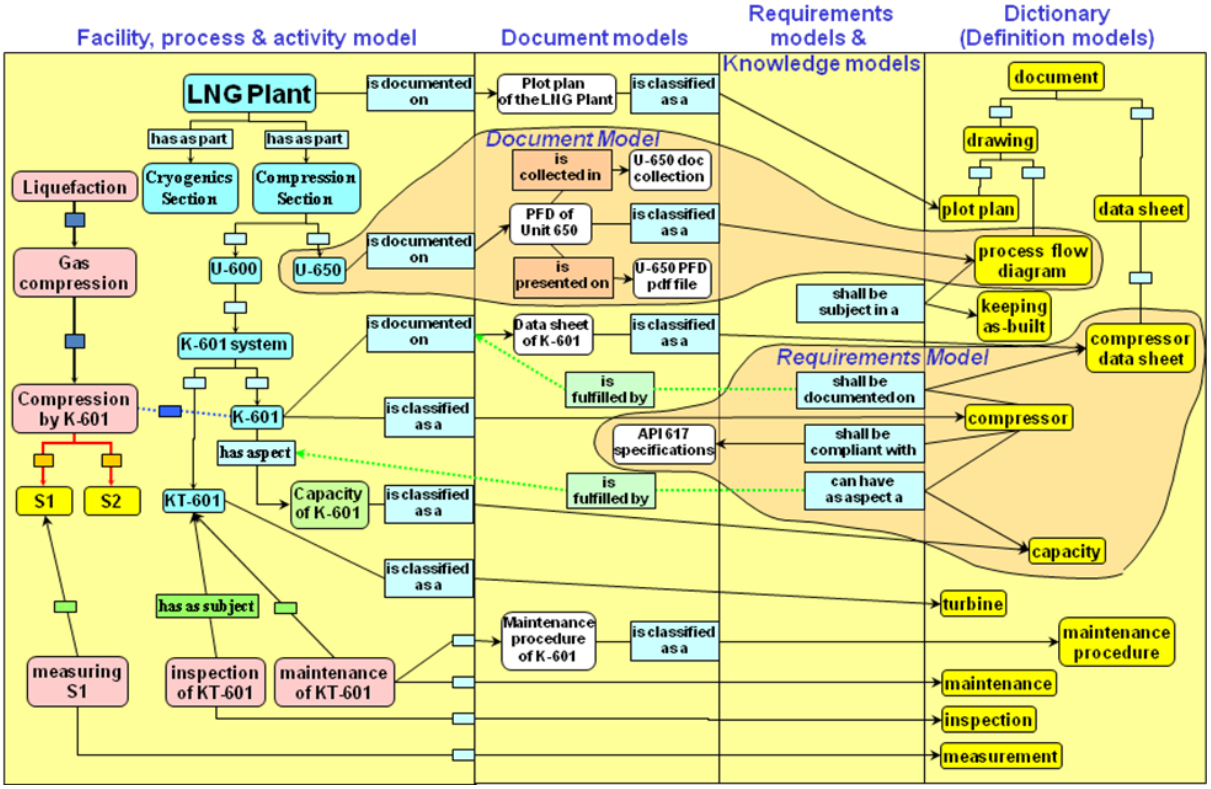


Figure 1. Such a table is created as follows.

6.1 Make coded names and unit names consistent

To ensure that the Facility Information Model becomes a consistent whole it is essential to ensure that coded names (tag names) and names of facility, sections, units, systems, equipment, instruments and components are consistent with a defined naming convention. This means that a coding system and naming convention shall be adopted.

For existing facilities with a history of more or less consistent numbering and naming conventions, it is usually not justifiable to modify existing names, such as tag names and document titles. Nevertheless, the names in the Facility Information Model database may be changed without changing the information in the documents and synonyms may be added that increase the consistency of the names and titles.

For example, actions to make names consistent may include:

- Add or remove dashes and spaces, names in proper case and classes in lower case, correct typing errors, and change abbreviations to full terms.
- Harmonize codes for equipment types.
  For example:
    - choose FL or FIL in case of filters, but not both, or add synonyms if existing conventions should be retained.
    - choose SV or PSV in case of safety valves, but not both, or add synonyms if existing conventions should be retained.
- Remove use of plural or add singles as synonyms and define collections.

If old names are and will remain in use, then the old names need to be retained in a mapping (synonyms) table, in order to build up a table for automated renaming in case the old conventions are used again.

6.2 Classify equipment and units

The total facility and all its components at various aggregation levels, including systems and equipment need to be classified by standard classes that are selected from the Gellish Dictionary.

Each item shall be given a unique identifier (UID) in the range that is allocated for the category in the project (see paragraph 0 Unique Identifiers).

Each classification relation shall have a status of validity of the fact (not shown in the table below), which initially may be ‘proposed’ and finally shall be ‘accepted’ for valid facts. Such a classification relation records the existence of the item. If the item is removed from the facility, then the status shall be set to ‘deleted’ or ‘replaced’, which indicates that the item does not exist anymore.

For the classification relation the standard relation type ‘is classified as a’ shall be used (selected from the STEPlib Dictionary).

This shall result in a Gellish Data Table of which an example of the core is as follows:

UID of left hand object	Name of left hand object	Old name	UID of Fact	UID of Relation type	Name of relation type	UID of right hand object	Name of right hand object
108,000,000	LNG site	LNG site	208,000,000	1,225	is classified as a	700,000	site
108,000,062	Air Compressor System	Air Compressor	208,000,080	1,225	is classified as a	130,073	compressor system
108,000,001	LNG plant	LNG plant	208,000,001	1,225	is classified as a	160,081	process plant

The consistency of this table should be verified using the Gellish Browser after the table is completed by filling the column 'Name of the Context for the Left hand object' and after completing the top three rows with the numbers of the columns as specified in the Gellish Database Definition document (ref. 1.)

Missing data, duplicate lines and inconsistent data shall be corrected and fed back to the project team.

6.3 Decompose the facility

For each component in the facility it shall be specified of which assembly (or assemblies) it is a part. When this is done in a consistent way, then a complete facility breakdown structure is created. This enables navigation through the facilities via a computer for finding data and documents.

This shall result in a Gellish Data Table. The core of such a table is illustrated by the following example:

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
108,000,062	Air Compressor System	208,000,080	1,190	is a part of	108,000,156	Unit-1300
108,000,156	Unit-1300	208,000,307	1,190	is a part of	108,000,001	LNG plant
108,000,001	LNG plant	208,000,002	1,190	is a part of	108,000,000	LNG site

The example illustrates that the same object that appears on the left hand side on a line with a role as part in a part-whole relation may appear on another line on the right hand side with a role as a whole in another part-whole relation.

Note that the lines may be included in any sequence. The sequence has no impact on the meaning.

If no equipment systems would be defined, then drivers, gearboxes, and other auxiliary equipment cannot easily be arranged in the decomposition hierarchy. That is the reason why it is required to define equipment systems and to define that the main equipment items are a part of their respective systems. Those systems shall be included in the decomposition hierarchy. Each system shall be classified and shall be given its own UID.

> Note: A system name shall be created by a combination of the tag name of the main equipment item, a space and the term 'System'. For example: P-1265 System. A macro in Excel is available that allocates UID's for items that have a name, by copying the UID's from the lines where the items are classified.

To facilitate searching it is recommended to include that a piece of equipment is part of an equipment system as well as a part of a process unit, whereas the equipment system is also

specified to be a part of the process unit. For example, it is recommended to include all following three facts:

P-1265	is a part of	P-1265 system
P-1265	is a part of	Boiler unit
P-1265 system	is a part of	Boiler unit

The resulting Gellish Data Table shall be verified using the Gellish Browser.

7 Step 2, Create Physical File meta-data

In information management we make a distinction between two different kinds of objects:

- physical information carriers
- documents, being the content of such carriers.

For example, and a paper copy that all contain the same information.

Multiple information carriers can contain the same information. This is the reason why we distinguish between documents and information carriers and therefore a document (a 'common content') is a separate object that can have a relation with multiple physical files, as is described in paragraph 7.3.

A document is defined as (a collection of) information that forms the common content of one or more physical information carriers. An information carrier is a physical object on which the information is presented. Examples of information carriers are: a sheet of paper, a bundle of sheets of paper, an electronic file, such as a doc file or a pdf file, it can be a collection of electronic files (possibly collected in a directory).

So a document is defined here as either:

- A drawing in a particular version. Its identifier therefore includes its revision id.
  For example: Drawing 'T-123 sheet 1 of 4 - rev. A'.
- A data set in a particular version.
  For example: Spreadsheet 'U-1200 Equipment list – rev. B'
- A multi-page document in a particular version.
  For example: Report 'R-123 version 2.1'.
- An information content of any other medium in a particular version.
- A collection of documents.
  For example: the collection of drawings on different sheets that make up T-123 or the content of binder 'Volume 2 Gas turbine Equipment Operation'.

When we use the term 'document', then we mean any of such common content. A document is typically in textual form. However, content can also be a drawing, images or sound or information in another form. It can even be a computer program (software). However, for readability reasons we keep using the term document, although the term can denote information content in any form on any medium.

When an information carrier contains content in tabular form, whereas the table columns and rows have defined meanings, then we usually call such a document a 'data set'. A data set can be the content of an electronic data file or a database. Examples of data sets are: the content of

a spreadsheet or the content of an MS-Access or Oracle database. The physical information carriers are then the xls files, mdb files, etc.

This chapter describes how to create a Gellish Data Table with data about a physical file and about the document(s) that is (are) contained in those file(s). Such data about documents and files is also called meta-data about document, which thus also includes information about drawings, data sets, etc. and the files that contain them.

The chapter consists of three parts:

1. Meta-data about the documents (the content of the files).
2. Meta-data that relate the documents to the facility and its components, area's and processes.
3. Meta-data about the physical files and their relations to documents.

7.1 Specify meta-data about the information (the content of the files).

7.1.1 Make document titles consistent

Document titles and data set titles shall be compliant with 'Requirements for Information about Documents' (ref. 2). If existing documents or data sets need to be included in a Facility Information Model that do not comply with those requirements, then they shall be improved according the guidelines in that document. However, it may be decided to improve only the titles in the database and not change the titles on the physical documents (files). The mapping to the old titles shall then be retained in a separate column of the mapping table in preparation of automated conversion in case information is delivered using the old document titles again.

