Confidential

Gellish Modeling Method

Part 1

Architecture

Version 1.0

by

dr. M. van Koetsveld

dr. ir. Andries van Renssen

Table of content

1 Introduction

There is an increasing interest in the use of computer interpretable knowledge. An electronic knowledge library is an extremely practical instrument to make definitions and knowledge available in a structured, standardised and transparent way via computer programs, possibly via the Internet.

There is also a fast growing need to integrate data from different sources by exchanging the data between various parties and by storing the integrated data in general purpose databases. This requires the harmonization and standardization of the data and their definitions, which implies the need for the use of a common terminology and a common way to express information in a system independent way.

The Gellish Modeling Method is an effective and powerful method to address the above-described needs. This document describes the architecture of that method. The method includes the process of development of Knowledge Libraries, which are also called object libraries, and the development of Smart Dictionaries, Taxonomies and Domain Ontologies (in the form of Definition models) that provides a common language for such libraries. The Knowledge Libraries are databases that are composed of collections of Knowledge Models and Requirements Models. The method also includes the usage of such models for computer-assisted design of products and for information storage and retrieval about products and facilities.

This document also provides an introduction into ISO standards based Gellish English as the universal language that is used in the method to define objects and the capture knowledge as well as to describe designs.

This Architecture is part 1 of a series of documents. The following parts are detailed instruction workbooks that provide dedicated guidance for specific tasks in the total process to either create smart dictionaries, to model knowledge or to use knowledge models in design systems and in information storage and retrieval systems.

A library of knowledge models includes common definitions of kinds of objects as well as knowledge about those kind of objects. It is a source for knowledge about kinds of things (classes). Such knowledge can be used, for example, for the design of individual objects, by deriving individual instances from the general knowledge about the kinds of things. It can also be used for guiding the search for knowledge about kinds of things or for documents and other information about individual objects.

Individual organizations or branch organizations can extent the general dictionary and knowledge bases that are available from the support organization to develop their proprietary knowledge bases to support their design processes and to enable storage and retrieval of information and knowledge and to arrange standardized, system independent data and knowledge exchange with their business partners.

Furthermore, the uniformity of Gellish based knowledge bases provides a basis for sharing knowledge in larger communities and for the creation of widely applicable tools that support design, storage, retrieval and exchange processes in those communities.

Using knowledge bases can significantly increase the speed of design processes of individual objects in projects and of the recording of information about those objects during or during operation. It will result in quality improvements so that projects can be executed faster and at lower costs.

An understanding of the Architecture of the Gellish Modeling Method is required to be able to develop a sound dictionary, to create powerful knowledge models and to make optimal use of the available knowledge.

The Architecture is influenced by the ISO based Gellish language that is used to model the knowledge. Therefore, the description of the Architecture starts with a brief introduction into the Gellish English.

After the description of the Architecture, this document describes the various stages in the creation and extension of knowledge libraries through modeling of knowledge and the specification of general requirements according to the Architecture.

Finally, this document describes the usage of knowledge and requirements models in projects and in operational situations, to create designs or to verify designs against the requirements. It also describes the development of Facility Information Models that describe real world facilities, their documentation and the storage, retrieval and exchange of information about them.

To facilitate the understanding and application of the Gellish Modeling Method using Gellish, this document contains a vocabulary with the most important concepts. The used concepts are presented in the form of a Gellish Database table, which is an illustration of an application of Gellish.

Finally, various workbooks belong to the description of the Gellish Modeling Method, in which the process of modeling knowledge and of creating and extending knowledge libraries is presented in detail. For example:

- Part 2 - Creation and use of Smart Domain Dictionaries
- Part 3A – Knowledge Modeling
- Part 3B – Specification of Requirements and Verification of Deliverables
- Part 4 – Facility Information Management – Creation and Use of a Facility Information Models
- Part 5 – Modeling of Activities and Processes

1.1 Structure of this document

This Architecture document is structured as follows.

Chapter 2 provides an introduction in the world of Knowledge Libraries and Knowledge Models in general and provides an outline of the Gellish Modeling Method.

Chapter 3 describes the architecture of the Gellish Modeling Method. An understanding of that architecture is important for a successful creation and usage of object libraries, knowledge models, requirements models and facility information models.

In the modeling process a number of phases can be distinguished, whereas within each phase a number of steps should be followed. Chapter 4 is about those phases. Chapters 5 through 8 provide a stepwise explanation of the modeling method in outline.

Chapter 9 illustrates how knowledge that is modeled and is stored in a knowledge library can be used, for example by a designer or by a cost estimater.

Chapter 10 provides advice about the procedure for the addition of objects to an object library..

The main concepts that are used in this document are included and explained in the vocabulary in chapter 11. That chapter presents the vocabulary in the form of a Gellish Database table.

2 Knowledge Models and Information Models

The objective of the Gellish Modeling Method is to enable computers to interpret knowledge and information in a common language and in a system independent way. This means that computers can use that knowledge to provide computer aided support during design and procurement of facilities and products. or It also means that computers can be used to integrate and manage information from different sources about individual facilities and products.

In this chapter it is explained why it is required to express knowledge about kinds of things in the form of formalized knowledge models and to express information about individual things in information models and what the main characteristics of those models are.

