# Document Management in Gellish

**Andries van Renssen**
**Gellish.net**

**Juni 2015**

## Inhoud

# 1 Introduction

This document describes the application of Gellish Formal English in combination with usage of the Gellish Search Engine as a document management system that uses a product model as a backbone for searching documents about the objects in the product model. As an example this document uses a real case product model of a large Liquefied Natural Gas (LNG) plant from a large oil company. The application didn't need any development of software, because the generic Gellish Search Engine is a generic piece of software that is able to read, browse and search for any information that is expressed in one of the formal languages of the Gellish family of formalized languages, whereas it can launch various document viewers. Thus, the only thing that was needed was: expressing the product model and the document information in Formal English.

The application demonstrates that Gellish formalized languages enable that universal software is developed that is capable of handling a large variety of application areas. Document management is just an example.

The Gellish Search Engine is a generic software system that uses the universal Gellish expressions as a universal database. The system is built as an application of the generic and powerful technology that is provided by the Gellish family of formalized natural languages, among which Formal English and Formal Dutch and according to the Gellish Information Modeling Methodology. Expressing information in a computer interpretable, universal and system independent formal language has various benefits. Examples of benefits are that it enables:

- Exchanging information between various systems of different parties without a need for data conversion;
- Combining and integrating information from multiple sources, including consistency management;
- Intelligent querying and applying reasoning with the rules of logic that are included in the definition of the formal language;
- Expressing and use of knowledge and requirements for generating design proposals in computer enhanced design as well as verification of products and processes on compliance with expressed requirements;
- Harmonization of data in systems.

The Gellish language definition is available as a computer interpretable basic language defining ontology and a number of domain specific taxonomic dictionaries [Ref. 4]. The language definition is described in 'Semantic Information Modeling in Formalized Languages' [2014, Ref. 2]. The Gellish modeling methodology is described in 'Semantic Information Modeling Methodology' [2015, Ref. 3]. The usage of the formal language is supported via the website www.gellish.net. The fundament of formal languages and the semantic modeling methodology is described in 'Formalized Natural Languages' [2014, Ref. 1].

# 2 The problem

An old LNG facility of about 3000 pieces of main equipment needed to be revamped and extended. However the documentation about the existing facilities was only available on paper and was not allowed to leave the site for a significant period of time. Therefore, the question was to provide a solution that would make the existing documentation available and accessible for one or more engineering contractors.

The existing documentation comprised of:

- 13000 paper drawings
- 500 binders with many tenth of thousands paper documents.

# 3 The solution

The solution was:

- Scanning of all drawings and documents, while recording their drawing headers and title pages (meta data) and file names in Formal English.
- Development of and Semantic Information Model of the facilities and the documents in Formal English, which included relations between the facility components and the documents and drawings.
- Usage of the Gellish Search Engine as a generic application, browser and launcher of various viewers to display pdf-files, doc-files, xls-files, dwg-files, etc.

# 4 Implementation

A company specialized in scanning large quantities of documents and drawings created an archive of files, together with data about the documents and drawings. This was combined with similar files and data about drawings and documents that were created by an Engineering Contractor. The combined meta data was converted into Formal English expressions.

According to the Gellish Semantic Information Modeling Methodology, a facility decomposition hierarchy was created, which consists of sections, sub-sections, process units, systems, equipment and instruments. Each design and actual piece of equipment and instrument, denoted by tag names and registration numbers of installed equipment was classified conform the kinds of objects that were available in the Formal English Taxonomic Dictionary. In the mean time, the data was cleansed by harmonizing various naming conventions, and typing errors, by introducing synonyms names for objects and drawings.

Because the facility consisted of two parallel trains, the facility was decomposed from a maintenance and operations perspective in two ways, one in sections, and one in trains. This resulted in a backbone facility model consisting of two integrated composition hierarchies as shown in Figure 1 and Figure 2.