7.1.2 Qualification of information

Each document (thus including also data sets) shall be a qualification (according to its content) of a type of information. Typically that type is a standard document type or type of data set that is selected from the Gellish Dictionary. Such a qualification relation (1726) specifies the existence of the qualitative information. It is also possible to give the qualitative information a name and to include the information as full text in the description of the qualitative information. For example a piece of information that is a requirement for bridges can be called requirement 35, and such a qualitative requirement may be specified in full text, whereas the information on a drawing is only denoted by the drawing number and possible a title. These two examples result in the following Gellish Data Table:

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	Description	Status
108,000,511	requirement 35	208,000,511	1,726	is a qualification of	970,007	requirement	A bridge shall be able to carry a load that is …	accepted
108,000,600	T-123 sheet 1 of 4 - rev. A	208,000,600	1,726	is a qualification of	490,183	P&ID	P&ID of Unit 3100	accepted

A Unique Identifier (UID) shall be allocated to each (piece of) information and to each piece of aggregated information.

Each qualification relation shall have a status of validity of the fact. The status may be 'proposed', 'accepted', 'deleted' or 'replaced'. If the status is 'deleted' or 'replaced', then the information is archived or its physical (master) copy(s) may be destroyed.

The relation between the information and the information carriers is discussed in paragraph 7.3.

7.1.3 Aggregate or decompose information

For some document types (such as binder contents or multi-page drawings) it may be valuable to define also one or more higher levels of aggregated information. For example, it may be decided to arrange information in sections or in packages (also physically, such as in binders), whereas the sections may be aggregated in volumes. For example, it is recommended to include the pages of multi-page documents in an aggregated document, especially because some information will be valid for the whole content and references may be may to the whole, instead of to the separate sheets or versions. Typically the structure of the aggregated information will reflect a table of content, as it will describe which pieces of information are included in which section or whole of the aggregated information, such as a package.

This should be done as follows:

1. Define the aggregated information and qualify it. For example as a package.
2. Define pieces (as required), being parts of the information that are included in the larger piece of information and qualify those pieces. For example as section, or as subsection or package.
3. Define for each document (or sheet) that it is part of a particular subset collection.
4. Define for each piece that it is included in the larger aggregated information.
5. If an electronic file is created for aggregated information, then it shall be specified that the information <is presented on> a particular electronic file, in the same way as is done for documents. This is further described (and illustrated) in par. 7.3.

This will result in a Gellish Data Table as follows:

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
108,000,601	LNG Plant Documentation	208,000,601	1,726	is a qualification of	970,492	aggregated information
108,000,602	Section 1, Heat exchangers	208,000,602	1,726	is a qualification of	970,493	information in section
108,000,602	Section 1, Heat exchangers	208,000,603	5630	is information that is included in	108,000,601	LNG Plant Documentation
108,000,603	E-6004 construction drawing	208,000,604	5630	is information that is included in	108,000,602	Section 1, Heat exchangers

Note that the above table does not describe a physical drawing, neither on paper, nor as an electronic file, but only the (common) content of such paper copies and files.

7.1.4 Version succession of documents and withdrawal

Documents that are succeeded by new versions shall be related to their successor. This is done by specifying a version succession relation between the succeeded document and the successor document.

For example:

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
108,301,080	LD1-Rev. A	208,301,123	1,393	is a next version of	108,300,046	LD1-Rev. O
108,302,345	PP11-1 sheet 1 of 2 Rev. B	208,302,222	1,393	is a next version of	108,300,160	PP11-1 sheet 1 of 2 Rev. A

If documents are withdrawn, then the status of its classification fact shall be modified from 'accepted' into 'deleted' or 'replaced'. In the case a document is replaced by another document, then on the line in the table with the classification fact of the replaced document it shall be recorded by which new document it is replaced.

For example, this shall be recorded as follows in a Gellish Data Table: One fact is replacing the other fact. Therefore, the UID of the fact that defines the new document by a qualification relation shall be included in the column called 'UID of successor of fact' on the line that defined the replaced document by a qualification relation. The status of the fact about the qualification of the replaced document shall become 'replaced'.

For example:

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	Status	UID of successor of fact
108,300,046	LD1-Rev. O	208,201,501	1,726	is a qualification of	490,183	P&ID	replaced	208,202,502
108,301,080	LD1-Rev. A	208,202,502	1,726	is a qualification of	490,183	P&ID	accepted	

7.1.5 Arrange documents in a table of content

A large number of physical documents of the archive may be filed in binders. There is usually an overall table of content of the binders and per binder there may then be a detailed table of content (also called a document index). A table of content arranges documents, usually in a logical order. Therefore, that order is reflected in the database, to facilitate searching of documents.

For example, the table of content of a 'Mechanical Catalogue' part of an archive is built-up as follows:

- The individual documents are collected in purchase order packages, indicated by a HEG-number (e.g. HEG36721 Electric Motors).
- For a number of individual documents it is indicated about which piece(s) of equipment the document contains information (tag number).
- Each purchase order package is collected in a Section or a Subsection. A Section or Subsection is a logical grouping of documents of similar nature. For example, all documents about heat exchangers are collected in Section 8 Heat Exchangers and Subsection 15 G contains documents about Salt Water Screens.
- Each Section or Subsection covers either a part of a binder, a full binder, or it covers several binders. This means that Sections or Subsections may be segmented over different binders.

For example, Subsection 15 G is split over Volume 39 and 40 and all documents about HEG36501 Nitrogen Generation Units are collected in Section 16, which consists of three subsets, which are distributed over Volume 41, 42 and 43.

- The collection of sections forms the total archive.

The distribution over Volumes (binders) is not relevant for an electronic archive. This implies that the Volumes might be eliminated of the table of content. However, it might be relevant to maintain the relation with an existing paper based situation. Therefore, the distribution over the Volumes can be reflected also in the database.

As a result, the above information is recorded in a database table as follows:

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
108,300,656	365-PP12-1-V28-59-1	208,302,879	1,726	is a qualification of	490196	drawing
108,200,704	NG-601	208,302,980	5,046	is described in	108,300,656	365-PP12-1-V28-59-1
108,300,656	365-PP12-1-V28-59-1	208,302,946	5630	is information that is included in	108,300,955	Volume 42 Section 16 Part of Nitrogen generation units
108,300,955	Volume 42 Section 16 Part of Nitrogen generation units	208,302,947	1,726	is a qualification of	970,495	archive subsection
108,300,955	Volume 42 Section 16 Part of Nitrogen generation units	208,302,948	5630	is information that is included in	108,301,337	Section 16 Nitrogen Generation Units
108,301,337	Section 16 Nitrogen Generation Units	208,301,456	1,726	is a qualification of	970,493	archive section
108,301,337	Section 16 Nitrogen Generation Units	208,301,457	5630	is information that is included in	108,301,357	Mechanical Catalogue of Liquefaction Plant
108,301,357	Mechanical Catalogue of Liquefaction Plant	208,301,458	1,726	is a qualification of	970002	information

Note, the individual documents are arranged in collections. The relation types that are used to describe collections differ from relation types that are used to describe assemblies, because assemblies are about single objects, whereas collections are about plural objects. For example, to indicate that one collection is a subset of another collection we use the 'is a subset of' relation whereas for an assembly we use the 'is a part of' relation.

7.2 Relate documents to facility components

Specify for each facility component, such as equipment items, on which drawing it appears or which document contains information about it. This also holds for relevant activities and processes. An alternative specification of the same information is to specify the inverse relations that describe for each document about which facility component or activity it contains information.

For example:

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
108,200,000	D-326	208,300,892	5,046	is described in	108,300,046	LD1-4-C1-0100 Rev. 0

Note: When the document is succeeded by a next version (which strictly speaking is another document), then by default it is assumed that this object is (probably) also described on next versions of the document. This means that application software that support this rule should implement this "fact propagation" rule, including an indication that it is a conclusion by propagation.

7.3 Relate documents to files at addresses

For each document it shall be specified on which a physical medium it is presented, either as an electronic data file or on paper, microfilm or any other medium. A file name shall include a file extension.

In addition to that each physical file shall be classified as an electronic data file and each physical files shall have its own UID allocated. Finally a directory shall be defined. The directory shall be classified and for each file it shall be specified of which directory it is an element.

This will result in a table. For example:

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
108,300,046	LD1-4-C1-0100 Rev. 0	208,301,124	4,996	is presented on	108,300,308	BRPFD09.dwg
108,300,308	BRPFD09.dwg	208,302,224	1,225	is classified as a	490,533	electronic data file
108,300,308	BRPFD09.dwg	208,302,223	1,227	is an element of	108,300,567	C:\LNG\Files
108,300,567	C:\LNG\Files	208,302,225	1,225	is classified as a	492,017	directory

The resulting table may be verified using the Gellish Browser. This show whether the expressions are correct Gellish expressions.

7.3.1 Multi-file documents

Sometimes a document is physically presented on multiple files, which may or may not be collected in a zip file. Typically a document and its appendices are often presented in different files, whereas the file types of those appendices may also differ from the document and from each other. For example, a main document may be stored as a doc file, whereas one of the appendices may be stored as an xls file.

In such multi-file cases an object shall be created in the database that is a collection of files. For each physical file that forms part of the multi-file document it shall be specified that it is an element of that collection. If the files are physically collected in a zip file, then that collection has zip as file type. If the files are not physically collected in a zip file, then the

collection is not represented by a separate physical file, but nevertheless that collection is created as a recognized object in the database.

Either the directory of the separate files or the directory of the zip file shall be specified.

For example, if the collection is not arranged in a zip file, the table will become as follows:

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
108,300,900	MCC_C single line diagram	208,302,500	4,996	is presented on	108,300,956	MCC_C collection of files
108,300,956	MCC_C collection of files	208,302,501	1,225	is classified as a	108,300,956	MCC_C collection of files
108,300,957	MCC_C1.dwg	208,302,502	1,225	is classified as a	492,018	collection of electronic data files
108,300,957	MCC_C1.dwg	208,302,503	1,227	is an element of	108,300,956	MCC_C collection of files
108,300,957	MCC_C1.dwg	208,302,504	1,227	is an element of	108,300,567	C:\LNG\Files
108,300,958	MCC_C2.dwg	208,302,505	1,225	is classified as a	490,533	electronic data file
108,300,958	MCC_C2.dwg	208,302,506	1,227	is an element of	108,300,956	MCC_C collection of files
108,300,958	MCC_C2.dwg	208,302,507	1,227	is an element of	108,300,567	C:\LNG\Files

7.4 Decomposition of documents

Documents usually consists of monolithic pieces of text. This implies that information about such documents (meta-data) can only be related to those documents as a whole. Such documents may contain so called hyperlinks that consist of pointers that are included in the document and that refer to other documents or parts of the same document. Such a pointer operates only in one direction and its operation is dependent on the location and name of the referred document. This is a draw-back that often leads to broken links. Such hyperlinks are internal document content and therefore that are not managed explicitly in a Facility Information Model.