2.1 What is a Knowledge Library?

A knowledge library is a collection of definitions and knowledge about kinds of things that are specified and captured in distinct knowledge models and definition models. A knowledge model (being a model of some knowledge about a kind of thing) is a representation of that knowledge in a formal language, which means that it is expressed in a language in which all the used concepts and relations are defined explicitly and in an unambiguous way, so that the knowledge can be retrieved by using a systematic method.

A definition model is a special kind of knowledge model about a kind of thing that contains only expressions of facts that define that kind of thing and that thus are by definition always true for such a kind of thing.

Those knowledge models and definition models are contained in a database, which therefore can also be called a knowledge base.

The kinds of things in a knowledge library about which knowledge is contained form a kind of framework for further knowledge that is attached to those things.

Such knowledge can be:

- Explicitly modeled knowledge, which can be interpreted by a computer.
- Knowledge that is not explicitly modelled, but is contained in documents and drawings that are not computer interpretable.

The knowledge models may contain knowledge at various levels of explicitation, as is described in chapter 6.

A knowledge library consists of the following parts that are described in more detail later in this document:

- Definition models, each of which expresses a definition of a kind of thing;
- Knowledge models, each of which expresses a particular amount of knowledge about a kind of thing;
- Presentation forms of the kinds of things, such as terms, symbols, names, abbreviations, codes, etc.

Each definition of a kind of things, as well as knowledge about a kind of thing can be contained in a number of facts. In the Gellish Modeling Method those facts are expressed in an unambiguous way in a formalized subset of natural language that is interpretable by a computer as well as by human beings. That language is called the Gellish language. It is further described in chapter 3.2. In the Gellish language each elementary fact is specified as a relationship between two things, whereas each type of relationship is standardized in Gellish. Such a way of specification is sometimes called an object<>relation type<>object method or ORO method.

A knowledge library contains a collection of common definitions for kinds of things that are known and used by a wide community, but it can also contain collections of definitions and knowledge that are specific for particular subject discipline areas. That is the reason why the development of a knowledge library is organized per discipline area and is coordinated by an interdisciplinary body.

To make the knowledge in a knowledge library usable for any kind of user and system it is required that the domain specialists model their knowledge an unambiguous way. Therefore, knowledge library should be tool independent.

Too independency means that the modeling method is not dependent on particular tools (such as particular software or databases) and that the resulting models can be migrated from one tool to the other without the need to make dedicated transformations. In other words, the models need to be captured only once and does not need to be recreated or transformed when in future other systems will be applied.

2.2 What is an Information Model?

An information model is a collection of expressions of facts about one or more individual things that are captured in one or more distinct models as well. An information model (being a model of some information about an individual thing) is an expression of facts in a formal language including also expressions that explicitly classify the individual thing and its components and aspects by kinds of things (concepts) that are defined in definition models.

The expressions that form the information models are also contained in a database.

The information in an information model can be verified on its compliance with a knowledge model or with a requirements model. It is also possible to use the knowledge in a knowledge model or requirements model to create an information model about an individual thing.

2.2.1 Three basic concepts

In the above paragraph we used three terms that denote three basic concepts for the Gellish Modeling Method. Those three concepts need to be clearly understood and are therefore discussed here in more detail.

Kinds of objects

As said, an object library contains definitions and knowledge about kinds of objects. In this context the term ‘objects’ denotes *physical* objects, which are things that obey the laws of physics and that possess aspects, but are not aspects themselves and that can play roles in occurrences. Examples of kinds of (physical) objects are: tree, mountain, country, stream of water, building, road, car, ship, refinery, pump, bolt, etc. Kinds of objects are also called object classes.

Kinds of things or concepts

A knowledge library contains definitions and knowledge about kinds of things, also called concepts or classes. In this context the term ‘thing’ denotes anything, including physical objects, aspects such as properties and qualities, values, units of measures, roles, functions, activities and occurrences, documents, symbols, etc. Kinds of things are also called classes.

Individual things

We use the term individual thing (or individual object) when we denote an example that is of a kind. Individual objects in reality are for example, the Eiffel tower, the M1 motorway, my car and John Doe. An individual object may also be an imaginary individual object. For example, a new motorway that is being designed and is called the A45 motorway. Such an imaginary object 'exists' only in the designer's mind, whereas information about it is stored in a database and in documents. An individual thing has a wider meaning as it also includes individual activities, individual aspects of individual objects, etc.

A design process according the Gellish Modeling Method, for example of a building, a factory or a part thereof, thus results in information models about individual objects. Those individual objects are assembled and thus together they form the designed object. This means that the integrated set of information models of the components of a whole assembly, such as a building or a process plant, form a building information model or a plant information model of the whole object.

Thus an information model contains information about either real world individual objects, or as imaginary individual objects. Each of those individual objects can be related with a kind of object that classifies the individual object. For example, the A45 is classified as a motorway. A classification relation thus indicates that the individual object is classified by a kind of object, whereas it should be the case that a definition of that kind of object is included in the object library. Furthermore, that object library might contain knowledge about the kind of object. For example, it may contain one or more design solutions, or requirements for that kind of object.