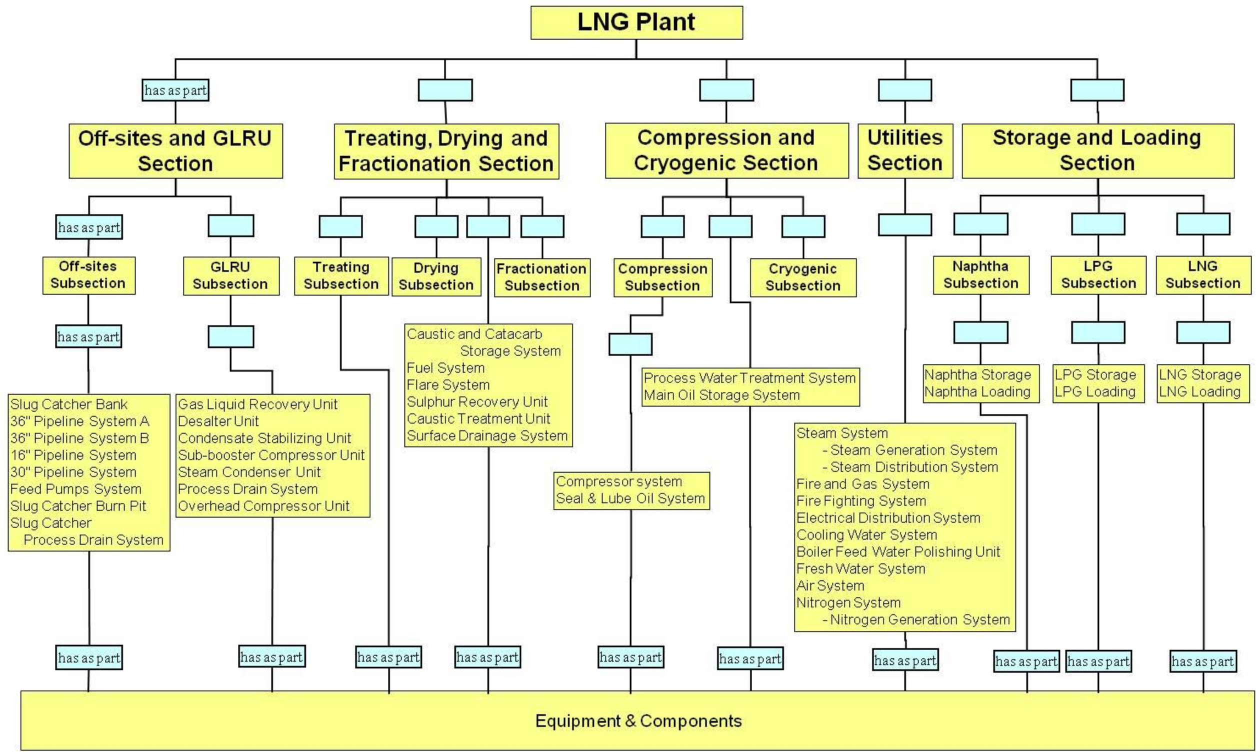


**Figure 1, Facility composition hierarchy by section**

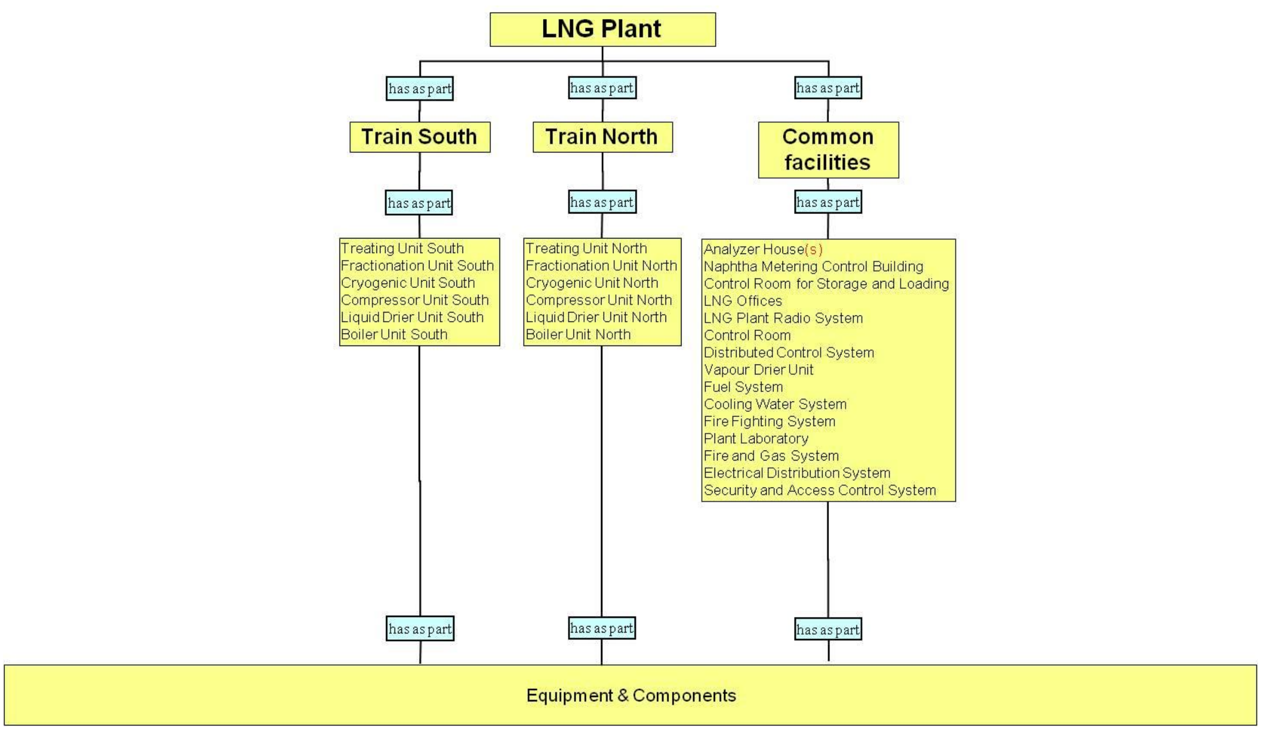


**Figure 2, Facility composition by trains**

Note that all relations between the various components of the facility model were 'composition relations' indicated by the phrase <has as part>, which is a standard Formal English concept and phrase. The big box at the bottom of both figures denote the same 3000 pieces of main equipment and instruments. In addition to the composition relations all component classification relations were added.

Furthermore, relations of other kinds between the various components were added. Example of such other relations are:

- Relations between motors and driven equipment, which kinds of relations are denoted as <is driver of> relations, which can also be searched via the inverse phrase is <is driven by>.
- Relations between safety valves and the equipment that they protect. Such kinds of relations are denoted by the phrase <is protected by>, which can also be searched via the inverse phrase: <is protecting>.
- Relations between design items (denoted by tags) and actually installed items (denoted by equipment registration numbers), which kinds of relations are denoted by the phrase <is installed for>.

The tables of content of the binders, and the information about the structure of the archive was used to create Formal English expressions about where to find the actual physical paper documents in the binders. This information, together with the drawing headers and meta data about each scanned document enabled to create 'document models' that express in Formal English about which facility components the documents provide information and how complex documents are composed.

The next step was to relate the documents and files to the facility components about which they provide information.

This provided an already sufficiently rich model for searching and displaying the drawings and other documents.

A general search facility enables to search on any term or term fragment. For example, Figure 3 shows a search for all things that are classified as a 'compressor'.

