Confidential

Gellish Modelling Method

Part 8

Integration of 1D, 2D and 3D Product Models

by
dr. ir. Andries van Renssen
Gellish.net

March 2010

Table of content

1. Introduction

An integrated 1D, 2D and 3D product model is a model that is stored in one or more databases and that integrates conventional product data with drawings and spatial shape representations (3D models). Integration of such information from different sources requires that the information is expressed in a 'common language'. This document describes how this can be achieved using the system independent Gellish language, in this case Gellish English. The same can be done in Gellish Nederlands (Dutch) or any other Gellish variant.

A product model in 3D is a model of a physical object that includes a specification of its shape. When it is a part of a bigger assembly, such as a building, a transport system, a ship or a process plant, then the model also includes its position and orientation in a 3D space. A 3D model forms just a fraction of all the information that is typically stored about the product and the processes in which it is involved. Those other aspects include for example non-shape 1D properties, such as materials of construction, strengths, temperatures and pressures and also relations, such as assembly relations, connection relations with other products and involvements in processes and activities. This means that 3D shape models should be integrated in a complete set of information and knowledge about a facility. The modelled collection of all facts about a facility is called a Facility Information Model (FIM) or more specialised a Building Information Model (BIM), a Plant Information Model, etc.

3D product models are typically created by using software packages of various suppliers. The resulting models are typically stored in the proprietary formats of those packages. For example, in DWG, DXF, CMG, DGN or RVT format. Each of such formats has its own limitations on the type of data that can be stored. Data that is stored in one format can only be converted to another format when a single or bi-directional converter is created and when the receiving format is suitable for storing all kinds of data that is to be converted.

Non-3D information about a facility is typically stored in other proprietary systems with their own proprietary formats or data structures. For example the structure of ERP databases or PLM databases. Each of such systems has a different structure of its data and also uses different definitions for the concepts that it applies. Therefore, exchange of data between systems and integration of data of different sources raises all sorts of data conversion issues.

The Gellish Modelling Method offers a solution to these data integration issues by providing:

- An open standard data structure for a semantic database (the Gellish Database tables).
- A standard electronic Knowledge Base, consisting of a smart English Dictionary that defines concepts and relation types, a Taxonomy and an Ontology.
- A method to apply the language to store and exchange information, such as information in Facility Information Models (FIMs, BIMs, etc.), or to extend the dictionary, taxonomy and ontology.

The first two components together define the Gellish English language, the last component defines the application of the language.

The method is based on various ISO standards, such as ISO 10303 and ISO 15926. For example, its basic geometric concepts (primitives) to describe shapes as included in the Gellish Dictionary are based on the ISO STEP standard for geometric objects: ISO 10303-42 and ISO 15926-3.

This document describes how to create, store and exchange any data about a facility, being 1D, 2D and 3D, in one consistent, standardised, integrated and system independent way, using the Gellish

Modelling Method. The focus of this document is on 3D shapes and their integration with 1D data. Other parts are:

- Part 1, Architecture, describes an overall architecture for information and its management.
- Part 2, Creation of Domain Dictionaries / Taxonomies.
- Part 3A, Knowledge Modelling (Creation of Ontologies) and Part 3B, Requirements Modelling.
- Part 4, Creation and Use of Facility Information Models", focuses on the creation of models with primarily 1D data and their integration with documents.
- Part 5, Modelling of Activities and Processes. This part specifies how occurrences are modelled, together with the involved parties and objects.
- Part 5A, Measurements and Observations.

2. Integration of data and documents

During a project data are usually created independent of each other, using various different systems. The conventional approach is that a subset of the most important data is later centralised by:

- Extracting the data from the various systems,
- Converting the data to a common standard in terminology and data structure,
- Resolution of inconsistencies with data in the receiving system and
- Importing the data in an all-including database.

Real integration of such data is nearly impossible. The main reasons are:

- The conversion of terminology and data structures appears to be very costly and time consuming, especially because of the many inconsistencies and differences of definitions.
- Standards usually cover only a part of the scope of the data and they are fixed.
- There are no systems available that can store and manage all the kinds of 1D, 2D and 3D data that are created during a project.

Integration of data therefore requires that the data are kept decentralised, but managed across systems.

Real integration of data that are created during a project using different source systems can only be achieved by applying two different rules:

1. By using common standards for terminology and data structure (semantics) for all project data.
2. By sending queries and responding by messages in a standard language.

The Gellish Information Management Method and the part that is described in this document uses a flexible, extendable standard language which enables that it provides a guide towards such a solution.

3. 1D, 2D and 3D product models

An integrated product model typically contains one dimensional (1D), two dimensional (2D) and three dimensional (3D) aspects. A model may also contain higher dimensional aspects such as shape variations as a function of time and it may be integrated with documents and data sets in proprietary file formats, such as doc, ppt or xls formats.

1D aspects are aspects that are described by a single dimension (or value), such as length, temperature, density, diameter or duration. Their magnitude can be quantified by a single number on a scale or by a single qualitative qualification, such as 'flat' or 'hot'.

2D aspects typically describe positions in a two dimensional flat plane or points on a curve. The magnitude of these kinds of aspects are normally described by two values each on its own scale. An example of a 2D aspect is the location of building B1 within an area A1. The position can be described by two value (200 m, 130 m) in respectively the x and y direction from the origin of a Cartesian coordinate system of area A1. Another example is the operating curve of a pump, which can be described by a differential pressure as a function of flow rate.

A 3D spatial shape can be described in a quantitative way by quantifying three dimensions, or three directions in a coordinate system. Basically this holds for every point in or on a physical object. However, a line, surface or volume can also be described by mathematical functions (formulae) in a coordinate system. Each shape function covers a large collection of points. By specifying the position of such a shape function in a 3D coordinate system the position of the whole line, surface or volume is determined. In addition to the position where a shape is located, it is also required to specify the orientation of the shape relative to the environment in which it is placed. Such an orientation requires three additional aspects, being angles of rotation.