Chapter 8.2 discusses in more detail why and how objects and other things are classified.

N.B. It is important to clearly distinguish when is talked about individual things and when we talk about kinds of things. The distinction is fundamental as object libraries are about kinds of things and information models are about individual things, whereas both domains use different relation types to express facts about the things.

2.3 What is the purpose of a Knowledge Library?

A knowledge library is intended to make knowledge accessible in a structured transparant way, using computer programs.

Thus, a knowledge library is a source of knowledge about kinds of things (classes of things), because the knowledge about a kind of things is by definition applicable to every individual thing that can be classified by such a kind. The knowledge that is contained in a knowledge library can therefore be used in various ways. For example, it can be used during the design of individual objects, to verify whether designs are in accordance with general knowledge and specifications of requirements or to provide knowledge for inexperienced users.

In order to express that an individual thing is of a particular kind, the individual thing is classified, which means that we create a classification relation between the individual thing and the kind of thing.

For example:

B-23 is classified as a road

Classification of individual things ensures standardization and eliminates ambiguity within projects and companies. Adherence to those standards causes that everybody uses the same terms for the same concepts.

Figure 1 illustrates how a classification relation for road B-23 provides access to knowledge in a knowledge library.

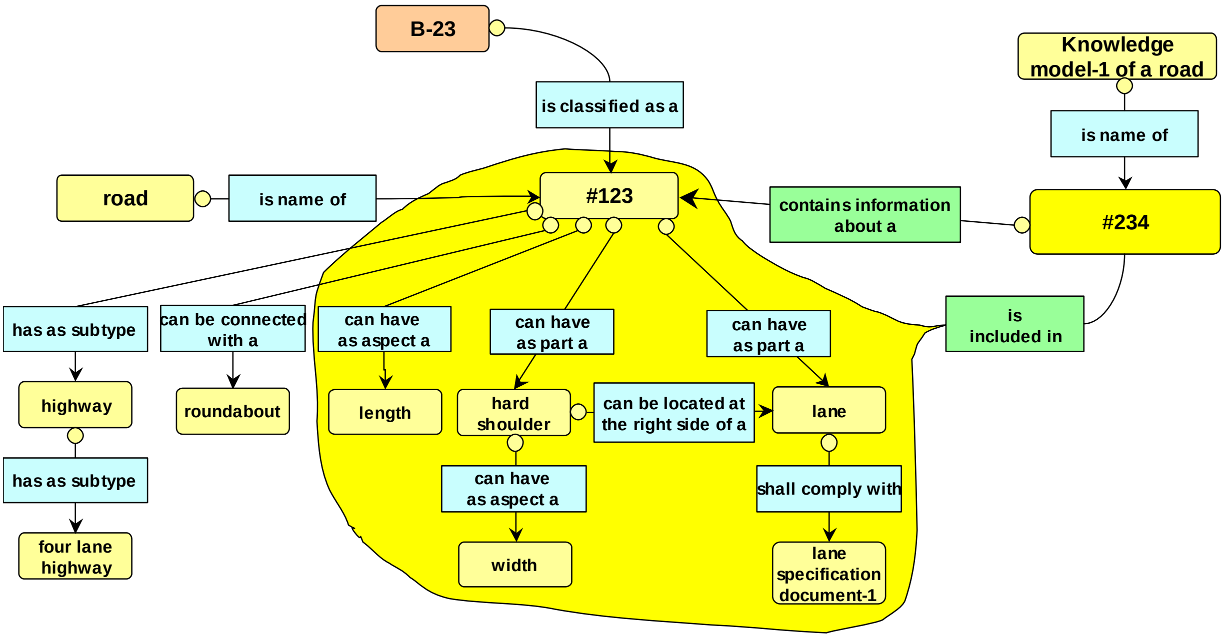

Figure 1, Classification as an access path towards knowledge

In Figure 1 the object B-23 at the top of the figure represents for example a new road that is going to be designed. The fact that B-23 is a road is expressed by the above-mentioned classification relation between the individual object B-23 and the concept #123 that is selected from the Gellish English Dictionary. In that dictionary that concept #123 is named road as is indicated by the relation with the name 'road' at the left hand side of the figure.

The yellow area in Figure 1 indicates a knowledge model, which consists of a collection of facts about the concept road (#123). That the knowledge model is represented itself as collection of facts #234. The box with the phrase 'is included in' represents a number of inclusion relations between the facts and object #234, which relations specify that those facts are included in collection #234, whereas that collection is called 'Knowledge model-1 of a road'. For completeness, it is also specified that the knowledge model #234 contains information about a road (#123).

Other facts, such as the facts that highway is a subtype of road, as is expressed at the left hand side of the figure indicate that there may exist other knowledge about the concept road that does not belong to knowledge model #234.

So, by classifying the individual object B-23 by the concept road that appears in the Gellish English Dictionary as well as in a Knowledge Library, we make the knowledge about that concept available for the design of the individual object (B-23).