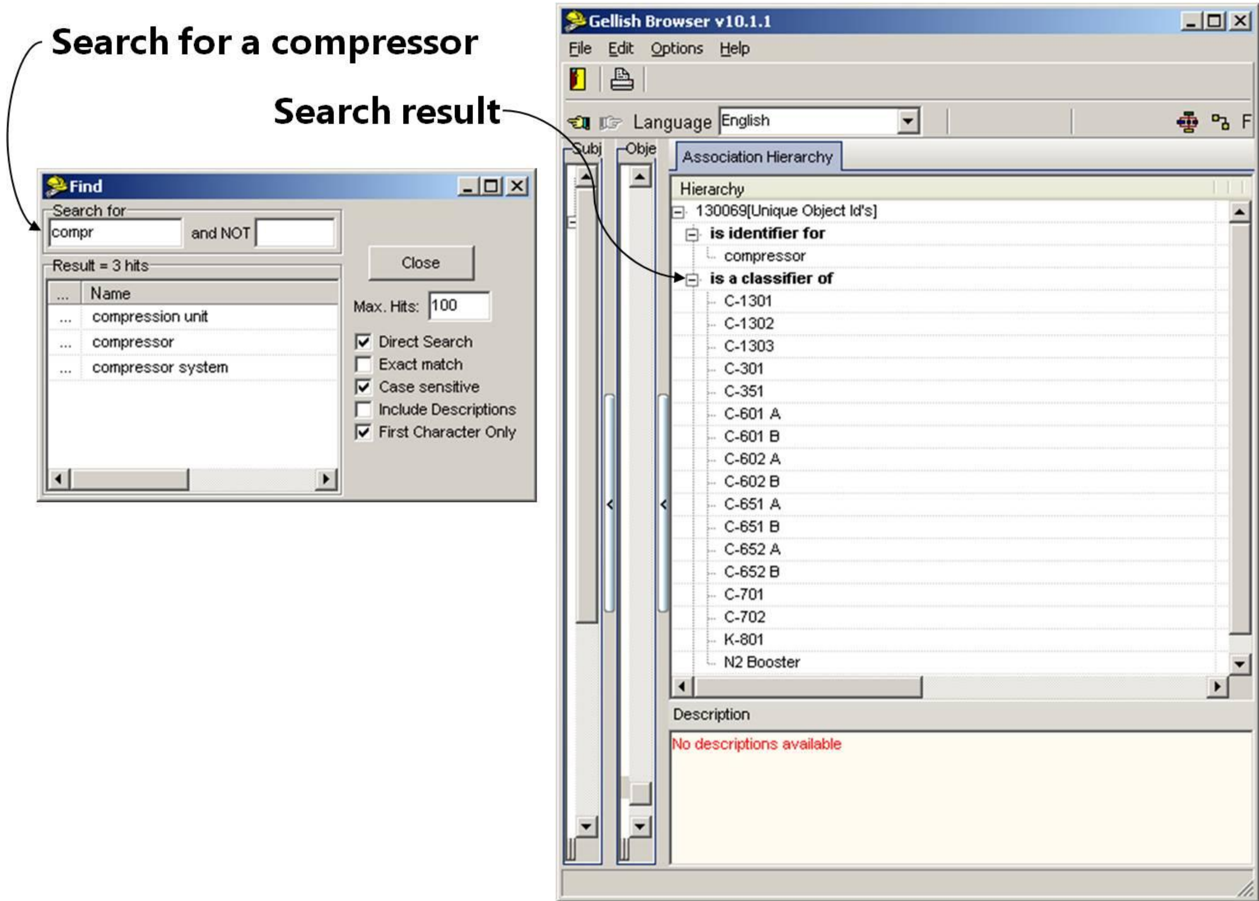


**Figure 3, Search for compressors**

The left hand of the figure shows a search window for any text string in the whole database.

The information model is in fact one big network of relations between all things in the model. A search implies a jump to one of the nodes in the network that form the 'object in focus'. The display window shows the selected object in focus, together with things that are related to it by relations of particular kinds as shown. Thus, after clicking the term 'compressor', the search results in Figure 3 show that the 'object in focus' is 130069, which is denoted by the term compressor in the taxonomic dictionary. Furthermore. The results show that all compressors are found, including also things that are classified by any of the subtypes of compressor, such as centrifugal compressor, etc.

By clicking on any of the displayed objects we can jump to another node in the network. For example, when we click on C-351, the application will display information about C-351 as shown in Figure 4.