Another possibility is that documents are decomposed in separate pieces of text, on in a level of granularity that may vary. For example, a document may be decomposed in words and phrases or paragraphs and chapters, whereas the sequence in which the pieces of text in the documents appear is recorded in a database. This enables that the separate pieces of text can be related individually to elements of the Facility Information Model. For example, a paragraph may express a requirement for a particular component. An explicit relation between that component and that requirement is a very powerful means to manage requirements and to verify whether they are satisfied. The specification of such a decomposed document in a computer interpretable form is described below.

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	Full description
2	paragraph-2	101	1,726	is a qualification of a	970,007	requirement	The lubrication system shall have a capacity that ….
2	paragraph-2	102	5,398	is a specification for a	130,166	lubrication system for a compressor	
2	paragraph-2	103	5,519	is by definition a part of	3	API 617	
2	paragraph-2	104	5,332	is the next element after	1	paragraph-1	

8 Step 3, Specify data sets with additional Facility Model data

The previous chapters described how document files can be integrated in a Facility Information Model. Those documents are treated as 'black boxes', because they are expressed in natural languages or data structures that cannot be interpreted by the system(s) that implement the Facility Information Model.

However, it is also possible that files contain computer interpretable data ('data sets') that can be integrated *as data* in the Facility Information Model. This means that each individual data element in such a data set (for example the fields in a spreadsheet or in a database table) can be included in the database system(s) that implement the Facility Information Model. In other words, such files can be imported and interpreted *as data* in those systems.

The following paragraphs describe how (parts of) data sets can be created that are expressed in a system independent way, so that they can be directly imported without a need for data conversion.

8.1 Specify fabricated items and relate them to designed items

Installed physical items are real fabricated items, each of which typically has a name that is a combination of its manufacturer and its serial number. Fabricated items are distinguished from the tagged items, which are their imaginary counterparts. A fabricated item that is installed at a particular place may be replaced by another fabricated item. Thus, in the course of time, there may be more than one item installed at the place of a tagged item.

Each fabricated item shall be defined by classifying it by the 'manufacturer's model' (or 'model and size') of which it is an exemplar. The manufacturer's model itself is a type of thing that shall be added to the dictionary by specifying that it is a subtype of a more general equipment type, using the relation type <is a model of>. The manufacturer's model shall have a name that is a combination of the manufacturer's name and the model name and possibly its size. Fabricated items are added and integrated in the Facility Information Model via a specification of a relation with the tagged item at which place the real item is installed.

For example, a particular compressor with serial number Elliott E8368 is installed at the place of tagged item C-6002 A. That C-6002 A is classified elsewhere as a centrifugal compressor. The Elliott machine is an exemplar of a type that is a model of a centrifugal compressor. The

addition of this installed item to the model will result in a table with a number of lines that express the above kinds of facts as follows:

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
108,400,125	ELLIOTT E8368	208,400,501	1,313	is installed at the place of	108,200,575	C-6002 A
108,400,125	ELLIOTT E8368	208,400,125	1,225	is classified as a	108,400,056	ELLIOTT model 46MB-4
108,400,056	ELLIOTT model 46MB-4	208,400,056	5,396	is a model of	130,057	centrifugal compressor

Measured data will always relate to the fabricated items, because the designed counterparts (the tagged items) are imaginary things that cannot be measured.

8.2 Specify activities and processes (functions) of/in facility components

Processes are the occurrences that take place in or with the facilities and its components, such as the burning of fuel in a diesel engine, the production of clean water in a water purification facility, or the transport of people and goods via a railway, but also the control of those processes, the measurement of the properties and the performance of the facilities, etc. These processes are often called the functions of the facilities that perform the processes.

Activities are occurrences that are performed by people, such as fabrication, inspection, operation and maintenance of the facilities and its components.

Both kind of occurrences, processes and activities, can be decomposed in smaller occurrences, such as tasks, events, and the like. Their decomposition is described in the same way as the decomposition of a facility in components.

For example, the decomposition of a liquefaction in a compression and a cooling step are described as follows:

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
108,500,041	Compression in K-1301	208,500,040	1,190	is a part of	108,500,040	Liquefaction of S1
108,500,042	Cooling in E-1304	208,500,041	1,190	is a part of	108,500,040	Liquefaction of S1

Similarly the decomposition of a commissioning activity in sub-activities can be expressed as follows:

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
108,500,051	Cleaning of C-1201 System	208,500,051	1,190	is a part of	108,500,050	Commissioning of C-1201 System
108,500,052	Inspection of C-1201 System	208,500,052	1,190	is a part of	108,500,050	Commissioning of C-1201 System

The further decomposition into tasks is done in the same way.

Any occurrence can be described by its relations with the physical objects and/or people that are involved in their particular role in the occurrence.

For the description of physical and chemical processes there are additional objects involved, bing the input streams and output streams of process steps (unit operations). Furthermore, the same stream that is output of one process step is also input in the next process step. This is described by the input and output roles that the streams play in relation to the process steps. For example, the liquefaction process that was decomposed above into a compression and a cooling process step has the following flow diagram:

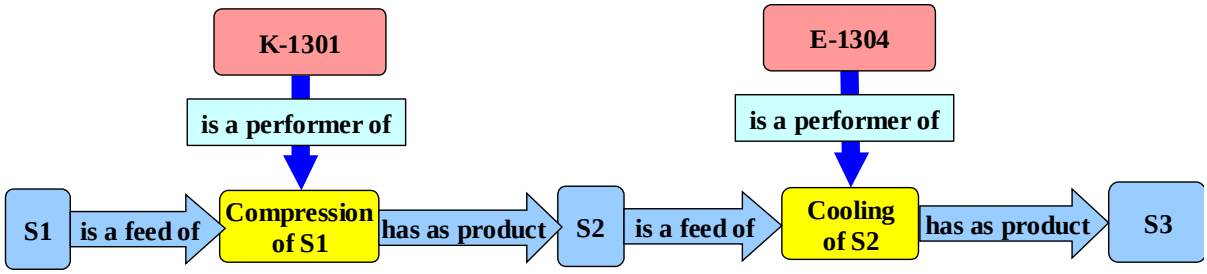


Figure 4, Process Flow Schema of gas liquefaction

The compressor K-1301 is performer of the Compression of S1 process step, which has S1 as input and S2 as output stream. That same S2 is input in the Cooling of S2 process step, that has E-1304 as performer and S3 as output stream. In other words: Compression of S1 is a function that is performed by compressor K-1301.

These process steps, streams and performers can be added to the Facility Information Model for the liquefaction facility as follows:

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
108,500,001	K-1301	208,500,001	4,761	is performer of	108,500,002	Compression of S1
108,500,002	Compression of S1	208,500,002	1,225	is classified as a	191,936	compression
108,500,003	S1	208,500,003	4,785	is an input in	108,500,002	Compression of S1
108,500,004	S2	208,500,004	4,786	is an output of	108,500,002	Compression of S1
108,500,003	S1	208,500,005	1,225	is classified as a	432,099	gas stream
108,500,004	S2	208,500,006	1,225	is classified as a	432,099	gas stream
108,500,005	E-1304	208,500,001	4,761	is performer of	108,500,006	Cooling of S2
108,500,006	Cooling of S2	208,500,002	1,225	is classified as a	191,804	cooling
108,500,004	S2	208,500,003	4,785	is an input in	108,500,006	Cooling of S2
108,500,007	S3	208,500,004	4,786	is an output of	108,500,006	Cooling of S2
108,500,007	S3	208,500,005	1,225	is classified as a	432,099	gas stream

An occurrence can be specified also by aspects of the occurrence, such as the fact that it begins at a particular moment in time at a particular location and that it has a certain duration (which may be negligibly short). The addition of such aspects is described in the next paragraph.

When activities are defined it might also be needed to specify which objects are subjected to the activities and in which sequence the activities need to be or are performed. This can be done by specifying relations an object and an activity and relations between the activities. For example, a decomposition of activities in tasks, or a sequence of activities, can be described by relations that specify that an activity is a predecessor of another activity, such that an activity can or may only start when the previous activity is completed. Such relations between activities and involved objects are specified as follows:

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
108,500,050	Commissioning of C-1201 System	208,500,051	4,760	has as subject	108,500,040	C-1201 System
108,500,051	Cleaning of C-1201 System	208,500,051	1,388	is a predecessor in time of	108,500,052	Inspection of C-1201 System

Note that the first line describes another kind of involvement in an activity than the input and output or performer kind of involvements. It is the inverse of the relation type <is a subject in>, so that the inverse expression would be:

C-1201 System <is a subject in> Commissioning of C-1201 System

Documents about the way in which this kind of activities should be executed as well as documents that describe the results of a specific activity can be added in the same way as described above for the relation between physical objects and documents.

8.3 Specify aspects of facility components and products

Aspects are characteristics, properties and qualities of assemblies and components or streams, activities and processes in a facility. The addition of aspects to the Facility Information Model can vary from simple aspects, to complicated descriptions of aspects, such as pressure and temperature dependent properties, behaviors and complicated shapes. This part describes the addition of the simpler aspects of physical objects only. The addition of more complicated aspects is described in part 4A of the Gellish Modeling Method (see also the Gellish Application Handbook).

Any physical object in a facility may have various aspects. This holds for example for whole facilities, complete assemblies, their components, the streams and the documents. Aspects can have qualitative values and quantitative values. The qualitative values are values that are usually selected from a list of allowed values, such as provided by a pick-list in a system. For example, a colour is an aspect which values may be selected from a list of allowed values, such as red, green and blue. Those qualitative aspects should be selected from the Gellish Dictionary or should be added to it if they do not exist yet. The quantitative values are usually quantified by a number on a scale. For example, an operating temperature is an aspect which values can be quantified on a Celsius scale or on a Fahrenheit scale. The numbers and the units of measure should also be selected from the dictionary (and maybe they need to be added to it).

The specification of aspects consists of two stages: the specification that an individual object has an individual aspect and the classification and quantification of the aspect.