The process of classification is explained in further detail in chapter 8.2

2.4 Who will use a Knowledge Library?

The Gellish Modeling Method includes a general and common knowledge base for any user. That knowledge base is formed by the technical dictionary with encyclopedic knowledge about the concepts that are of general interest. Furthermore, there are several domain specific knowledge libraries, which main target users are those who are involved in the subject area that forms the scope of that particular knowledge library. For example, for the soil, road and waterway section the target users are public and private owners, designers, suppliers and constructors of objects about which knowledge is contained. Furthermore there are knowledge providing companies that use knowledge libraries to make their knowledge available and accessible by computers in a practical and efficient way.

2.5 What is the purpose of a Facility Information Model?

The purpose of a facility information model is to provide easy access to data and documents about an individual object (a facility). For example, a facility information model may contain documents and data about a building (then it is called a Building Information Model, BIM), a ship or a process plant. There are three categories of facility information models:

- Document oriented information models
- Data oriented information models
- Combined documents and data oriented information models.

A document oriented facility information model includes a model of the individual object which primarily consists of a breakdown of the facility in parts, parts of parts, etc. and that may contain other relations between the parts, its functions, processes, etc. The model also includes documents and relations between those documents and the components of the facility about which the documents provide information. Such a facility information model forms a sophisticated document management system, that enable either to search directly on document titles or first to navigate through the facility model to search for a component of the facility after which the system displays all the documents about the selected component. Selecting the required document will launch a browser to view the document.

A data oriented facility information model will also include a model of the individual facility. However in addition to that it includes a database with data about the facility and its components, processes, etc. For example, it may contain diameters and lengths of components, operating temperatures of fluids, etc.

A combined facility information model related the facility model with both documents and data.

2.6 The Gellish Modeling Method

The Gellish Modeling Method is intended to provide guidance for the unambiguous development of a knowledge model or a facility information model. With that method it is possible that various specialists, such as cost estimators, specifiers of requirements, designers, manufacturers and operators or maintenance engineers, can capture and add knowledge about the kinds of things within their knowledge domain. The method enables that knowledge from various experts about the same kinds of things can be developed independently and can still be integrated into a single knowledge library.

This document provides an outline of the Gellish Modeling Method in order to give insight in the process to capture knowledge. The three main perspectives are:

1. How is a knowledge library filled?
2. In which way is the knowledge in a knowledge library used in a design project
3. How is a Facility Information Model created and used?

The capturing of knowledge is organized in a systematic way. Therefore, the modeling process is split in a number of phases and each phase is further subdivided in a number of stages.

3 Architecture

To bring consistency in the process to create and use knowledge models and information models it is important to use a systematic common approach. Therefore we first present an overall architecture in which the basic building blocks and their cohesion is explained.

3.1 Outline of the architecture

The architecture of the Gellish Modeling Method consists of four main parts as is illustrated in Figure 2.

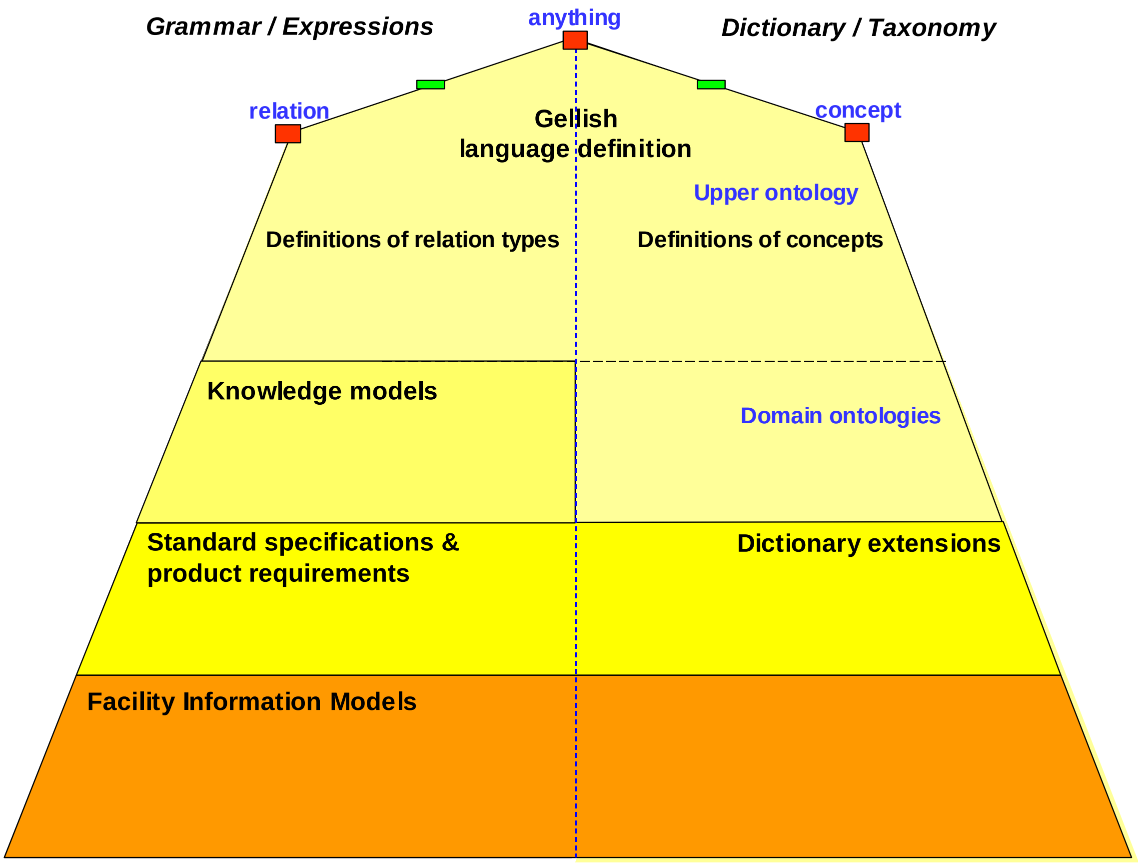

Figure 2, Overall architecture

The four parts that are illustrated in Figure 2 are:

1. The ISO/Gellish Language Definition.

   This includes the definition of the Gellish Grammar and a general purpose Smart English Dictionary / Taxonomy that is based various ISO and other standards[1]. The Gellish Grammar defines how facts can be expressed in Gellish English, Gellish Dutch, etc. In other words, how facts can be expressed. The Smart Dictionary consists of two parts: a definitions part and a presentation part. The definitions part contains definitions of concepts (kinds of things) in the form of definition models. The

[1] Especially concepts that are also defined in ISO 10303-221, ISO 15926, ISO 1998, ISO 14224, IEC 50, ISO/IEC 1131-3, VDI/VDE 3696, ISO 31, ISO 1000, ISO 10303-42, 50, 225, 227, NORSOK and various other standards.

presentation part contains terminology, such as names, abbreviations, symbols, etc. by which the concepts can be presented to people. The Smart Dictionary also contains relations between the concepts and the terms, etc.

2. Knowledge models.

   This is a collection of knowledge models in which knowledge is captured about the kinds of things. This includes also specifications, requirements, standardized designs, etc.

   Note: Models that include requirements, standard design solutions, etc. are sometimes called specification models or requirements models.

3. Standard Specification models and Requirements models

   These are specialized knowledge models that express specifications of standardized objects. For example objects that are standardized in national, international or company standards. These models also include requirements models that express requirements that are valid within a particular context. For example, requirements that are expressed in standards or in best practice guides.

4. Object Information models.

   These are coherent models about individual objects of which each component has a relation to the concept in the Dictionary that classifies that component.

A knowledge library, being essentially a library of knowledge models, in general also includes an extension of the general purpose Gellish Smart Dictionary. This is usually the case because concepts that are specific for the application domain for which a particular knowledge library is meant may not yet appear in the general-purpose dictionary and thus it needs to be extended with definitions of those domain specific concepts.

The following paragraphs provide a clarification of the content of these parts.

A knowledge library, such as a library that is offered by a knowledge provider (for example a branch organization) should not be an isolated body, but should be integrated with the upper part, being the language definition and should make use of the concepts that are defined in the Gellish dictionary and possibly its domain specific extensions. Similarly, standard specifications of components and standard requirements should also use the language definition and may build on knowledge in a knowledge library. The information models (designs as well as realized facilities) should also use the same language definition and may use the knowledge models as well as the standard specification models and standard requirements models, but may also model an individual facility without using those models.

The next chapter describes the various components of the Architecture in more detail.

3.2 ISO/Gellish

As mentioned before, the Gellish Modeling Method uses Gellish as its modeling language. The name Gellish is originally an acronym of ‘Generic Engineering Language’, although the language developed far beyond the engineering domain.

Gellish is derived from natural languages and is based on a number of standards, especially ISO 10303-221 and ISO 15926. The English version of Gellish (Gellish English) includes an English Dictionary with ordinary English terms, synonyms and abbreviations. That Dictionary not only includes definitions of concepts, but also includes definitions of relation types that are used to create expressions of facts. This means that the dictionary complies with the criteria for a Smart Dictionary[2]. The Dutch version of Gellish (Gellish Nederlands) includes a Dutch Dictionary with ordinary Dutch terms, etc.

The usage of natural language concepts and structures means the knowledge and information that is expressed in Gellish English or any other Gellish variant, is hardly sensitive for changes in the information technology. It also means that the semantics are stable and that the expressions can be transformed in a simple way to other or newer technologies. For example, if required it is possible to express Gellish in the form XML or OWL technology.

3.2.1 Language definition (grammar of the language)

The Gellish English language is defined in the Gellish Smart dictionary itself. The definition of concepts in the Gellish English Dictionary is illustrated in Figure 3.