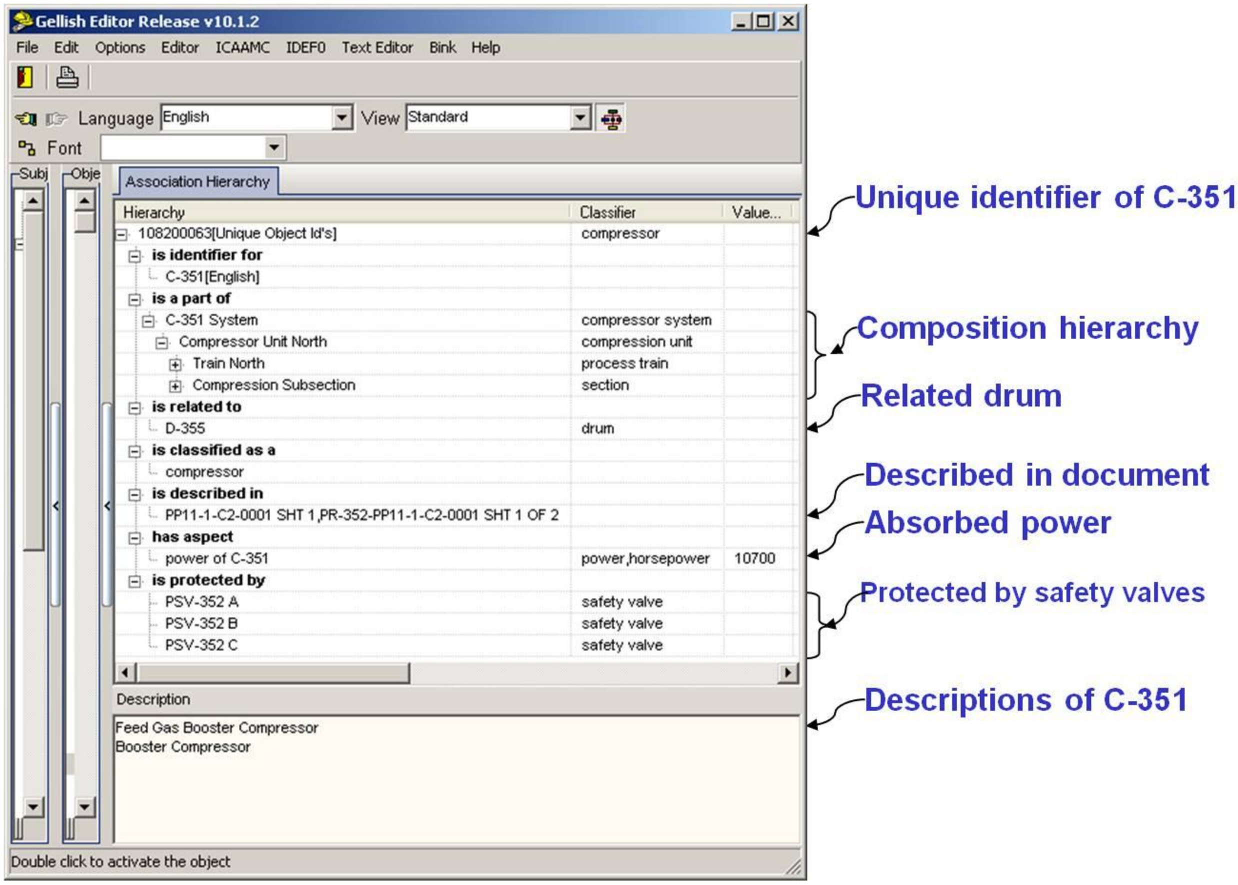


**Figure 4, Information about C-351**

Figure 4 shows the C-351 is classified as a compressor (the inverse relation of the previous view) and it shows at the bottom that the compressor is sometimes described as Booster Compressor and sometimes as Feed Gas Booster Compressor. The <is a part of> relations show that C-351 is a part of the C-351-system, which is a part of Compressor Unit North, which is part of two different higher level assemblies, etc.

If we click for example on 'Compressor Unit North', then we jump to that node in the network, which shown among others the composition (equipment list) of Compressor Unit North, as is shown in Figure 5.

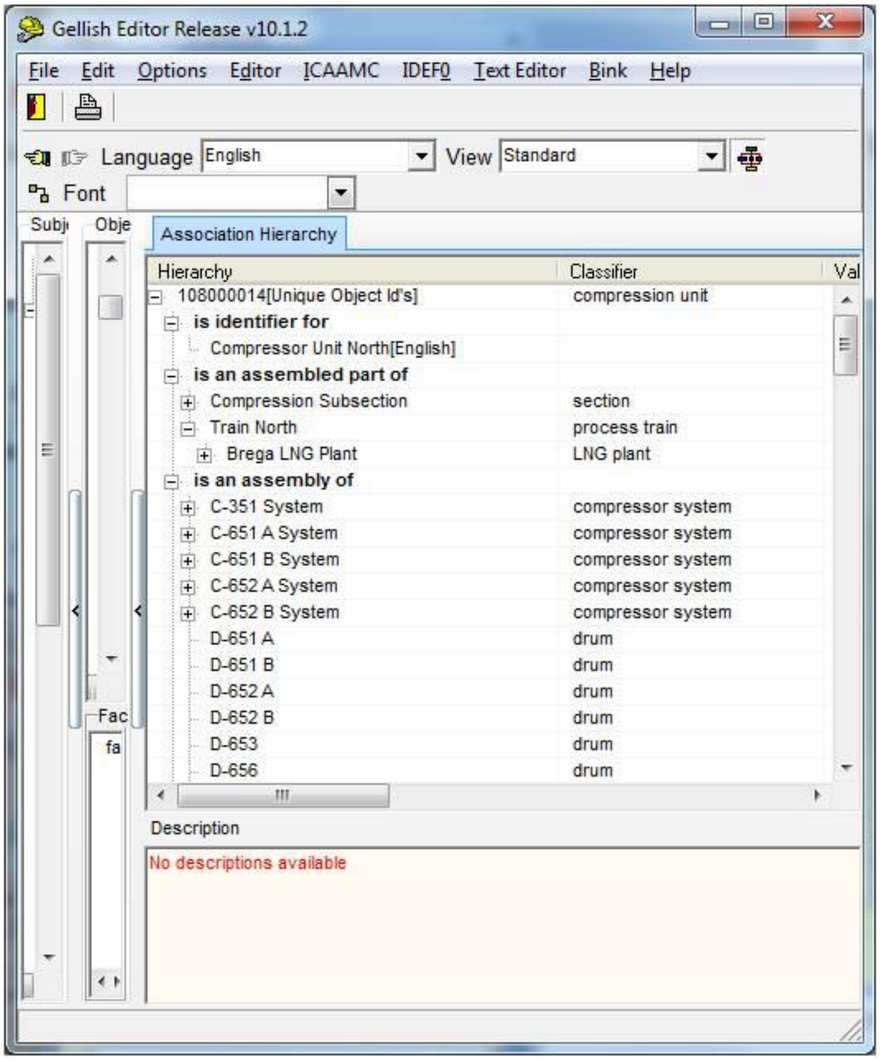


**Figure 5, Equipment list of Compressor Unit North**

Figure 4 also shows that C-351 is described in a document called PP11-1... If we click on that document the system will display its document model as shown in Figure 6.