For example, vessel V-6060 has an inside diameter of 2500 mm and is painted in a green colour. This is specified as follows:

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	UID of Unit of Measure	Name of Unit of Measure
108,400,601	V-6060	208,400,601	1,313	has aspect	108,400,701	diameter of V-6060		
108,400,701	diameter of V-6060	208,400,602	1,225	is classified as a	550,134	inside diameter		
108,400,701	diameter of V-6060	208,400,603	5,025	has on scale a value equal to	922,305	2500	570,423	mm
108,400,601	V-6060	208,400,604	1,313	has aspect	108,400,702	colour of V-6060		
108,400,702	colour of V-6060	208,400,605	5,020	is qualified as	551,666	green		

For the specification of a material of construction there is a special relation type that enables to specify the material on one line, instead of two as follows:

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	UID of Unit of Measure	Name of Unit of Measure
108,400,601	V-6060	208,400,606	5,423	is made of	280,192	stainless steel		

The discussion of aspects of processes, streams, inspection and maintenance activities is discussed in part 4A.

9 Unique identifiers

In order to avoid overlap between the objects in facilities and the Gellish Dictionary as well as within the site/project it is important to assure that each object is identified by a Unique Identifier (UID). Therefore, the allocation of Unique Identifiers for the facilities shall be managed.

For example, the following Unique Identifiers (UID's) may be reserved for the information about a particular facility:

Object UID's 108000000 - 109000000

Fact UID's 208000000 – 209000000

Within those ranges for example the following subsets may be defined:

	Object UID's	Fact UID's
Equipment items	108000000 – 108200000	208000000 – 208200000
Documents	108300000 - 108400000	208300000 - 208400000

10 Storing Data in a Gellish Universal Database

The facts that compose the Facility Information Model can be combined in one or more Gellish Data Tables. Such tables are self-contained, because they can be interpreted independently. Therefore they can be sent as Gellish Exchange Messages to other parties. A party that wants to store such a data table in a database can use the Gellish Universal Database API function Add_Facts (see Ref. 3) to verify the consistency of the facts with facts that are already in the database, to report inconsistencies and to store and integrate the facts in the database.

11 Knowledge-aided design (Propose_Facts)

The above process to create a Facility Information Model assumes that the user knows which facts about an individual object or activity should be created. However Gellish also enables that this process is supported by proposals for facts, which proposals are generated from knowledge that is stored in a Gellish Database. Note that proposals can be generated for individual things of any kind, including physical objects as well as activities and processes.

This chapter describes how such proposals can created by the software. The method to model and express knowledge in Gellish is described in part 3 of the Gellish Modeling Method.

Facts that express knowledge and requirements (and definitions) about *kinds* of things can be interpreted as potential facts about individual things. For example, the fact that expresses the knowledge that a 'car' can have as part an 'air bag' means that there are potential facts about individual cars such as: 'car-123' has as part 'air bag-123'. The availability of such modelled knowledge implies that, when an individual thing is classified by a kind of thing, the software can determine potential facts about the individual thing. This software functionality is supported by the Gellish Universal Database API (Application Programming Interface) through a function called Propose_Facts. The function is defined as follows:

Function: Propose_Facts (Individual_thing, Kind_of_thing, Query, Proposed_Facts)

Input:

- Individual_thing The individual thing about which new facts should be proposed.
- Kind_of_thing The kind of thing by which the individual thing is classified.
- Query The (standard) knowledge or requirements query that generates facts from which proposals are to be derived.

Results:

- Proposed_Facts Gellish facts that are proposed on the basis of knowledge and/or requirements (array).

The process to create proposals from knowledge is illustrated by Figure 5.

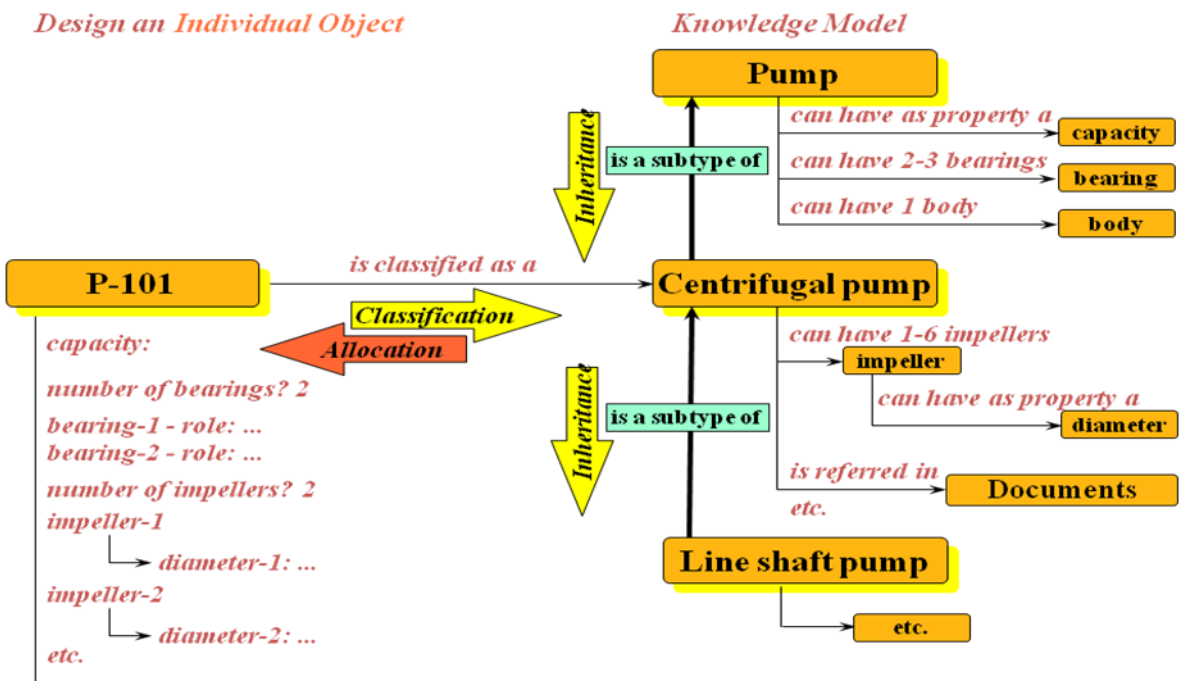


Figure 5, Example of allocation of knowledge to an individual thing

The right hand side of Figure 5 illustrates knowledge about pumps. For example, it states that a centrifugal pump is a subtype of a pump. Therefore, knowledge facts about a pump are inherited by the concept centrifugal pump. Additional knowledge facts are added that are specifically valid for centrifugal pumps, such as the fact that such a pump can have 1-6 impellers with their properties. When an individual object, such as P-101 is classified as a centrifugal pump (is related to that concept by a classification relation), then the knowledge facts about that concept can be used to generate proposals for individual facts about P-101.

With another example we will illustrate how the facts are generated and expressed in Gellish. Assume that a standard query for knowledge about 'sliding door' results in the following facts that express (partly inherited) knowledge and requirements:

door	shall have as aspect a		width
door	can have as aspect a		material of construction (MOC)
sliding door	can have as part a	2, 4	wheel
sliding door	is a specialization of		door

Furthermore, assume that from the arguments in the Propose_Facts function it follows that D-101 is classified as a sliding door. Thus the first proposed relation is the following classification relation that follows from the arguments:

D-101	is classified as a	sliding door

Other proposals for facts can be generated on the basis of the above facts that specify knowledge and requirements about a door, in combination with facts in the TOPextensions section of the Gellish Dictionary. For many relation types that can be used to express knowledge about a kind of thing, the TOPextensions section states which relation type should be used to express a fact about an individual thing of such a kind. For example, the TOPextensions section contains the following expressions:

can have as aspect a	can be realized by a	has as aspect
can have as part a	can be realized by a	has as part

In other words: a <has as aspect> relation type is a realization of a <can have as aspect> relation type.

If for a relation type there is no counterparty specified in the TOPextensions section, then usually there is a counterparty specified for its supertype or higher level supertype.

Thus from the above knowledge and requirements the following proposals for facts about D-101 can be generated:

D-101	is classified as a	sliding door	
D-101	has as aspect	width of D-101	
width of D-101	is classified as a	width	
width of D-101	has on scale a value equal to	?	mm
D-101	has as aspect	MOC of D-101	
MOC of D-101	is classified as a	material of construction	
MOC of D-101	is qualified as	?	
D-101	has as part	wheel of D-101	
wheel of D-101	is classified as a	wheel	

Proposals can be further enhanced in various ways:

- Proposals should also deliver the possible subtypes of the counterpart relation types as alternative proposals. For example, the fourth relation type might also be <has on scale a value greater than>.
- Proposals may present options for the values (indicated by question marks (?)). How options are specified is discussed later.
- The options for unit of measure may be derived from the fact that width is a subtype of distance for which a 'length scale' is specified that is qualified by a list of scales. Furthermore a default unit of measure may be derived from a fact about the preferred scale for a unit (in a particular context).

It should be verified whether potential new objects and new facts exist already and whether they are in conflict with other existing facts (for example conflicting classifications). When they do not exist yet, then they should get proposed new UID's. For that reason it is required that ranges are specified from which the new UID's should be selected after verification of their non-existence.

The user of an application should have the option to accept, reject or modify the proposals.

12 Verification against requirements (Verify_Facts)

Once a model of an individual object or activity is created, Gellish enables to verify the content of the model against facts that express requirements for such a kind of thing. The verification (of the model) of an individual object against requirements is supported by a function that is included in the Gellish Universal Database API, called Verify_Facts. The functionality of that function includes that a collection of facts about an individual thing is verified on completeness and consistency with a number of facts that express the requirements. The function is defined as follows:

Function: Verify_Facts (Individual_thing, Kind_of_thing, Facts_to_be_Verified, Query, Verified_Facts)

Input:

- Individual_thing The individual thing (UID) about which facts are to be verified.
- Kind_of_thing The kind of thing (UID) by which the individual thing is classified.
- Facts_to_be_Verified The Gellish Message (Array or URL) with facts about the individual thing that needs to be verified.
- Query The (standard) knowledge or requirements query that generates facts against which facts are to be verified.

Results:

- Verified_Facts Gellish facts that are verified on the basis of knowledge and/or requirements (array). This indicates:
  - which facts satisfy which requirements
  - which facts are present without a requirement,
  - which facts are in conflict with a requirement,
  - which facts are missing.

Figure 6, Example of verification of facts against requirements illustrates an example of a verification.