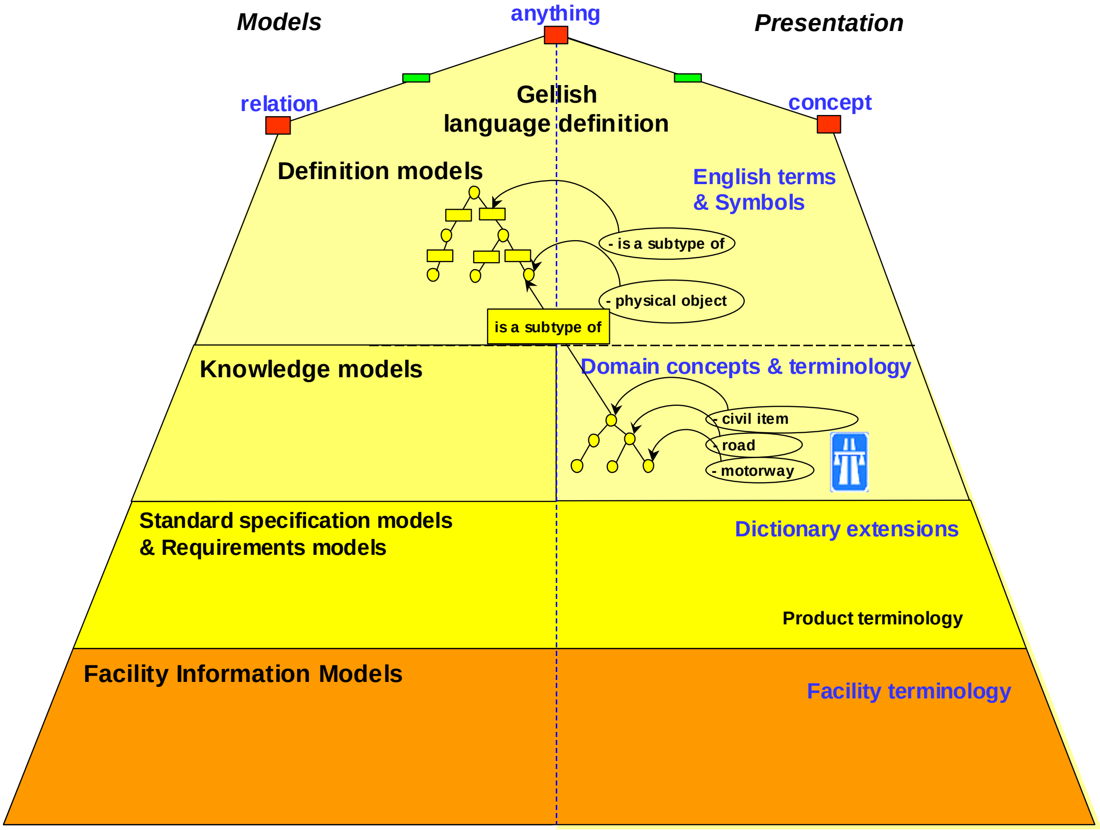

Figure 3, The Gellish language definition part of the Architecture.

A definition model of a concept consists of three parts:

1. A unique identifier (UID) that represents the concept in a language independent way. This is represented in the figure by a small circle. For example, 40153 is the UID of the concept road.
2. One or more terms, synonyms, abbreviations, codes, or symbols that are related to the concept and that form various visual presentations of the concept in a particular language and/or language community. Examples in the figure are the English terms for the object types: physical object, civil item, road and motorway (also presented as a symbol) and the phrase for the relation type <is subtype of>, which is a phrase that is the name (in English) of a specialization relation (subtype-supertype relation).

[2] http://en.wikipedia.org/wiki/Gellish_English_dictionary

3. A collection of definitional facts that specifies the supertype of the defined concept, that provides a textual definition and that may specify the discriminating aspects or components that the defined concept by definition has. The figure illustrates a definition model as a collection of relations between concepts that are represented by lines that connect circles via rectangles, whereas the rectangles represent relation types.

A definition model follows a typical pattern that is described in this part of the architecture. Detailed guidelines for the creation of definition models are given in the document 'Gellish English Dictionary Extension Manual' and detailed instructions are given in the applicable Gellish Dictionary Extension Workbook.

Each fact in the pattern is build-up, as said before, as an object-relation-object expression. This approach is widely known under the term ORO method. Other language, such as OWL, RDF (Notation 3) and IFD (Lexicon) and methods such as ORM are based on this method, although the Gellish language is enriched by the explicit distinction between the relation and the roles that the related objects play in the relation.

3.2.2 Relation types

Pieces of text can be decomposed into atomic expressions. We call this decomposition process explicitation of a text. Each resulting atomic expression should be an expression of an atomic fact. Thus, each expression of a fact according to the basic pattern is represented by a relation between two objects or between two kinds of objects.

As a fact is defined as 'that which is the case', this means that everything 'that is the case' can be expressed according to the basic pattern. Actually the pattern is also suitable to express what is thought to be the case, what is wanted to be the case or for a question whether something is the case.

The concepts in the Gellish Dictionary as well as the standard relation types are required to specify facts with which new concepts are defined and added to the dictionary. Additional concepts and relation types are available to express knowledge and requirements about kinds of things and again other relation types contain the semantics that is required to express information about designs and real world individual objects or to relate objects to information objects, such as documents and drawings.

For example: the fact that a road can have as part a lane, is expressed according to the basic pattern in Gellish English as follows:

road <can have as part a> lane

whereas the brackets are added to illustrate that the phrase that represents a part-whole relation type is one unity. This example illustrates that the expression according to the basic pattern is an ordinary English sentence. It is simple and unambiguously interpretable, not only for human beings, but also for computers. As you will reckon this expression is also a correct Gellish English expression, although the expression is usually stored in a Gellish Database table as follows:

Name of left hand object	Name of relation type	Name of right hand object
road	can have as part a	lane

Other examples of similar expressions of facts are:

viaduct	<is a subtype of>	civil item
viaduct	<can have as part a>	pillar
viaduct	<has by definition as part a>	bridge deck
pillar	<can have as aspect a>	height.