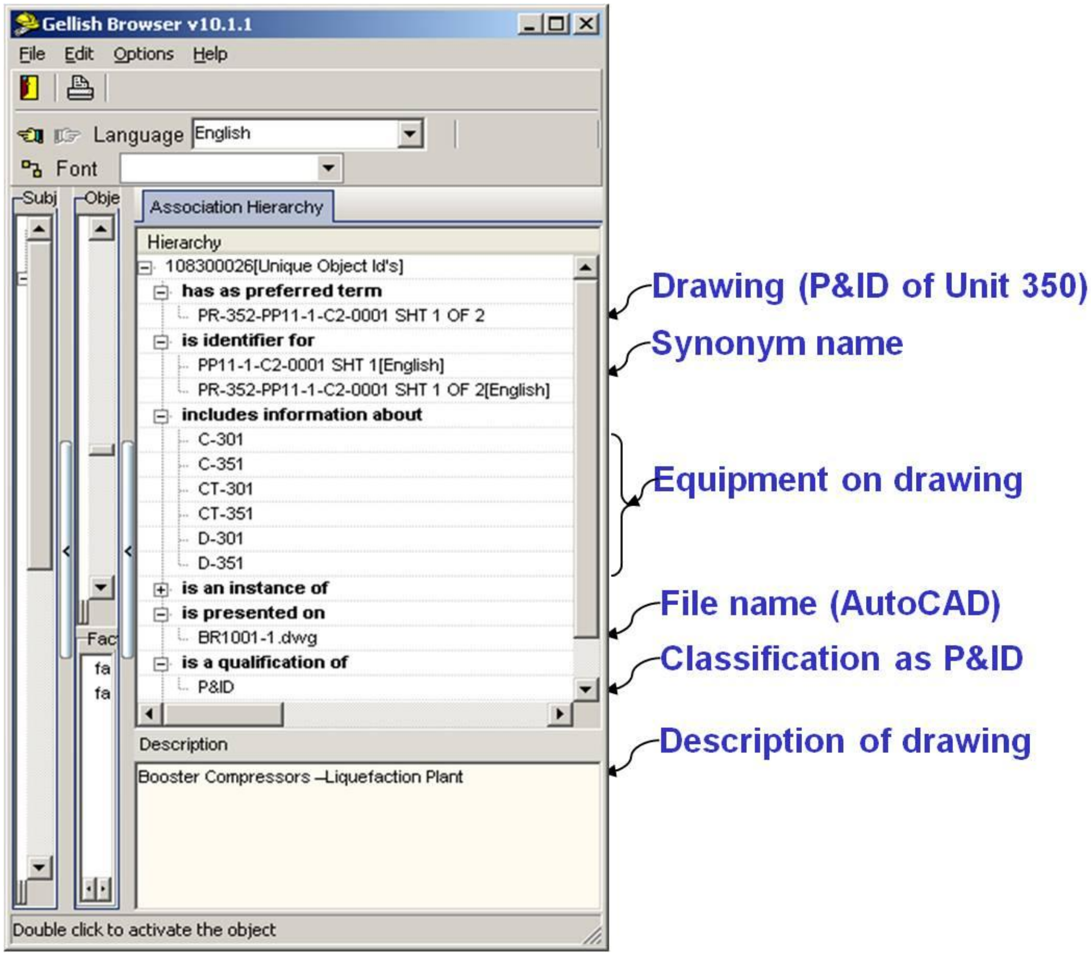


**Figure 6, Document model**

Figure 6 illustrates that the document is denoted by two different names at different occasions and it shows that the document <includes information about> the mentioned pieces of main equipment, it is presented on a particular dwg file and is qualified as a P&ID.

Thus clicking on the file name displays Figure 7.