4D aspects, or higher order dimensional aspects are for examples of four dimensional aspects that are time variation of shapes or movements or the operating space of a centrifugal pump, where the flow rate and differential pressure are dependent on the speed of rotation and the diameter of the impeller.

These higher order dimensional aspects and the relations with files in other formats can also be described in Gellish, but are not further discussed in this document.

4. Coordinate systems

The 2D or 3D shape of a symbol, cross section or physical object is normally defined with respect to the 'own' 2D or 3D coordinate system of the described thing. Furthermore, those shapes are then placed in the coordinate system of a bigger assembly, which on its turn may be placed in again another coordinate system, such as the geographic coordinate system of the earth. Therefore, we should first discuss how coordinate systems are defined and used in Gellish.

A coordinate system is a system of multiple scales for multiple dimensions. Those scales may be different or may be the same for all dimensions. Thus a 3D coordinate system has three dimensions, each with its own scale. For example, the three dimensions are three distances and the scale for each distance is mm.

The scale of any coordinate in the same direction is the same for each measure in that direction in the coordinate system. For example, if an x-coordinate is measured in dm3/s, then every point in that coordinate system has an x-coordinate value with that same scale (unit of measure). This means that it is not required to specify the scale for each value in the coordinate system.

There are several kinds of coordinate systems. For example:

- Cartesian coordinate systems (2D and 3D)
- Cylindrical coordinate systems
- Spherical coordinate systems
- Geographic coordinate systems
- Parametric coordinate systems

The Gellish Dictionary and Taxonomy already contains definitions of the first four kinds of concepts, but anybody can add his own concepts or propose to add concepts to the standard language definition. The next paragraphs describe how these coordinate systems are defined in Gellish.

4.1 Cartesian coordinate systems

A coordinate system is typically a Cartesian coordinate system. It is defined by two or three axis that point in directions that are perpendicular to each other. The axes in a Cartesian coordinate system are normally called the x-axis, the y-axis and the z-axis. The z-axis in a right handed 3D Cartesian coordinate system has the direction that a corkscrew goes when turning from the x to the y-direction.

For a spatial coordinate system, each of the three dimensions is a distance. The Dictionary may define that as follows:

924995	3D Cartesian coordinate system	4664	has by definition as dimension a	550212	distance in x-direction
924995	3D Cartesian coordinate system	4664	has by definition as dimension a	550213	distance in y-direction
924995	3D Cartesian coordinate system	4664	has by definition as dimension a	550214	distance in z-direction

A right handed 3D Cartesian coordinate system is illustrated in Figure 1.

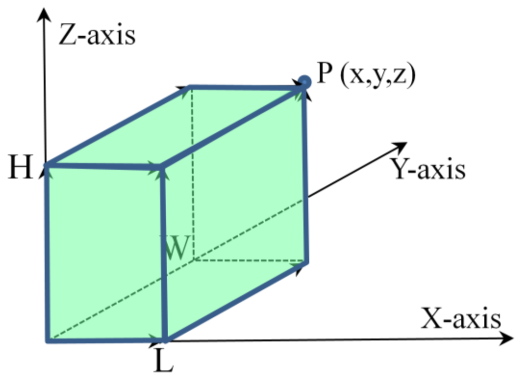

Figure 1, A Cartesian coordinate system

A coordinates of a point (P) in a particular (individual) coordinate system that is classified by this kind is by definition specified by three distances x, y and z, being three distances in the direction of the X-axis, Y-axis and the Z-axis. The three scales for the distances in the three directions are normally the same. Thus, if the scale for a particular 3D spatial Cartesian coordinate system is mm, then the scale of the x-direction, the y-direction and the z-direction are all in mm. This can be specified in one go by specifying the scale for the whole system.

For example, if a physical object, such as the solid block in Figure 1, is defined with respect to a particular (individual) 3D coordinate system CoordSys-201, then that coordinate system shall be defined including the scales of its dimensions as follows:

1	CoordSys-201	1225	is classified as a	924989	right handed 3D Cartesian coordinate system	
1	CoordSys-201	5716	has as dimension a	550212	distance in x-direction	mm
1	CoordSys-201	5716	has as dimension a	550213	distance in y-direction	mm
1	CoordSys-201	5716	has as dimension a	550214	distance in z-direction	mm

The representation of a shape (or space) can be implemented in various ways, such as a solid model, a surfaces model or a wire frame model. Each implementation is based on one or more geometric formulae that are defined in one of the possible coordinate systems. When a shape is described in a Cartesian coordinate system, then that shape is essentially defined by one or more mathematical functions (algorithms or formulae) in three dimensions, abbreviated as x, y and z, which function has the sizes of the shape as its parameters.

The various possible implementations mean that a shape can be visualized (presented) through the use of various different computer executable programs, for example in the form of Macros, written in some computer interpretable programming language.

The modeling of shapes is further clarified in the next chapter.

4.2 Cylindrical coordinate systems

A cylindrical coordinate system has by definition three coordinate axes, as is illustrated in Figure 2.

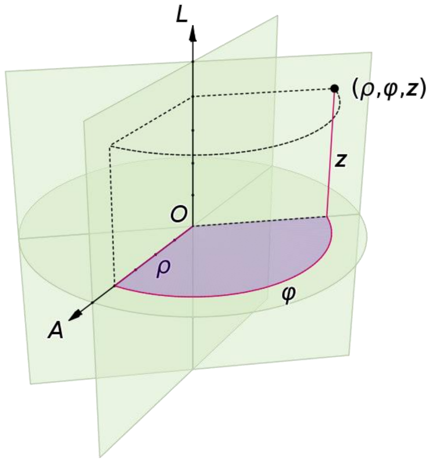

Figure 2, A cylindrical coordinate system

The coordinates are by definition: a distance or radius from the centre (ρ or rho), an angle (φ or phi, also called the azimuth angle) relative to a reference direction and a z-axis that is perpendicular to a chosen plane in which the radius and the angle are defined. The positive z-axis is in the right hand corkscrew direction. If the chosen plane is assumed to be horizontal, then the z-axis is called the height. This is defined in the Dictionary as follows:

928237	cylindrical coordinate system	4664	has by definition as dimension a	550681	radius
928237	cylindrical coordinate system	4664	has by definition as dimension a	550029	angle
928237	cylindrical coordinate system	4664	has by definition as dimension a	550126	height

For example, a particular 3D cylindrical coordinate system CoordSys-301 is defined as follows:

2	CoordSys-301	1225	is classified as a	928237	cylindrical coordinate system	
2	CoordSys -301	5716	has as dimension a	550681	radius	mm
2	CoordSys -301	5716	has as dimension a	550029	angle	deg
2	CoordSys -301	5716	has as dimension a	550126	height	mm

4.3 Spherical coordinate systems

A spherical coordinate system is a 3D coordinate system as is illustrated in Figure 3.

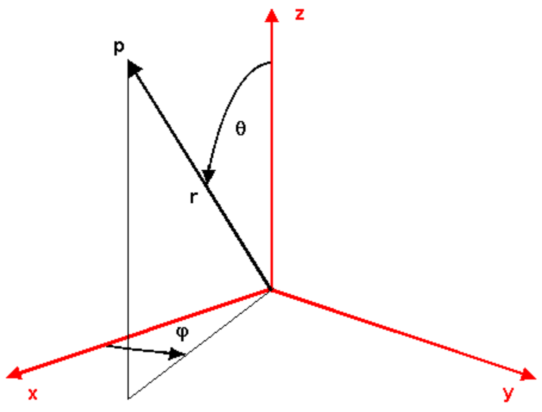

Figure 3, A spherical coordinate system

The position of a point (p) in a spherical coordinate system is specified by three coordinates (r, θ, φ): the radial distance of that point from a fixed origin (radius or r), its inclination angle (theta or θ) measured from a fixed zenith direction, and the azimuth angle (phi or φ) of its orthogonal projection on the x-y plane.

The second angle (the inclination angle) is often replaced by an elevation angle measured from the reference plane perpendicular to the zenith direction. Note that this is the same kind of spherical coordinate system, but the second parameter is classified as an elevation angle. An elevation angle is 90 degrees (π/2 radians) minus an inclination angle.

The radial distance is also called the radius or radial coordinate, and the inclination may be called colatitude, zenith angle, normal angle, or polar angle.

The sign of the azimuth is determined by choosing what is a positive sense of turning about the zenith. This choice is arbitrary, and is part of the coordinate system's definition: right handed or left handed.

A spherical coordinate system may be defined in the Dictionary as follows:

928252	spherical coordinate system	4664	has by definition as dimension a	550681	radius
928252	spherical coordinate system	4664	has by definition as dimension a	553705	inclination angle
928252	spherical coordinate system	4664	has by definition as dimension a	553705	elevation angle
928252	spherical coordinate system	4664	has by definition as dimension a	553706	azimuth angle

For example, a particular 3D spherical coordinate system CoordSys-401 is defined as follows:

3	CoordSys-401	1225	is classified as a	928252	spherical coordinate system	
3	CoordSys -401	5716	has as dimension a	550681	radius	mm
3	CoordSys -401	5716	has as dimension a	553705	inclination angle	deg
3	CoordSys -401	5716	has as dimension a	553706	azimuth angle	deg

Note: If other choices are made for the definition of the angles, then they shall be classified by the otherwise defined kinds of angles.

4.4 Geographical coordinate systems

A geographic coordinate system is a coordinate system that enables every location on Earth to be specified in three coordinates, using mainly a spherical coordinate system.

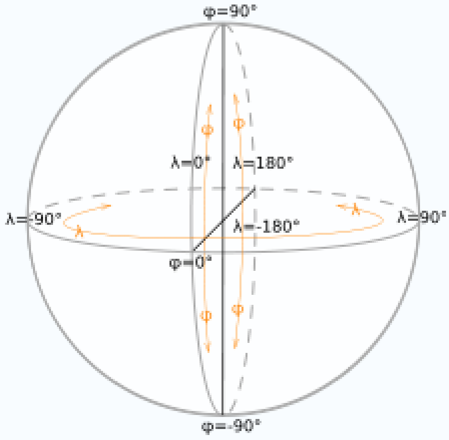

Figure 4, Latitude phi (φ) and Longitude lambda (λ) in a geographic coordinate system

In geography and astronomy, the elevation and azimuth (or quantities very close to them) are called the latitude and longitude, respectively; and the radial distance is usually replaced by an altitude or depth.

Thus, a geographic coordinate system may be defined in a Gellish Domain Dictionary as follows:

928252	spherical coordinate system	4664	has by definition as dimension a	553707	altitude
928252	spherical coordinate system	4664	has by definition as dimension a	550095	depth
928252	spherical coordinate system	4664	has by definition as dimension a	553060	latitude
928252	spherical coordinate system	4664	has by definition as dimension a	553059	longitude

The altitude (with a positive value in upward direction) is measured from a datum. Typically being the mean sea level (MSL) surface or above ground level. To specify which reference datum is used for a particular height, it shall be classified by one of the subtypes of altitude, such as a true altitude (means sea level altitude) or an absolute altitude (ground altitude).

The radius relative to the centre of the earth can be specified either as an altitude or as a depth (with a positive value in downward direction), both with reference to a datum. Furthermore, the depth can be specified in various different ways, especially as true vertical depth (TVD) or as measured depth (MD) along a path that is not necessarily vertical. Furthermore the reference datum can vary, so that a subtype of TVD or MD shall be selected for classification of an individual length. For example as true vertical depth below ground level (TVDGL). For example, the radius of an individual point can be specified in two equivalent ways:

7	H1	1225	is classified as a	553708	true altitude
7	H1	5025	has on scale a value equal to	300	m

or as:

7	H1	1225	is classified as a	553713	TVDGL
7	H1	5025	has on scale a value equal to	-300	m

4.5 Individual coordinate systems

4.5.1 RD coordinate system

The RD coordinate system (Rijksdriehoekscoördinatensysteem) is an individual 2D Cartesian coordinate system for the Netherlands that is based on the European Terrestrial Reference System (ETRS89) as a reference coordinate system. The origin (0,0) of the RD coordinate system is defined at a point 155 km west and 463 km south of the point of a tower in Amersfoort (the 'Onze Lieve Vrouwentoren').