Examples that express facts about individual things are:

V-1	<is classified as a>	viaduct	
Pillar-1	<is a part of>	V-1	
Pillar-1	<has as aspect a>	Height of Pillar-1	
Height of Pillar-1	<is quantified on scale as>	4.3	m

The parts of the sentences between brackets are called relation types, because they denote the kind of relation between the related objects.

These relation types are used to express knowledge, requirements and information such as expressed in knowledge libraries and in product models or facility information models.

By expressing facts according to the same structure, using the same standard relation types and using the same concepts and definitions from the electronic Gellish Dictionary it become possible to integrate knowledge and information from different sources in one consists language. This enables computers to search for knowledge and information in various databases, combine the search results and present the integrated results to a user. It also enables that the knowledge and information is interpreted and used by computers.

This prevents problems that occur with many search engines, such as the fact that a query many irrelevant search results are presented. It also significantly increases the chance that you find what you are looking for, whereas conventional representations of information use a large variety of terminology that is often not mentioned in the search criteria.

3.2.3 The smart Gellish Dictionary

The Gellish Dictionary includes not only concepts and their names and synonyms, but also symbols, illustrations, codes, abbreviations, etc.

A part of the Gellish English Dictionary is shared with other standards, especially the ISO 15926-4 standard, which is also called a Reference Data Library.

The Gellish Dictionary is a ‘Smart Dictionary’ because of the following characteristics:

- It distinguishes between a concept and the terms with which the concept can be denoted. Therefore, it provides definitions of concepts. When another concept exists that is denoted by the same term (a homonym), then that other concept will have a different unique identification. In that respect it differs from a conventional dictionary, because a conventional dictionary is based on words (term), whereas for each term there are given several definitions, whereas the user iss left with the task to determine whether the definitions are valid for the same concept or for different concept. Each concept is represented in Gellish by a different unique identifier (UID). This is further explained in paragraph 3.2.4.
- The concepts are arranged according to a particular hierarchy. That hierarchy expresses a specialization of the concepts in the form of a tree-structure: a

specialization tree, which is also called a subtype-supertype hierarchy.[3] This hierarchy can be used by computers for the inheritance of aspects from a supertype concept to all its subtype concepts. It can also be used to facilitate searching for things of a particular kind, because the computer can then also find things that are classified by a subtype of that kind.

Every concept in the dictionary a subtype of a more generic concept. At the very top of the hierarchy there is only one concept that covers anything and is thus called 'anything'. This demonstrates that the Gellish language covers in principle everything, as any concept can be defined as a subtype of a concept that already exists in the language.

Lower in the hierarchy the more specialized concepts are located, such as the concepts that are defined in specialized knowledge libraries (object libraries) and in various standards. At the lowest level of the hierarchy there are concepts located that represent standardized types of products, such as product catalogue items and manufacturer's models. The latter ones are also subtypes of more general concepts.

- It includes not only definitions of concepts, but also definitions of relation types. Those relation types determine the expression capabilities of the Gellish language.
- The Gellish Dictionary includes also relations between concepts that define the concepts and thus are by definition true. This means that knowledge about the concepts is included in a computer interpretable form.

A large number of definitions and terms that are considered international common concepts and terms are already included in the Gellish Dictionary (mainly in English). That also holds for the definitions and terms for concepts in many specialist disciplines, among others in engineering.

The Gellish language is an open language and anybody can extent the Gellish Dictionary with an unlimited number of definitions of subtypes of concepts and relation types from his or her own discipline on any level in the hierarchy. However, users should be aware that the Gellish definitions of their concepts should be communicated with others or be proposed as extensions of the official Gellish Dictionary.

This means that developers of knowledge libraries or users of the Gellish language don't need to be concerned whether their concepts already exists in the dictionary. If not, then they can simply be added by defining them as a subtype of an existing more general concept in the Gellish Dictionary. This also holds for the standardized relation types, which can be used for the effective recording of knowledge about kinds of things or for the description or specification of individual things or their designs. Also standardized allowed values for characteristics and scales (units of measure) that are included in the Gellish Dictionary.

3.2.4 Distinction between the identities and the names of things.

Every individual thing and every concept is represented in Gellish by a unique number, which identifies the thing uniquely and unambiguously. This holds for physical objects, concepts and knowledge models, but also for all other things, such as facts and relation types.

These unique numeric codes are called globally unique identifiers (UID's) and they are compliant with an ISO standard for identification of objects (ISO/IEC 11179-5).

---

[3] Strictly speaking it is not a tree structure, because the hierarchy includes not only multiple subtypes, but also multiple supertypes. This means that it is a hierarchical network structure.

A Gellish UID does not contain any coded information and is language independent. That is why a kind of thing always has the same UID, even in different languages.[4]

Everything, kind of thing or relation type is denoted in addition by one or more names, symbols, synonyms or codes in various languages and language communities, which are the contexts in which those denotations are used. The fact that the things can be denoted and facts can be expressed in a particular language, but are represented in Gellish by language independent UID's, enables computers to automatically translate facts to various other languages and vice versa.