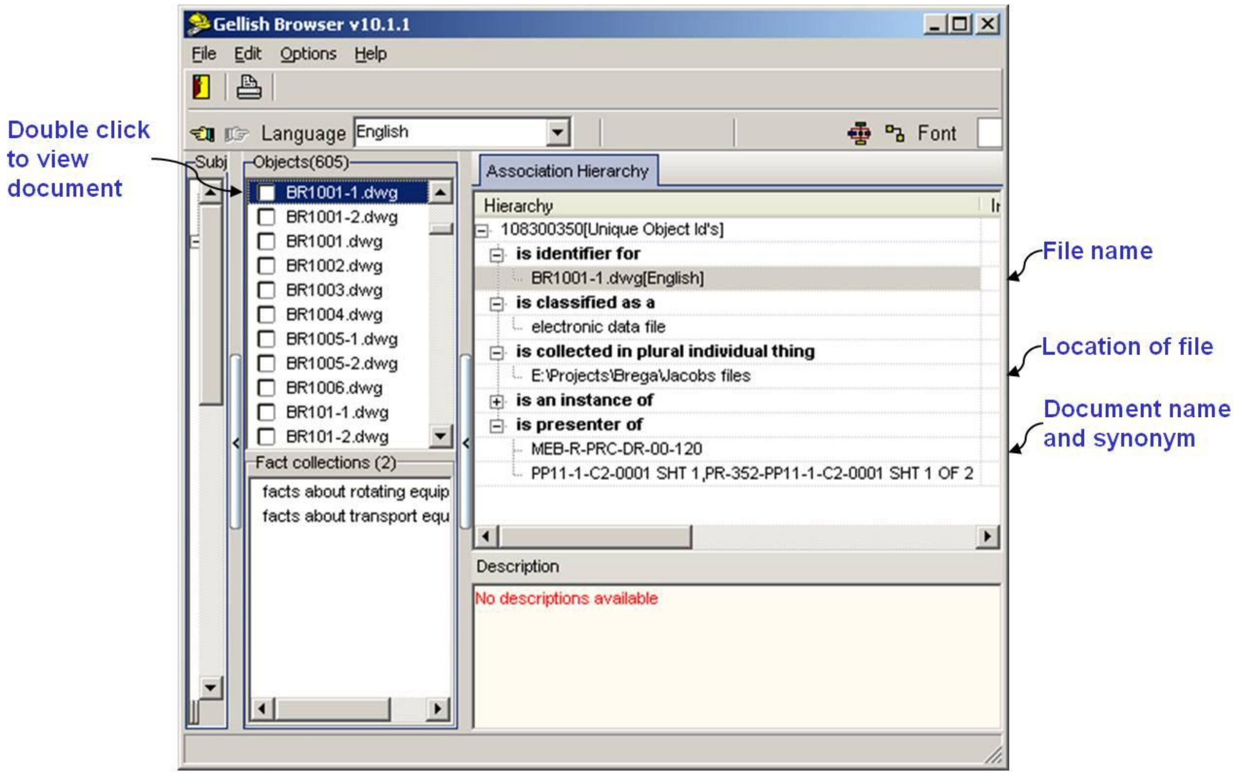


**Figure 7, Information about the found file**

Figure 7 shows where the file is located and enables to launch a viewer that displays the drawing, as is shown in Figure 8.