A Dutch national altitude as third dimension is a height relative to NAP. The NAP ('Normaal Amsterdams Peil') is a reference height that slightly differs from the geoid. The European Vertical Reference System (EVRS) uses the NAP as its reference surface.

4.5.2 European Terrestrial Reference System

The European Terrestrial Reference System 1989 (ETRS89) is a Earth-Centered, Earth-Fixed (ECEF) geodetic Cartesian reference coordinate system, in which the Eurasian Plate as a whole is static. The coordinates and maps in Europe based on ETRS89 are not subject to change due to the continental drift.

4.5.3 World Geodetic System

The World Geodetic System (WGS84), is a particular worldwide reference coordinate system that is defined for Global Positioning (GPS). It comprises a standard coordinate frame for the Earth, a standard spheroidal reference surface (the *datum* or *reference ellipsoid*) for raw altitude data, and a gravitational equipotential surface (the *geoid*) that defines the *nominal sea level*. The coordinate origin of WGS 84 is meant to be located at the Earth's center of mass; the error is believed to be less than 2 cm.

In WGS 84, the meridian of zero longitude is the IERS Reference Meridian. It lies 5.31 arc seconds east of the Greenwich Prime Meridian, which corresponds to 102.5 metres (336.3 feet) at the latitude of the Royal Observatory.

The World Geodetic coordinate system can be defined in Gellish as follows:

1	WGS84	1225	is classified as a	928252	spherical coordinate system

4.6 Placement in a geographic coordinate system

The position and orientation of an object on earth can be specified by the location and rotation angle of its own coordinate system with respect to WGS84.

For example, assume that building B-101 has a shape that is described by a Model-B-101. That model is defined in its own (individual) Cartesian coordinate system. That coordinate system CoordSysB-101 is placed relative to WGS84. This can be specified using the 'placement in a 3D space' function as follows:

2	B-101	1727	has as aspect	3	Shape-B-101
3	Shape-B-101	5714	is modelled by a correlation as	4	Model-B-101
4	Model-B-101	1772	is defined in coordinate system	3	CoordSysB-101

3	CoordSysB-101	1225	is classified as a	924989	right handed 3D CCSystem
3	PlacementMBinWGS	1225	is classified as a	5718	placement in a 3D space
3	PlacementMBinWGS	5722	is a correlation with as reference object	1	WGS84
3	PlacementMBinWGS	4886	is a correlation with as subject	3	CoordSysB-101
3	PlacementMBinWGS	5348	has as parameter	4	24º 28’ 0”
4	24º 28’ 0”	1225	is classified as a	553059	longitude
3	PlacementMBinWGS	5348	has as parameter	4	54º 25’ 0”
4	54º 25’ 0”	1225	is classified as a	553060	latitude
3	PlacementMBinWGS	5348	has as parameter	5	0, 0, 0
5	0, 0, 0	1225	is classified as a	553727	orientation in 3D

Note: CCSystem is a synonym of Cartesian Coordinate System.

In the above specification the height (altitude) is not explicitly specified. For WGS84 the default altitude is ‘on the surface of the earth’.

5. Dimensional aspects

The dimensional aspects can be distinguished in single dimensional aspects (1D aspects), such as distance, diameter, length and thickness, two dimensional aspects, such as lines, curves, circles, surfaces and 2D coordinates in a flat plane, and three dimensional aspects, such as 3D shapes and 3D coordinates in a 3D coordinates space.

We will use two examples:

1. The shape of a wall in a building, with an opening and a window.
2. The shape of a tank, with a bottom, a wall and a roof.

Example 1:

Wall-101 has as 1D aspects a length of 5000 mm, a height of 2300 mm and a thickness of 300 mm. Its 3D shape, ShapeW-101, can be represented by a ‘rectangular block’ in a 3D Cartesian coordinate system as is illustrated in Figure 1.

Example 2:

The shape of tank (Tank-201) is specified in a 3D Cartesian coordinate system as is illustrated in Figure 5. The tank has as 1D aspects a diameter D of 5000 mm, a height H of 2500 mm and a wall thickness of 5 mm. As 3D aspect it has a wall with a cylindrical shape. The tank could be completed with a cylindrical bottom plate and a conical roof (not shown in the figure).

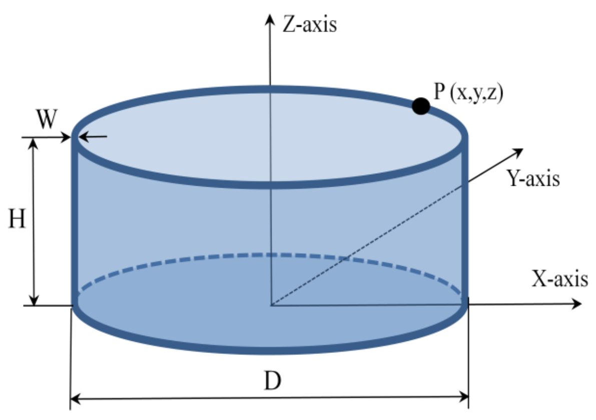

Figure 5, The shape of a cylindrical tank

5.1 One dimensional aspects (1D)

The recording of values of 1D aspects is usually pretty straight forward. The dimension is what we typically call a property or characteristic.