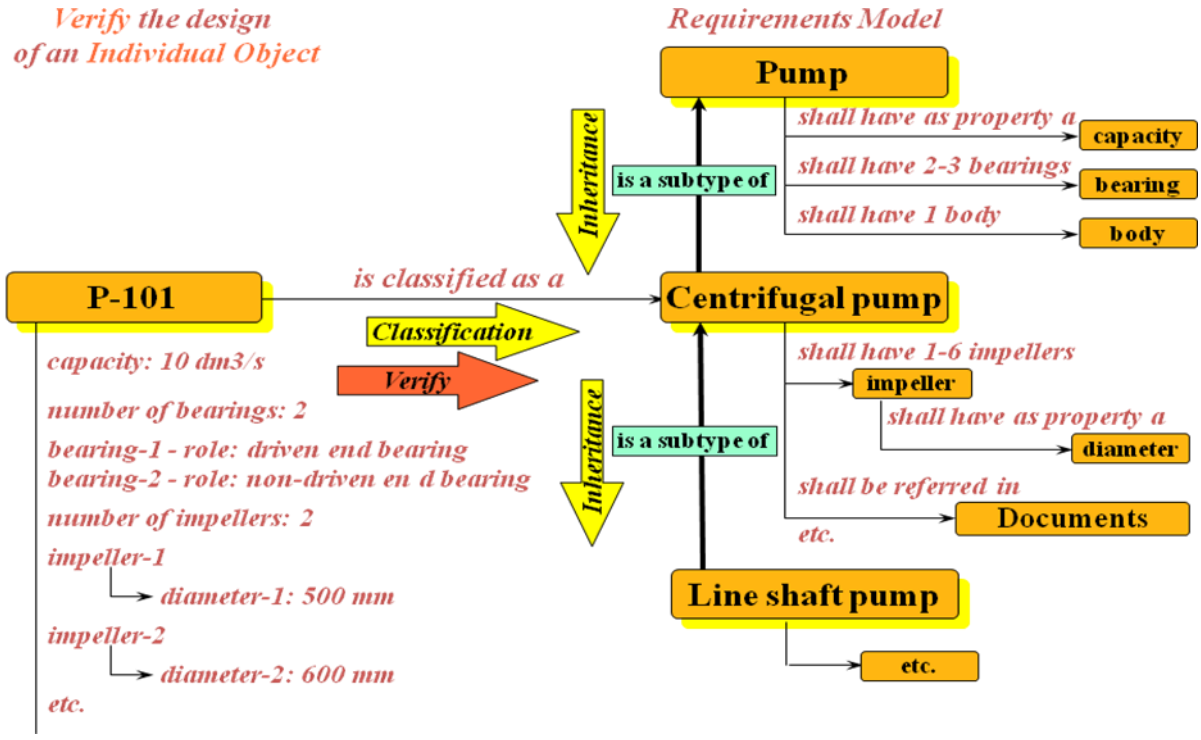


Figure 6, Example of verification of facts against requirements

The left hand side of Figure 6, Example of verification of facts against requirements illustrates facts about P-101, including also the fact that P-101 is classified as a centrifugal pump. The right hand side of the figure illustrates requirements that are expressed about a centrifugal pump and include requirements that are inherited from its supertype 'pump'. The verification

includes a check whether each requirement is reflected in a fact about the individual P-101 and whether the constraints are satisfied.

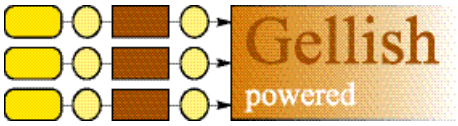
The verification process uses the same relations between relation types that express requirements about kinds of things and relation types that express facts about individual things as are used for the creation of proposals and that are documented in the TOPexpresions section of the Gellish Dictionary (see the previous chapter).

The application software should generate a report about inconsistencies and omissions.

13 "Gellish Powered" certification criteria

The "Gellish Powered" certificate can be obtained for software that is able to import and/or export and operate on data sets with Gellish compliant Facility Information Models in a way that is compliant with the Gellish Modeling Method. The "Gellish Powered" certificate can also be obtained for Facility Information Models themselves. The Gellish@work organization can perform the verification and has the right to issue such certificates. The certificate is protected as Gellish is a Registered Trademark.

Gellish compliant software or data sets can be recognized by the following Gellish logo 'Gellish powered' in the following layout:



This chapter describes the certification criteria for a Gellish Facility Information Model and for software that operates on it.

13.1 What is a correct Gellish Facility Information Model

Correct Gellish is a collection of Gellish expressions of facts (main facts with auxiliary facts) that are presented in one or more Gellish Data Tables or a representation thereof and that are compliant with the following rules.

Note that only those rules are mentioned here that are directly relevant for the verification of a Facility Information Model.

Rule 3: Explicit classification.

Each individual thing shall be classified by at least one explicit classification relation with a kind of thing (concept or qualitative aspect), whereas such a kind of thing shall be:

- Either selected from the Gellish dictionary,

or

- Properly defined as a subtype of a concept in the Gellish dictionary (possibly as an element of a domain dictionary),

whereas each individual relation is classified by a kind of relation (relation type) that is also selected from the Gellish dictionary.

How to verify rule 3:

1. Verify for each used individual thing whether it is classified by a classification relation (or a subtype of it) with a concept (UID) that is a subtype of the concept "anything" through a sequence of specialization relations (or through subtypes of

specialization relations) with more general concepts, which sequence, when following the hierarchy, terminates at the concept "anything". If the top concept is not "anything", then an error shall be generated including a report about the concepts that actually terminates the sequence.

This can be verified using the built-in verification rules of the Gellish Browser in combination with the Gellish Dictionary.

Rule 4: Grammatical correctness.

Each relations between things is classified by a kind of relation (relation type) as specified in column 60 and 3 of a Gellish Data Table. Such a kind of relation defines the kinds of roles that are played by the related things in a relation of such a kind. Each role player shall be able to play the role that it is required to play according to the kind of relation. The (required or specified) nature of a role player shall not be in conflict with the nature required by other relation types in which the role player plays a role.

The Gellish Dictionary (in TOPini) includes "facts about conceptual relations" that define the relation types of the Gellish language. These definitions also specify the required kinds of roles and the kinds of things that can play those kinds of roles (which implies that also their subtypes can play those roles).

Example 1: the Gellish expression C is a specialization of D is only correct if C is a concept, because the <is a specialization of> relation requires as first role a subtype, wheras a subtype role can only be played by a concept (or by a subtype of concept). The relation type requires as second role a supertype, which can also only be played by a concept. So, D must also be a concept.

Example 2: the Gellish expression "A is by definition a performer of a B" is only correct if A is a subtype of physical object, because the facts in the Gellish Dictionary specify that an <is by definition a performer of a> relation requires as first role a performer and it also specifies that a performer role can only be played by (a subtype of) physical object. The relation type requires as second role a "performed", which is a role that can only be played by (a subtype of) occurrence. So B must be (a subtype of) occurrence. These conclusions about the nature of A and B shall not be in conflict with the nature required by other relation types in which A or B plays a role.

How to verify rule 4:

1. Verify for each used relation type whether the relation type exists in the Gellish Dictionary (TOPini part). Proprietary extensions of relation types shall be verified separately.

   This can be verified using the built-in verification rules of the Gellish Browser in combination with the Gellish Dictionary.

2. Verify the consistency of the object type of the left hand and of the right hand object according to various relations in which those objects are involved as follows: When something is related to another thing by a relation of a particular kind, then it can be derived from the definition of the relation type in the Gellish Dictionary what kind of thing it must be. This can be determined as follows: according to the Gellish Dictionary (TOPini part) the relation type requires a first kind of role and a second kind of role. This is either directly specified for the relation type or it is inherited from its supertype relation type. The Gellish Dictionary also specifies which kind of thing can play such a kind of role. This can also be specified either directly or by inheritance from its supertype kind of role. When the object is a left hand object and the Gellish phrase for the relation type is a normal phrase (or both the opposite), then the thing shall be of the kind of thing that can

play the first kind of role. When it is a right hand object or an inverse phrase then it shall be of the kind of thing that can play the second kind of role, etc.

Conclusions about the kind of thing that are derived from various relations for something and its explicit classification relations or specialization relations (or their subtypes) shall not be in conflict with each other. There is no conflict if the conclusion about the kind of thing drawn from the classification or specialization relation(s) is the same or a subtype of the kind that is required by a relation type.

This should be verified by the built-in rules in the Gellish Browser.

If the specialization hierarchy of concepts is incomplete (rule 1 is not satisfied, see part 2 about the Dictionary), then this verification is impossible.

Rule 5: Semantic consistency.

The relation types that classify the relations in Gellish expressions determine the nature of the things that play the roles in the relation. Those natures shall be consistent with the natures that are required by other relations in which the same thing plays a role.

Example 1: from the Gellish expression C is a specialization of D it can be concluded that C is a concept. If there is another Gellish expression that states that C is a part of E, then from that relation it can be concluded that C is an individual thing. These two expressions are thus inconsistent as C cannot be a concept and at the same time be an individual thing.

Example 2: from the Gellish expression A is a performer of B it can be concluded that B is an occurrence. If there exists another Gellish expression that states that B is a performer of F then the two expressions are inconsistent, because B cannot be an occurrence as well as a physical object.

How to verify rule 5:

There is no conflict between a conclusion about the kind of thing drawn from one relation type and a conclusion drawing from another relation type for the same thing, when the required kind of thing is the same or is a subtype or supertype of the kind that is required by the other relation type. In other words they are in conflict when the conclusions about the kinds of things belong to different branches of the specialization hierarchy.

This should be verified by the built-in rules in the Gellish Browser.

Rule 8: Completeness of auxiliary facts.

A Gellish Data Table shall comply with the column definitions according to one of the subsets that are defined in the "Gellish Database Definition" document. Each field on each row in such a table shall have values that comply with the rules defined in that document.

How to verify rule 8:

The Gellish Browser shall verify whether a value in a field is present when that is obligatory and whether the value satisfies the format constraints for the field.

13.2 Requirements for Application Software

Gellish Powered software shall comply with the following criteria:

1. It shall correctly interpret Gellish expressions. Correct Gellish expressions are expressions that comply with the rules that are described in the “Guidelines for the use of Gellish English”.
2. It shall comply with definition of a Gellish Database as is specified in the document “The Gellish Database Definition” https://sourceforge.net/project/showfiles.php?group_id=28353&package_id=235010.
3. It shall have functionality as is described in this chapter.

13.2.1 Viewing document files

It shall be possible to display the content of any file that is classified as an 'electronic data file' (UID 490533) or a subtype of it, when appropriate viewer software is available, by launching the application software. Such launching of applications is based on file types and paths to directories (URI’s).