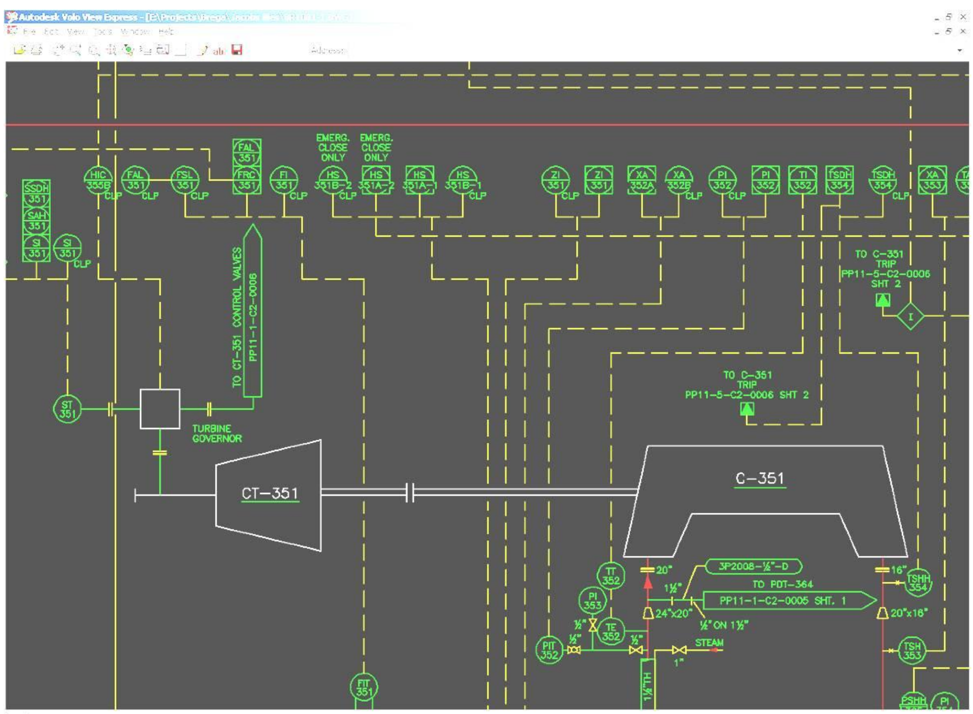


**Figure 8, Display of the drawing**

The above described principle can be extended by using 2D drawings as well as 3D models as user interface for searching information. This can be done with dumb graphics drawings by creating expressions in Formal English that describe the relations between the coordinates of text or symbols on drawings and views to the objects in the database. This enables software to find the objects by pointing to text of figures in the drawings. These capabilities are indicated in fig.

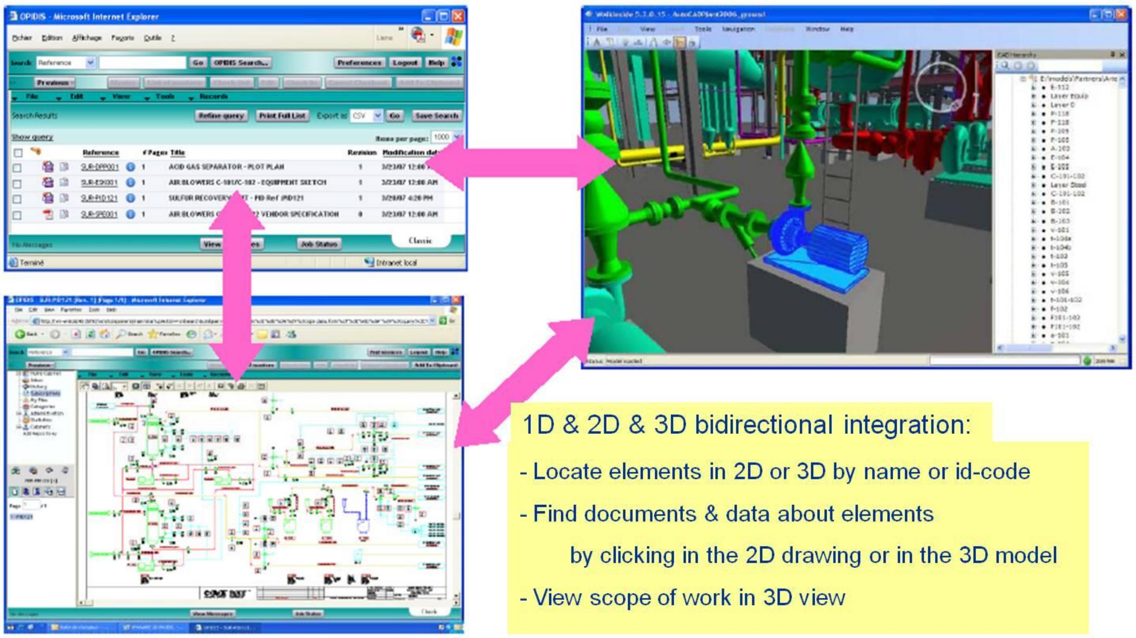


**Figure 9, 1D, 2D and 3D bidirectional data-document integration**

The above example allows that the data and documents are located in various locations. Thus it allows for a distributed database approach. There is no limitation to the scope of the database.

The whole modeling and implementation required four man month of effort.

# 5 References


1. Formalized Natural Languages, Lulu Inc, 2014; 2nd ed of Gellish, a generic extensible ontological language, PhD, Delft University of Technology, 2005. http://www.lulu.com/shop/dr-ir-andries-van-renssen/formalized-natural-languages/paperback/product-21904436.html.
2. Semantic Information Modeling in Formalized Languages, Lulu Inc., 2014. http://www.lulu.com/shop/dr-ir-andries-van-renssen/semantic-modeling-in-formal-english/paperback/product-21538016.html.
3. Semantic Information Modeling Methodology, Lulu Inc., 2015.
4. Gellish taxonomic dictionary-ontology of Formal English and Formeel Nederlands, http://www.gellish.net/index.php/shop/product/33-formal-english-dictionary-ontology-search-engine.html.
5. Example of a road network, http://www.gellish.net/index.php/downloads/file/3-gellish-english-example-road-v4-doc.html.