Example 1:

The properties of Wall-101 can be recorded in a standardized, computer interpretable way in Gellish as follows:

100	Wall-101	1225	is classified as a	40217	wall	
100	Wall-101	1727	has as aspect	101	WL-101	
101	WL-101	1225	is classified as a	551353	length	
101	WL-101	5025	has on scale a value equal to	922355	5000	mm
100	Wall-101	1727	has as aspect	102	WW-101	
102	WW-101	1225	is classified as a	550464	width	
102	WW-101	5025	has on scale a value equal to	920466	300	mm
100	Wall-101	1727	has as aspect	103	WH-101	
103	WH-101	1225	is classified as a	550126	height	
103	WH-101	5025	has on scale a value equal to	922285	2300	mm

These and similar expressions of facts contain a number of standard concepts and phrases that are selected from the Gellish Dictionary. For example, the concepts wall, length, 5000 and mm and also the phrases that express relation types, such as ‘is classified as a’ and the unique identifiers (UID’s) above 1000. Names of individual things and UID’s below 1000 are free.

Note: Details on allocation of other UID’s are discussed in part 4 of the Gellish Modelling Method.

Exercise: Define the material of construction of the wall in a similar way and select the standard concepts with their UID’s from the a Gellish Dictionary (such as the ‘Objectenbibliotheek voor de Bouw’: IFD-NL).

Example 2:

The properties of tank Tank-201 can be recorded in a standardized, computer interpretable way in Gellish as follows:

210	Tank-201	1225	is classified as a	520686	cylindrical tank	
210	Tank-201	1727	has as aspect	211	D-201	
211	D-201	1225	is classified as a	550206	external diameter	
211	D-201	5025	has on scale a value equal to	922355	5000	mm
210	Tank-201	1727	has as aspect	212	H-201	
212	H-201	1225	is classified as a	550126	height	
212	H-201	5025	has on scale a value equal to	922605	2500	mm
210	Tank-201	1727	has as aspect	213	Th-201	
213	Th-201	1225	is classified as a	550303	wall thickness	
213	Th-201	5025	has on scale a value equal to	920105	5	mm

In a similar way we can specify other properties of the tank, such as the material of construction.

5.2 Representation of 3D shapes

A shape of a physical object is an aspect of the object. Such an aspect can be represented by one or more generalized correlations (one or more mathematical functions or formulae) for kinds of shapes. Each correlation is defined in its own 3D coordinate system. Those generalized correlations are then tuned by a number of parameter values that are specific for the dimensions of the individual shape. Such generalized correlations are typically implemented in executable programs (or algorithms) that are written in some computer programming language, often called macro's. A library of such programs can be implemented for example as a DLL (dynamic link library). For example, a company might offer a DLL library with the graphical primitives that are defined in ISO 10303-42. *This document describes how such a software component library is linked to the data in a Gellish Database*.

The Gellish Dictionary, or your proprietary Gellish domain dictionary, typically contains the definitions (names and functionality description) and the parameters of existing generalized software components. This will be illustrated in the next paragraphs with the correlations for a block and for a hollow cylinder.

Given the availability of a library of algorithms, an individual shape can be described in Gellish by classifying it conform the kind of shape for which the algorithm is applicable and by providing the algorithm with the parameter values of the individual shape. As the Gellish Dictionary is extensible, any algorithm for any kind of shape can be added, so that any shape can be described in Gellish.

5.2.1 Separate components – separate shapes

Data exchange between various parties, such as in a supply chain, will include data exchange about components of an assembly. Therefore it is important that each component physical object in an assembly can be recognized as separate objects with its own properties and shape. Assemblies of physical objects should therefore be decomposed into single component physical objects, whereas each component is described by its own separate shape. This implies that the shape of a whole assembly should not be described by its own correlation, but should be composed of correlations that describe the shapes of its components.

5.2.2 Standardization of (basic) shape representations

Data exchange or data integration of definitions of shapes between systems of different suppliers is often impossible, because different software applications usually implement different correlations and make use of different libraries of executable programs (macro's) for shape representation.

The solution to this issue is to standardize the software components, or at least to standardize the functionality, names and parameters of shape representation correlations.

The fundament for harmonization of shape representations between different software packages is available when the ISO standard basic shapes (primitives) are commonly used. The Gellish Dictionary contains standard basic shapes that comply with the ISO STEP standard for geometric objects ISO 10303-42 and ISO 15926-3 and is extended with more advanced shapes that can be composed of basic shapes. The standard shapes can be selected from the Gellish Domain Dictionary for Mathematics and Shapes. Other shapes and their mathematical definition can be added conform the Gellish Dictionary Extension Procedure.

Even single component physical objects can have complex shapes for which no executable program is available. Then it is necessary to decompose the shape into partial shapes, until a standard shape is found for which an executable program is available that can present such a shape on a drawing or that can visualize it on a screen.

5.2.3 Parameterized representations of shapes

A shape representation executable program or Macro can usually present any shape of a particular *kind of shape,* because it normally contains a generalized algorithm that is tuned by particular parameter values to represent individual shapes and sizes. Such a program is an implementation of one or more parameterized mathematical functions. The program or Macro will have a parameter list that has as elements the properties of the physical object or part which shape is represented by the function and that may include other presentation parameters.

For example, the Wall-101 has a shape, called ShapeW-101, that can be classified as a 'block shape' as is illustrated in Figure 1. This is specified as follows:

100	Wall-101	1727	has as aspect	110	ShapeW-101
110	ShapeW-101	1225	is classified as a	550810	block shape

Such a shape can be represented by a mathematical function (correlation), called a 'solid block'. The 'solid block' correlation can be described by encoding as follows:

$$\mathbf{F(x,y,z) = x + y + z}$$
$$\textbf{where:}\quad \mathbf{0 =< x =< L}$$
$$\mathbf{0 =< y =< W}$$
$$\mathbf{0 =< z =< H}$$

Figure 6, A mathematical function that can represent the kind 'block shape' in general.