The software that shall launched to view the file can de determined in two ways:

- From a direct relation between the file and the application software (type).
- Via the determination of the file format and recorded relations that specify which file formats can be read by which software.

The process to determine the file and the application is illustrated by the content of Table 1.

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
101	BR3007.dwg	201	1,225	is classified as a	490,533	electronic data file
101	BR3007.dwg	202	1,227	is an element of	102	E:\Projects\Document files
102	E:\Projects\Document files	203	1,225	is classified as a	492,017	directory

Table 1, Specification of a file in a directory

Table 1 contains an example of Gellish expressions about file BR3007.dwg that may occur in a Gellish Database.

First it should be determined in which directory the file is located. This can be done by determining the collection of which the file <is an element of> whereas that collection shall be classified as a’ directory (UID 492017). The name of that directory is (should be) a URL.

The file format of the electronic data file can be determined in two ways:

- From the file extension.
  This means that the name of the file ends with a dot (‘.’) followed by a file extension. Examples of the qualitative file extensions in the Gellish Dictionary are: doc, xls, dwg, dxf, htm, vsd, pdf, tif, jpg, ppt, txt.
- From the qualification of the file format.
  This means that the electronic data file <has as aspect> a file format that is qualified by one of the file formats (UID 911853) or of a subtype of that (such as MIME type). Examples of file formats are the MIME types, such as 'text/plain charset=us-ascii', application/vnd.ms-excel.

A file format is expressed in Gellish as follows:

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
102	E:\Projects\Document files	204	1,727	has as aspect	103	format-1
103	format-1	205	5,020	is qualified as	492,017	text/plain charset=us-ascii

Table 2, Specification of file formats

13.2.2 Searching for individual things

When a query searches for individual things that are classified by a particular class, then the software shall find the individual things that are directly classified by that class as well as the ones that are classified by a subtype of that class. Note that this differs from the inheritance principle.
This is illustrated by Table 3, which is an example where two compressors are differently classified.

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
104	K-1301	206	1,225	is classified as a	130,057	centrifugal compressor
105	K-3001	207	1,225	is classified as a	130,019	compressor

Table 3, Classification by subtypes

When a query searches in Table 3 for individual things that are classified as a compressor, then the software should not only find K-3001, which is directly classified as a compressor, but it should also find K-1301 which is classified as a centrifugal compressor, because that is a subtype of compressor according to the Gellish Dictionary.

13.2.3 Independency of sequence of expressions

The results of the interpretation of the content of a Gellish Database shall be independent of the sequence in which the lines are present in the table(s). For example, a new concept is defined by a <is a specialization of> relation, whereas on that line a textual description of the defined concept may be present. However, it is possible that before that line the defined concept is already used, for example in a <is a synonym of> relation that precedes the line where the concept is defined. Nevertheless, the software should ensure that the concept has the textual definition that is given with the later specialization relation.

13.2.4 Display of remarks on facts

The remarks field on a line in a Gellish Database contains remarks that are applicable for the fact on that line, and are not applicable for the left hand object or the right hand object. Therefore those remarks should only be displayed where it is intended to be present that fact to a user. For example, a remark on a line with a specialization relation is applicable for the definition of the left hand concept and should not be displayed at every occasion where that concept is used.

13.2.5 Exclusion of strings in searches

It is optional to include a possibility to specify a string that should not appear in the result string(s). This option is a means to reduce the number of results for a search, because a search for objects with names that include a particular string of characters may sometimes deliver an

unwanted large number of results. It may be that the number of results can only be reduced when an additional string of characters is specified (in a separate field) that may not occur in the results.

14 Application system usage example

This chapter describes how a Gellish search engine can be used to search, retrieve and inspect documents and data in a Facility Information Model.

14.1 Preparation of the application system

14.1.1 Search engine installation

This example application uses the Gellish Browser as an example of a Gellish Search Engine.

The Gellish Browser must be installed in a directory to which the user has write access.

The database, typically called 'Facility Information Model.CLB', can be installed in the same directory or anywhere else.

14.1.2 Loading the database

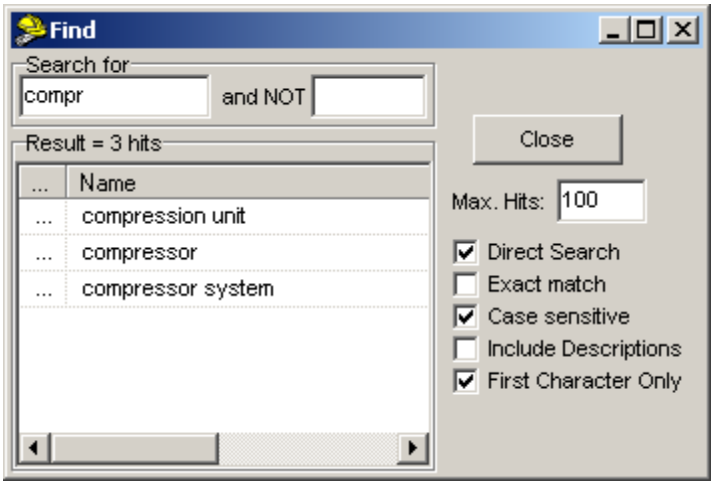
After the Gellish Search Engine is started the following action causes that the database is loaded:

- Select: File, Open, Native, New
- Browse to find the database.
- Select the file and Open the file (database)

14.2 Search for documents

1. Select: Options, Find (or Ctrl + F)

   This opens the following Find window (search window):



2. Mark the options.

   It is recommended to play with the options to familiarize with their power.

- Direct Search: Yes, recommended in nearly all cases. This means that after typing of any search character the search is immediately executed. No, means: first type search string, on "enter" the search is executed.
- Exact match: No, recommended in nearly all cases. Yes means that the result string shall exactly match the search string. Yes is only useful in exceptional cases when many strings satisfy the search string.
- Case sensitive: No, recommended. Use only YES when you are sure about use of upper and lower case. For example, names of kinds of things in the Dictionary are in lower case.
- Include Descriptions: Yes, recommended for search in document titles, because those titles are treated as descriptions. No, if you search for tag names or drawing numbers only.
- First Character Only: Yes/No depending on kind of search. If Yes, then the first character of the result string shall be equal to the first character of the search string.

3. Enter a search string:

   There are three ways to find a document:

   a) Search for a *tag name* and navigate through the plant model to find documents about the object.

   b) Search for a text string(s) in the *document title* or *document number* (drawing number) which can also be a binder name, section or sub-section.

   c) Search for a *kind* of object or *kind* of document and determine the required tag name or document that is classified by the kind.

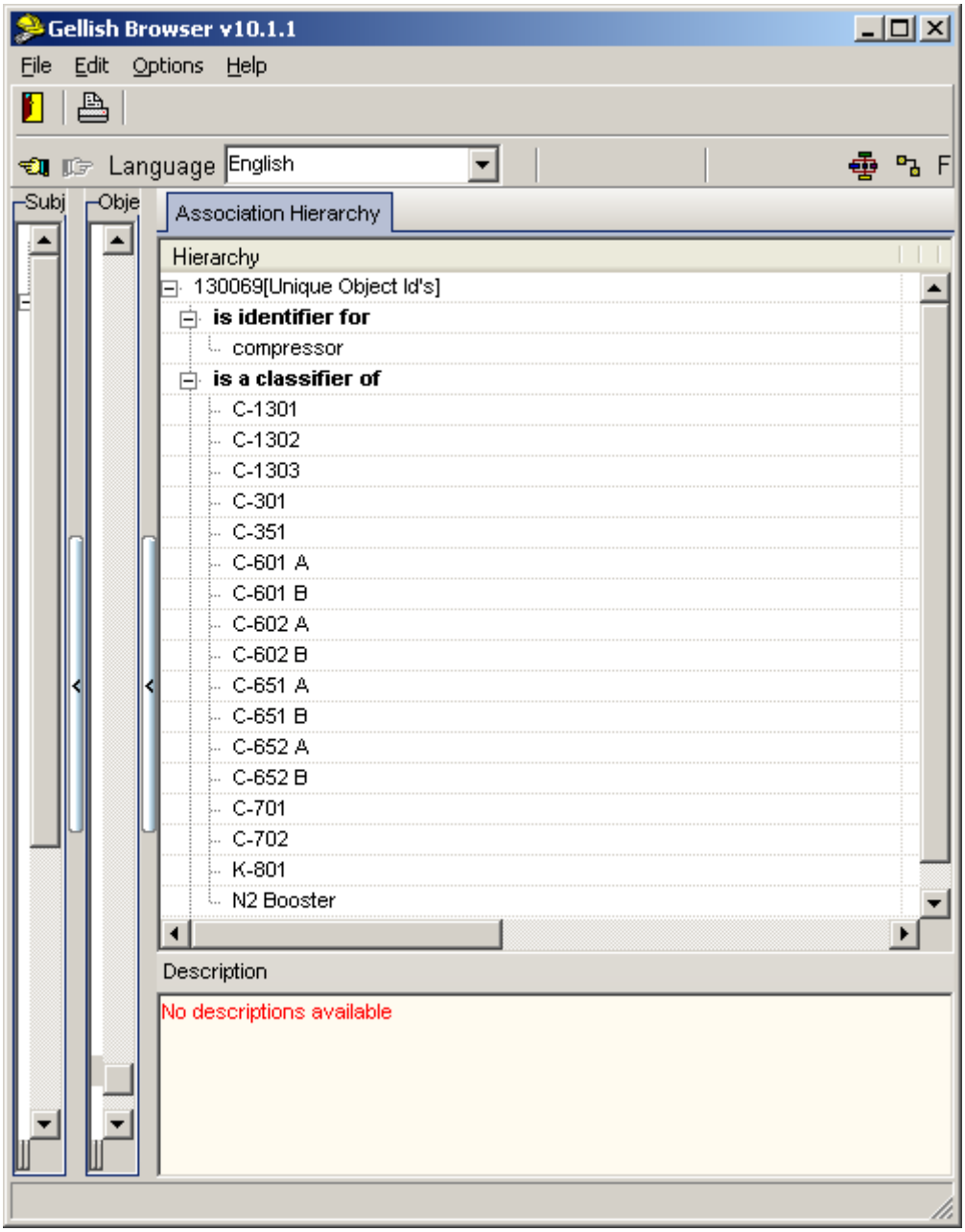
   For all three ways of searching the same single search field is used. There is no need to specify the way of searching.