Note that the above correlation is defined in a 3D Cartesian coordinate system with the origin of the block (0,0,0) located at a corner and that the correlation has three parameters: length, width and height.

An individual shape can be represented by such a parameterized correlation, by specifying the correlation (or an executable software program that implements the correlation), together with a specification of the values of the parameters.

So, assume that for the generation of a picture conform the above 'solid block' correlation with particular parameter values there is a proprietary Macro available, called SolidBlockMacro.

Then the shape of the wall can be specified in Gellish as follows:

110	ShapeW-101	5714	is modelled by a correlation as	111	SBlock-101
111	SBlock-101	1225	is classified as a	911239	solid block
111	SBlock-101	1876	is defined in coordinate system	150	CoordSys-101
111	SBlock-101	4962	has as parameter	101	WL-201
111	SBlock-101	4962	has as parameter	102	WW-201
111	SBlock-101	4962	has as parameter	103	WH-201

Notes:

1. On the second line it is specified that the shape is described by a particular (individual) correlation that has specific individual parameter values.
2. On the third line the correlation is classified by a predefined kind of correlation. The definition of this class defines which parameters are required for correlation SBlock-101.

3. The correlation (911239) 'solid block' is a standard primitive correlation that is selected from the Gellish Dictionary and that is also defined in ISO 10303-214. It may be replaced by an implementation that is a subtype of the standard correlation.
4. The last three lines refer to the parameters of the correlation. The parameters were already quantified and classified in par. 5.1. The classification of the parameters shall comply with the requirements of the standard correlation for the kind 'solid block'.

The Gellish Domain Dictionary on shapes may be extended with additional standard correlations to represent shapes. Such definitions may include the specification of the parameters of the correlation.

During data exchange it is essential that the sending and receiving systems both use the same (standardised) correlations or at least implementations of correlations that have identical functionality. This is the reason why basic geometric correlations (primitives) are standardised in ISO 10303-42 and in ISO 15926-3 and that those correlations are included in the Gellish Domain Dictionary for Shapes. Other correlations, such as from IFC may also be used. Description of a geometric model in Gellish requires that the software that uses the model complies with a library of correlations that appears in the model. For example, the above model requires that a 'solid block' correlation as defined above is available in the software.

Exercise: The square opening in the wall has a shape that can also be represented by a solid block. Create a Gellish representation of that shape, called ShapeO-101 in its own coordinate system.

5.3 Specification of components

As said before, an assembly should be decomposed until all single components are identified which shape need to be modeled.

For example, tank T-201 has a shape TankShape-201 as is illustrated in Figure 5. That shape is defined by a ShapeModel-201, which is a collection of mathematical functions. That shape model is defined in coordinate system of the tank CoordSys-201. This is specified as follows:

210	Tank-201	1727	has as aspect	233	TankShape-201
233	TankShape-201	4682	is described as	214	ShapeModel-201
214	ShapeModel-201	1876	is defined in coordinate system	201	CoordSys-201

Tank-201 is composed of three components, a wall, a roof and a base plate. The decomposition is specified as follows:

210	Tank-201	1225	is classified as a	520686	cylindrical tank
210	Tank-201	1190	has as part	215	Wall-201
215	Wall-201	1225	is classified as a	40210	wall
210	Tank-201	1190	has as part	216	Roof-201
216	Roof-201	1225	is classified as a	520069	conical roof
210	Tank-201	1190	has as part	217	Bottom-201
217	Bottom-201	1225	is classified as a	520042	base plate

Each of these three components has its own 1D properties. For the example we assume that the properties of the tank are the same as the properties of the wall. Thus:

215	Wall-201	1727	has as aspect	211	D-201
215	Wall-201	1727	has as aspect	212	H-201
215	Wall-201	1727	has as aspect	213	Th-201

Note that the diameter of the tank wall (D-201) is the same property as the diameter of the tank (D-201). These are not two properties, but both objects share the same property (the same UID). So, if the value of that diameter changes, then it changes for both the tank and the wall. The same holds for the height and the wall thickness.

The roof is assumed to have its own different diameter. That diameters can be specified independently, but it is also possible to correlate it to the diameter of the wall and wall thickness of the tank. *(Correlations between parameters is described below)*

5.4 Specification of components with standard 3D shapes

If a shape model of a single component cannot be described as a whole by one of the available standard shape correlations, then the shape has to be decomposed until each partial shape can be described by a standard shape. Assume that the shape of the wall can be represented by one of the standard shapes.

For example: Assume we create a 3D model of the wall. We distinguish between the wall and its shape. The shape of the wall is classified as a hollow cylindrical shape, as is illustrated in Figure 7.

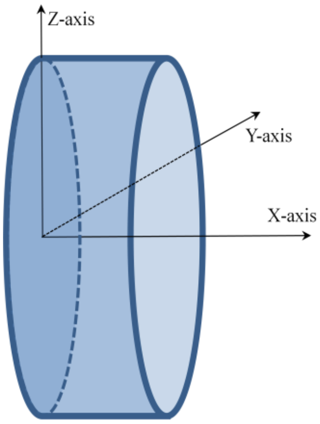

Figure 7, Definition of a hollow cylindrical shape

A hollow cylindrical shape is a standard shape that is available in the Dictionary. An individual shape is related to such a kind of shape by a classification relation as follows:

215 Wall-201	1727 has as aspect	221	Cyl-201
221 Cyl-201	1225 is classified as a	550936	hollow cylindrical shape

Kinds of shapes can be mathematically represented in various ways by mathematical functions. For example as by wire frame models or by 'constructive solid geometry' (CSG) models, which correlations (csg primitives) are specified in ISO 10303-42 and included in the Gellish Dictionary. For example, a hollow cylindrical shape *in general*, can be mathematically represented by the geometric function 'hollow cylinder' that is available in the Gellish Shapes Knowledge Base, where it is specified that:

550936 hollow cylindrical shape	2068	can be mathematically represented by a	911012 hollow cylinder
911012 hollow cylinder	1806	can be described by encoding as	ri**2 =< y**2 + z**2 =< re**2; 0 =< x =< L ; re > ri > 0; L > 0

The formula that describes a basic shape ‘hollow cylinder’ is defined in its own 3D Cartesian coordinate space (x, y, z). The formula requires three parameter values: an external radius (re), an internal radius (ri) and a height (L).

Note 1: Apparently the x-axis indicates the height of the cylinder and the origin (0,0,0) of its coordinate system is on the centre line in the bottom plane.

Note 2: Such a formula shall be implemented in software, for example as a Macro in JAVA code or in C++ code. The software that interprets the Gellish database content shall execute such code, in combination with the parameter values for an individual shape, to generate a presentation of a particular shape on an output device.

This means that an *individual* shape, such as Cyl-201, is described by an individual formula (F-201) that has three parameters. That formula is defined in the individual coordinate system WallCS-201 of Cyl-201. In other words, the shape of the component (Cyl-201) occupies a part of the space that is defined in the coordinate system of that component (WallCS-201).

This can be specified as follows:

221	Cyl-201	5714	is modelled by a correlation as	222	Func-201
222	Func-201	1225	is classified as a	911012	hollow cylinder
222	Func-201	1876	is defined in coordinate system	231	WallCS-201

The last step is the specification of the parameters and the values of the parameters of the correlation.

The parameters that are required for a particular kind of correlation for a standard shape is (normally) defined in a Domain Dictionary. For example, the Gellish Domain Dictionary for Shapes may contain the following specification:

911012	hollow cylinder	5725	has by definition as parameter a	550343	external radius
911012	hollow cylinder	5725	has by definition as parameter a	553155	inside radius
911012	hollow cylinder	5725	has by definition as parameter a	550126	height

This means that the parameters for Fun-201 can be specified as follows:

222	Fun-201	4962	has as parameter	223	Re
222	Fun-201	4962	has as parameter	224	Ri
222	Fun-201	4962	has as parameter	225	L

where

223	Re	1225	is classified as a	550343	external radius
224	Ri	1225	is classified as a	553155	inside radius
225	L	1225	is classified as a	550126	height

Note: As all parameters are classified explicitly it is not necessary to specify the sequence of the parameters, nor group the parameters in a list..

These three 1D parameters of Cyl-201 differ from the 1D dimensional properties of Wall-201. However, their values can be derived from the values of the properties of Wall-201 by applying the following correlations:

Re = D1 / 2
Ri = D1 / 2 – W1
L = W1

Note: These correlations assume that the quantification of the parameters is done with a consistent set of scales (units of measure).

The first of these three correlations is a mathematical law that relates a radius of a circle to its diameter. That law in general can be defined as a (qualified) subtype of a division function as follows:

220	f(radius,diameter)	1726	is a qualification of	911685	division function

whereas it has by definition two parameters: a radius and a diameter. So:

220	f(radius,diameter)	5725	has by definition as parameter a	550681	radius
220	f(radius,diameter)	5725	has by definition as parameter a	550188	diameter

The second correlation is a subtraction function (y = a – b) and the third correlation is an equality function (y = x). The way to specify such correlations in Gellish is illustrated below by the specification of the first correlation.

As the name of the individual function we use preferably the formula itself (as a string of text: “Re=D1/2”). Its parameter are denoted by their individual names as follows:

221	Re = D1 / 2	1225	is classified as a	220	f(radius,diameter)
221	Re = D1 / 2	4962	has as parameter	223	Re
221	Re = D1 / 2	4962	has as parameter	211	D1

The individual parameters are classified already as (subtypes of) radius and diameter, so that software can unambiguously determine which parameter has which role in the correlation and can then calculate the missing value Re from the known value D1.

EXERCISE: Specify the correlations for the parameters Ri and L in a similar way.

6. Placement of component shapes in a 3D coordinate system

The shape of an assembly of components is defined in the own 3D coordinate system of the assembly. Each shape of a component is defined in the own coordinate system of the component, with its own origin and its own orientation. It is possible to place the shapes of the components directly in the 3D coordinate system of the location where the assembly is built, but normally the shapes are first placed in the coordinate system of the shape of the assembly.

For example: Above it is described how standard shapes of simple components, such as Cyl-201, are defined in their own coordinate systems. The next step is that those shapes (or those coordinate systems) need to be placed in the coordinate system of the shape of the assembly. For example, shape Cyl-201 has to be placed in CoordSys-201 of TankShape-201.

> Note: Below it is specified that one shape is placed relative to another shape. However, semantically this means that the coordinate system in which the placed shape is defined is placed relative to the coordinate system in which the reference shape is defined.

This done in Gellish using the placement function as follows:

234	Placement C in T	1225	is classified as a	5718	placement in a 3D space
234	Placement C in T	5722	is a correlation with as reference object	201	CoordSys-201
234	Placement C in T	4886	is a correlation with as subject	231	WallCS-201
234	Placement C in T	5348	has as parameter	235	x1,y1,z1
235	x1,y1,z1	1225	is classified as a	553729	displacement in 3D
235	x1,y1,z1	5025	has on scale a value equal to	236	0, 0, 0
234	Placement C in T	5348	has as parameter	236	a1,b1,c1
237	a1,b1,c1	1225	is classified as a	553727	orientation in 3D
237	a1,b1,c1	5025	has on scale a value equal to	236	0, 0, 0

The displacement (x1, y1, z1) specifies the position of (the origin of) the coordinate system of the placed shape (the subject) in the coordinate system of the reference shape. The orientation (alpha1, beta1, gamma1) specifies the rotation angles of (the coordinate system of) the placed shape with respect to the coordinate system of the reference shape. Alpha is a rotation from y to z around the x-axis, beta is a rotation around the y-axis en gamma is a rotation around the z-axis.

The placement of the WallShape-201 (Figure 7) in the coordinate system of the TankShape-201 (Figure 5) requires a displacement of (0,0,0) and a rotation around the y-axis of pi/2 radians, thus (0,pi/2,0).