   A fast way of searching is by entering one or more strings separated by spaces, where each string is only a part of a word. Usually partial strings will be sufficient to find the concept or document. For example:

   - "cen com" will usually be sufficient to find "centrifugal compressor" (especially when options "Case Sensitive", and "First Character Only" are marked)
   - "TK-12" will usually be sufficient to find TK-1201 (and other tanks) and all documents about tank TK-1201 (when option "Include Descriptions" is also marked).
   - "-1201" without marking the option "First Character Only" will display not only TK-1201, but also C-1201, CM-1201, FIL-1201 A, etc.
   - Entering a string in the "and NOT" field will display only result strings that do not include the excluded string.

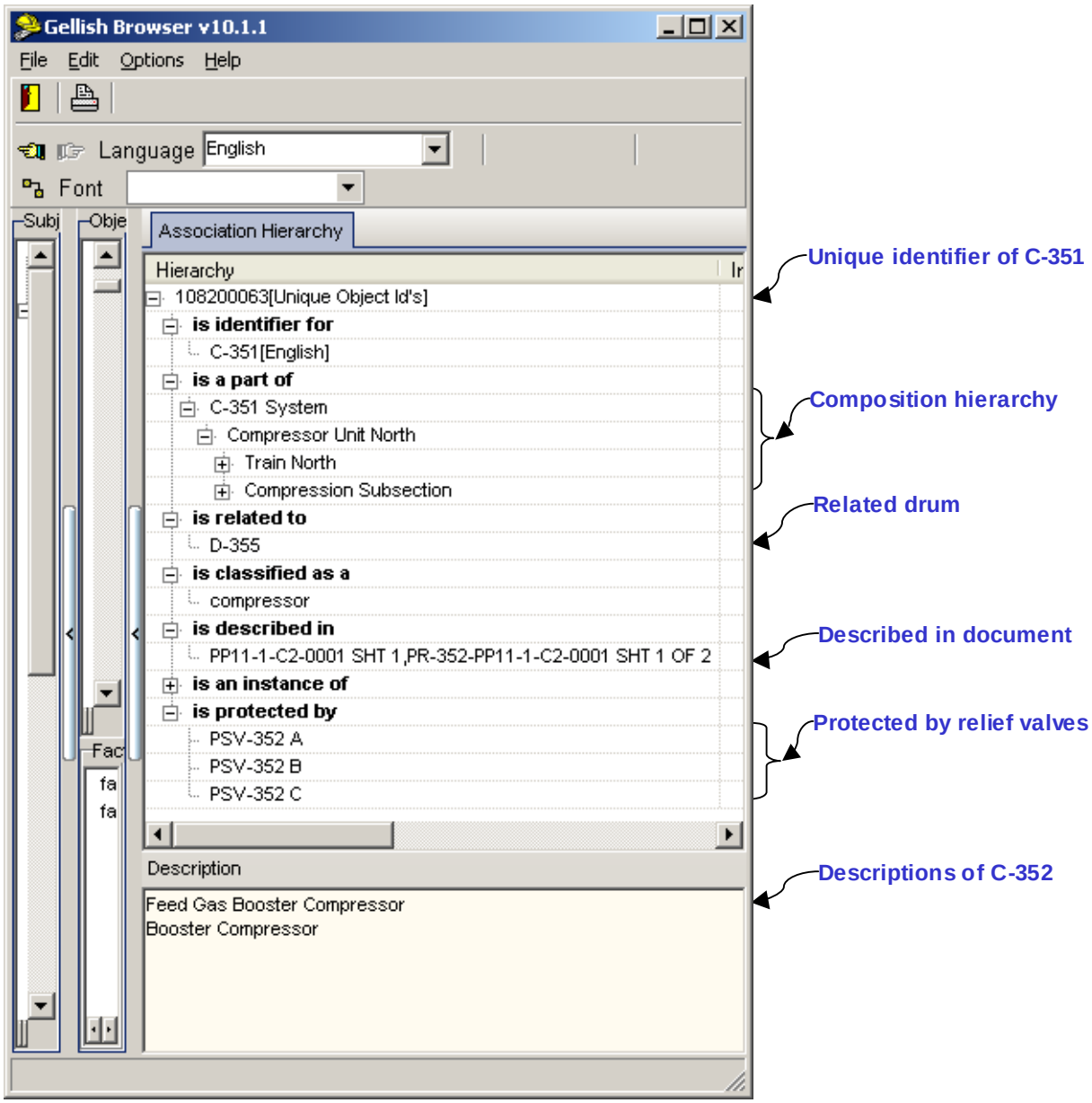
4. Select one of the results in the Find window (by double clicking). As a result of that the “object in focus” will be displayed in the window “Association Hierarchy”, together with relations to its “related objects”.

   For example, the result of a double click on compressor will put the concept “compressor” in focus and will also display a list of objects that are classified as compressor or as a subtype of compressor as well as other facts about compressors. The list of compressors is presented as follows:



5. Navigate through the network of related objects by double clicking on any object of interest.

   For example, double click on C-351 will put compressor C-351 in focus, while presenting information about it as follows:

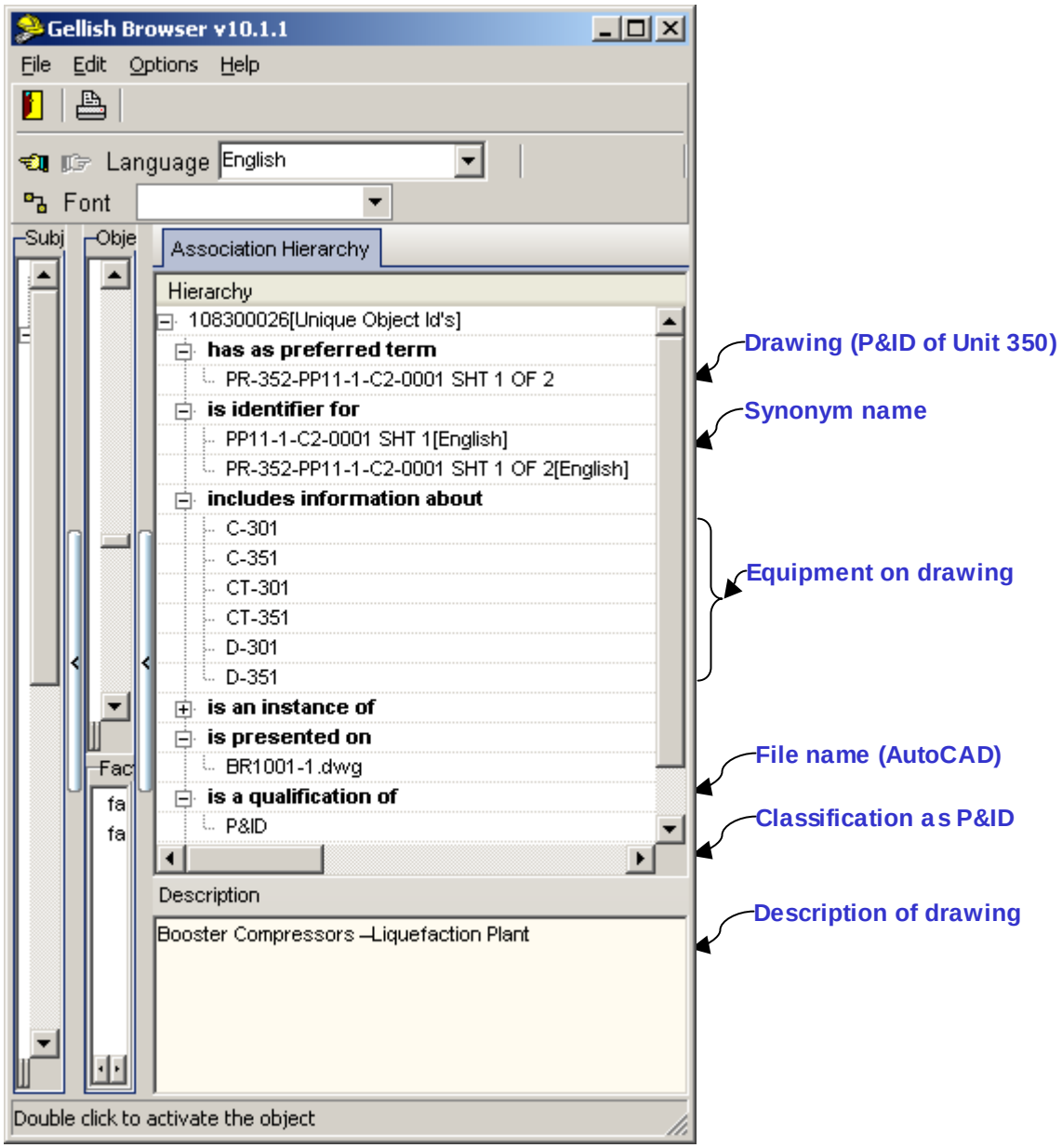


   The above information includes the fact that C-351 is described in document PP11-…

   Such a document can be selected by double clicking on its name.

6. Select a document as the object in focus.

   For example, double click on PP11-… This will result in the following information.