The coordinates are in the units of measure that are defined for the whole coordinate system (e.g. mm and rad).

7. Schematic drawings in 2D

Two dimensional drawings can be drawings to scale, such as side views and cross sectional views, or they can be schematic diagrams that use symbols for the representation of objects.

Drawings to scale are modelled in the same way as three dimensional shape models, by ignoring the third dimension, apart from taking into account the visibility and invisibility of shapes. Such drawings should be generated from the 3D model and then extended with annotation as required.

Schematic drawings are modelled differently, because the symbols and other annotation elements are not shapes of represented object, but they are physical objects in their own right, being ink on paper (or equivalent). Such objects, for example lines, have a width, a length, a colour, a shape, etc. Thus, schematic drawings are specified similar to the specification of physical objects that are placed in a 2D coordinate system of the sheet on which the drawing is presented.

Models that represent dumb schematic drawings only consist of annotation elements (symbols, etc.). The essence of ‘intelligent schematic drawing’ models is that those models distinguish between the symbols and the symbolised objects, whereas for each symbol it includes a relation that specifies which object is represented by the symbol. For example, a particular tank symbol and Tank-201 as follows:

210	Tank-201	2072	is graphically represented by	251	Symbol-201
251	Symbol-201	1225	is classified as a	610423	tank symbol

Note that each individual symbol is defined in its own coordinate system.

Individual symbols are created by applying correlation in the form of executable programs (macro’s). Those correlations are typically aggregates of basic correlations (primitives) for basic symbols, such as straight lines, circles, segments, etc. which shapes are defined by mathematical functions. Those correlations are normally parameterised, so that symbols of various sizes can be created at various locations by tuning the correlation by its parameter values. From a user perspective the individual symbols are copied from typical symbols that are standardised and that are included in a symbol library. In reality the typical symbol is itself also created by using the macro with fixed (default) parameter values.

System independent data integration and data exchange requires that the names, functionality and parameters of correlations (executable programs or macro's) are standardised. Typically they will be collected in proprietary, national or international 'symbol libraries' (which are in fact 'software libraries'). Smooth data exchange and data integration is only possible if the symbol libraries are either Open or supplied and available to all involved parties.

For example, assume that a macro is available that defines a standard general tank symbol. The macro is called TankMacro-1. Its correlation is defined in coordinate system TankCS-1 and has as parameters a scaling factor, a colour, line type and line weight. Then TankSymbol-201 is defined and placed on a drawing T-123 with drawing coordinate system T-123CS as follows.

251	Symbol-201	5714	is modelled by a correlation as	252	TankMacro-201	
252	TankMacro-201	1225	is classified as a	4679	correlation	
252	TankMacro-201	1876	is defined in coordinate system	253	SymbolCS-201	
252	TankMacro-201	4962	has as parameter	254	Scale-201	
254	Scale-201	5020	is qualified as	589198	1:1	
252	TankMacro-201	4962	has as parameter	255	LineColour-201	
255	LineColour-201	5020	is qualified as	551759	red	
252	TankMacro-201	4962	has as parameter	256	LineType-201	
256	LineType-201	5020	is qualified as	790383	dashed	
252	TankMacro-201	4962	has as parameter	257	LineWidth-201	
257	LineWidth-201	1225	is classified as a	1660	line width	
257	LineWidth-201	5025	has on scale a value equal to	920538	0.3	mm

Once the symbol and its parameters are defined, the symbol has to be placed in the coordinate system of the drawing. This is specified as follows:

260	Place S-201inT	5714	is classified as a	5717	placement in a 2D space
260	Place S-201inT	5722	is a correlation with as reference object	261	T-1234-CS
260	Place S-201inT	4886	is a correlation with as subject	253	SymbolCS-201
260	Place S-201inT	5348	has as parameter	262	250, 20
262	250, 20	1225	is classified as a	553728	displacement in 2D
260	Place S-201inT	5348	has as parameter	263	0, 0
263	0, 0	1225	is classified as a	553726	orientation in 2D

Note that the units of measure are defined for the reference coordinate system as a whole.

7.1.1 Assemblies of annotation elements

An annotation element, such as a symbol, can be defined as an assembly of other annotation elements. Such a composition hierarchy may continue until the shapes of the component annotation elements can be described by Macro's that are available in a library of executable programs that can present such shapes. A library of basic geometric primitives, such as a line, a triangle, a polyline and a circle, is defined in various standards. The assembly process for symbols is illustrated in Figure 8.

Figure 8, A typical pump symbol

Assume that three macro's are available: circle (radius), equilateral triangle (length) and connect point, whereas the point has no parameters. Each of the shapes is defined in its own 2D Cartesian coordinate system and has its own location of its origin. The shape of the pump symbol as a whole is then defined by creating individual annotation elements, with specified parameter values, followed by the placement of those elements in the coordinate system of the pump symbol. These placements are specified in a similar way as is described for the placement of a 3D object.

8. Auxiliary facts

Each of the facts in Gellish shall have a unique identifier (UID) that is to be included in an additional column in a Gellish Database table. The facts that are illustrated in the examples in this document are called 'main facts'. Each of such main shall be accompanied by a number of auxiliary facts, such as the language in which the fact is expressed, its status, its date of start of validity, author, etc. Those facts shall be recorded in additional columns in a Gellish Database table. Details of the definition of those facts and columns is specified in the 'Gellish Database Definition' document (Ref. 1).

9. References

1. Definition of Gellish Databases – A Universal Data Structure, by dr. ir. Andries van Renssen, http://www.gellish.net/index.php?option=com_docman&task=cat_view&gid=2&Itemid=3.
2. ISO/CD-TS 15926-3
   Life cycle data for process plant: Ontology for geometry and topology.
3. ISO 10303-221
4. IFC
5. IFD-NL
6. http://sourceforge.net/apps/trac/gellish/
7. www.gellish.net
8.