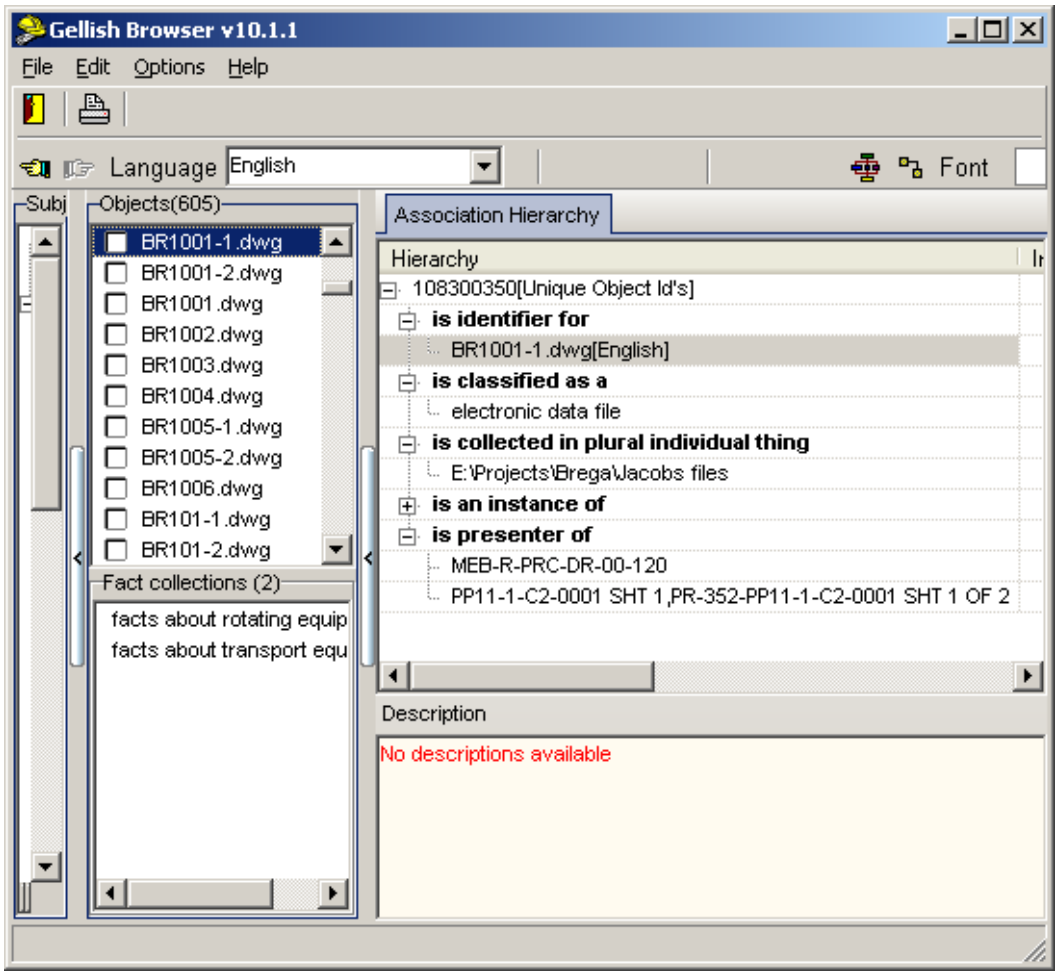
   This display illustrates that this document contains information about six pieces of equipment. The document could have been found via any of them.

   Typically the document is related to one or more files by a <is presented on> relation. In the example the document PP11-… is presented on file BR1001-1.dwg.

7. Select a file by double clicking on the file name.

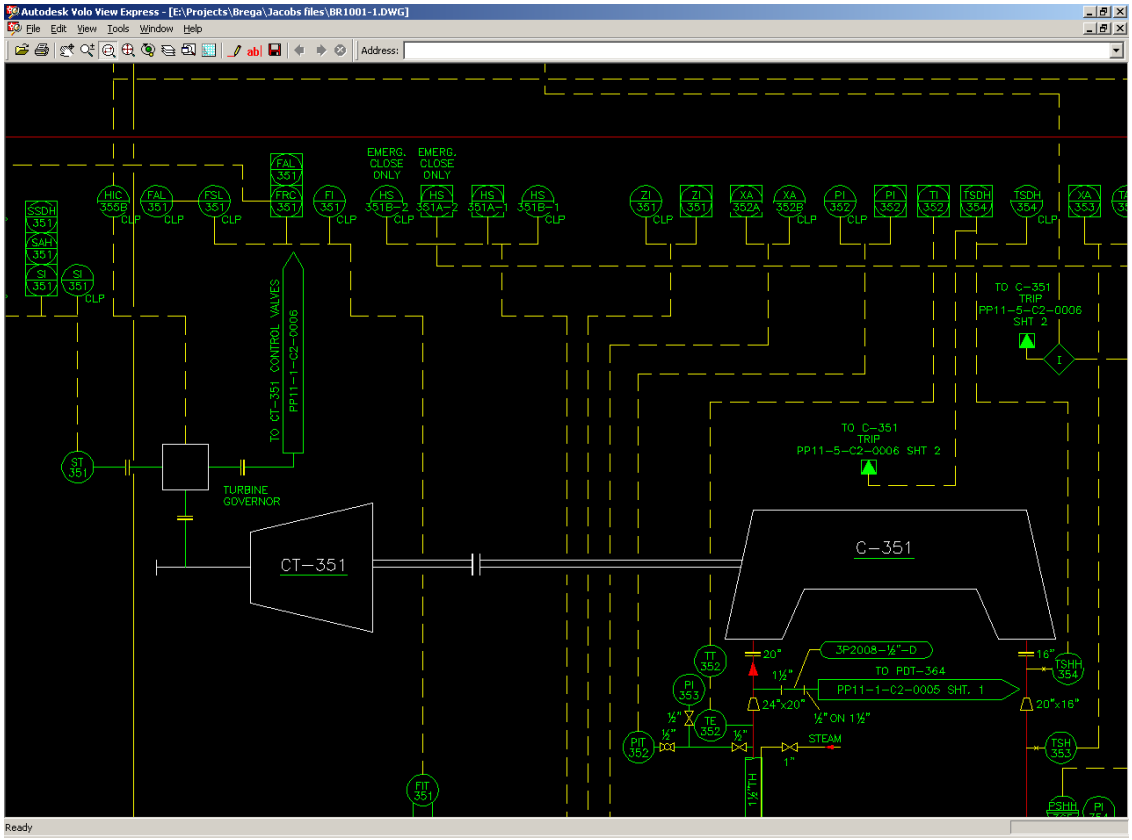
   For example select file BR1001-1.dwg.

   When the file is the object in focus, in the window “Objects” the file name will be highlighted in blue and underscored.



For example BR1001-1.dwg

8. Double click on the blue, underscored file name. This will launch a Viewer/Browser to view the document as follows.



15 References

1. Definition of Gellish Universal Databases and Data Exchange Messages, http://www.gellish.net/downloads/cat_view/2-gellish-files-english.html.
2. Gellish Modeling Method – Part 4B - Document Titles and Document Identification.
3. Gellish Universal Database API, http://www.gellish.net/downloads/cat_view/2-gellish-files-english.html.

16 Appendix A, Summary of used relation types

The following relation types are used to create a core Facility Information Model as described in this document. Additional relation types will be needed to add other kinds of information in the form of data or when the content of documents is modeled.

Note that each relation type has an inverse expression. When the inverse expression is used the left hand object and right hand object shall be exchanged.

Relation type UID	Relation type name	Description
1,225	is classified as a	is a relation that relates an individual thing to a kind of thing. Typically the left hand object is an individual physical object in the facility or an individual document and the right hand object is a kind of thing (a class) selected from the Gellish English Dictionary.
1,190	is a part of	is a relation that relates two individual objects, that specifies that the left hand object is a part of the right hand object.
4,856	is a subset of	is a relation that relates a collection to a bigger collection, that specifies that all the elements of the left hand collection are also elements of the right hand collection.
1,227	is an element of	is a relation that relates an individual thing to a collection that specifies that the left hand individual thing is an element in the right hand collection.
1,393	is a next version of	is a relation between two individual objects, that specifies that the left hand object is a next version of the right hand object. Typically the right hand object is replaced by the left hand object.
5,046	is described in	is a relation between an individual object and a document (information) that specifies that the left hand individual object is described in the right hand document.
4,996	is presented on	is a relation between a document (information) and a physical information carrier, such as an electronic file, that specifies that the left hand document is presented on the right hand information carrier.
1,313	is installed at the place of	is a relation between a real installed item and an imaginary item that specifies that the left hand real item is installed at the location of the right hand imaginary item. Typically the real item is indicated by its asset registration number (serial number) and the imaginary item is indicated by a tag name.
5,396	is a model of	is a relation between a kind of thing that is specified in detail and a kind of thing that is less specified that indicates that the left hand (manufacturer's) model is a specified model that is a specialization of the more general right hand kind of thing.
4,761	is performer of	is a relation between an individual physical object and an occurrence that specifies that the physical object makes the occurrence to happen.
4,785	is an input in	is a relation between an individual physical object and an occurrence that specifies that the physical object is consumed or used by the occurrence. Typically a material, energy or signal.
4,786	is an output of	is a relation between an individual physical object and an occurrence that specifies that the physical object is produced by or result of the occurrence. Typically a material, energy or signal.
4,760	is subject in	is a relation between an individual physical object and an occurrence that specifies that the physical object as subjected to the occurrence. (inverse: <has as subject>)
1,388	is a predecessor in time of	is a relation between two occurrences that specifies that the left hand occurrence has be finished before the right hand occurrence may start. (inverse: <is a successor in time of>)
1,313	has aspect	is a relation between an individual thing and an individual aspect that specifies that the individual thing possesses the aspect.
5,025	has on scale a value equal to	is a relation between an individual aspect and a number that specifies that the individual aspect has a magnitude that is quantified by the number on a scale, whereas the scale is separately specified in an additional column.
5,020	is qualified as	is a relation between an individual aspect and a qualitative aspect that specifies that the individual aspect is qualified by the qualitative aspect.
5,423	is made of	is a relation between a kind of thing and a kind of substance that indicates that the kind of thing is made of material which substance aspect is of that kind of substance. For example, SS bolt <is made of> stainless steel.

17 Appendix B, Glossary

AIM

Asset Information Management, being a synonym of Engineering Information Management. This includes the management of technical data and documents about a facility and its operation during its complete lifecycle.

Data set

A data set is a collection of data values (character strings or numbers) that represent *facts* (expressions of "what is the case") and that is arranged in a formal structure. Typically arranged in a tabular structure. Ideally expressed in a

formal language. For example arranged in one or more database or Excel spreadsheet tables structure.

Document

A document is information in a textual and/or graphical (or even audible) form that contains representations of *facts*. Usually, but not necessarily, in natural language form.

Electronic smart dictionary

An electronic smart dictionary is a dictionary that is computer interpretable (expressed in a formal language) and includes relations between the defined concepts, especially subtype-supertype relations. This enables to link additional knowledge to the concepts in the dictionary.
For example, a smart electronic dictionary enables a computer to 'know' that a line shaft pump is a subtype of a centrifugal pump and thus also shall have an impeller as one of its parts.

Fact

A fact is something that is the case. A fact is typically expressed as a relation between two objects. Typically facts are expressed in data exchange files or database tables.
For example: K-1201 <is a part of> U-1200.

Formal language

A formal language is a method to express *facts* in an unambiguous formally defined way, using a standardized electronic 'smart' dictionary and standardized relation types (or standard 'data model') to create expressions. Expressions in a formal language are computer interpretable.

Information

Information includes *data sets* and *documents* that convey meaning about a subject. The same information may be expressed in various ways on different *information carriers.*

Information carrier

An information carrier is a physical object that carries information. For example, several electronic files in various formats and multiple paper copies are information carriers that may carry the same information.