For example:

The relation type with UID 1146 is denoted in Dutch as <is een soort> and in English as <is a subtype of> or <is a specialisation of>. For a computer only the number 1146 is of importance, whereas for a person who defines or uses a dictionary or library the terms are essential.

The definitions and terms used in several other standards, such as ISO 15926-4, are available as synonyms (with those standards as language community context). Similar synonym relations are made for terms and definitions in proprietary database implementations, such as for various SAP implementations. This forms the basis for communication (exchange of data) with such systems in Gellish.

3.3 Domain dictionaries - Gellish dictionary extensions

A knowledge model often requires the definition of additional concepts that do not exist yet in the Gellish Dictionary. Such extensions of the Gellish Dictionary may even result is complete Domain Dictionaries. Those extension fit in the architecture just below the language definition part as a natural extension that adds 'definition models' in the same way as ordinary Gellish concepts are defined.

Such dictionary extensions consist of Definition Models and a presentation part with terminology and possibly symbols, etc.

A Gellish Definition Model is a model that consists of a coherent collection of facts about the defined object that are always valid for every example of that kind in the real world.

3.3.1 Definition models

Definition Models may be models about definitions of any kind of concept, such as:

- kinds of physical objects;
- kinds of activities (for example: kinds of testing or maintenance activities);
- kinds of functions and other kinds of roles;
- kinds of aspects, such as characteristics, properties, qualities, etc.;
- allowed values for aspects of a kind.

All these kinds together are called concepts.

[4] The only things that have no unique identifier are the terms, because as character strings they are already unique from themselves. For example, the term 'bank' can be used to denote different concepts, even in different languages. Nevertheless the character string b-a-n-k is a unique and unambiguous sequence of four characters from the Latin alphabet.

Every definition model of a concept consists of relations between the defined concept and other concepts and possibly relations between those other concepts that express what is *by definition* the case for aspects and parts of the defined concept.

For example:

The facts that define what the concept viaduct is, we should only take into account the facts that are always by definition true for any viaduct, whatever subtype of viaduct it is. So those facts are valid for all things that may be classified as a viaduct. This means that a definition model does not cover facts that are sometimes true, under some conditions only. The latter kinds of facts are part of a knowledge model about what can be the case for some viaducts.

The concepts that are defined through definition models are an extension of the tree structure (specialisation tree) of concepts in the Gellish Dictionary. So together with the concepts that are already defined in the Gellish Dictionary they should form one consistent subtype-supertype hierarchy.

3.3.2 The presentation part

The presentation part of a dictionary extension or domain dictionary specifies how concepts are denoted or called. As described before, those denotations include names, synonyms, abbreviations, codes and symbols. For short, together these denotations are called terms. The presentation part includes relations of special kinds between the unique identifier of the concept and the terms by which it can be denoted in various contexts. For example, this is expressed in a Gellish Database table as follows:

Language	Language community	UID of left hand object	Name of relation type	Name of right hand object
English	engineering	130370	is called	pump
English	Company X	130370	is abbreviated as	P

Definition models and their presentation parts can be defined as public or proprietary extensions of the general Gellish Dictionary. The concepts that are defined in that way can be used in the same way as all other concepts that exist in the Gellish language. This is also the case when the extensions are stored in a separate database on another location. For users of the Gellish language who have access to the extensions, those extensions (when defined according to the rules of the Gellish Modeling Method) and the general Gellish Dictionary together behave as if it is one dictionary.

3.4 Knowledge models

The next part of the architecture is a part that consists of knowledge models.

Each concept in the Gellish Dictionary and its extensions can behave as a clamp to which knowledge can be attached in the form of one or more knowledge models.

A knowledge library is a database of knowledge models. Organisations may develop such knowledge libraries and offer services to extract knowledge models from the library for inclusion in application systems or as a guide to develop or design individual objects.

3.4.1 What is a knowledge model?

A knowledge model about a particular kind of thing is a collection of facts that may be the case for individual things that are of that particular kind. Those facts are expressed in Gellish as relations between concepts that already exist in the Gellish Dictionary or its extension. There are special relation types available in Gellish to express facts that describe such knowledge. A knowledge model is always also related to the particular kind of thing about which it provides information and that should also already exists in the dictionary or in the used dictionary extension.

So, knowledge models do not repeat the facts that are by definition always true, but it adds additional facts that are possible, but that are not always in all circumstances true.

In addition to that it is possible to express knowledge about requirements and constraints that are valid for individual things that are of a particular kind, which requirements and constraints are only valid in a particular context. These category of facts belong to models that are called requirements models or standard specification models, which are described in the next part of the architecture.

So, a knowledge model about a kind of thing can be regarded to be a network of expressions of facts (relations between concepts), each of which relation expresses a small piece of knowledge. The network of facts expresses explicit knowledge. Knowledge can originate from any kinds of sources and can even be expressed in natural language expressions. Such expressions can be small sentences or larger pieces of text (including also complete documents, drawings, multi-media files, etc.) that then are not modelled (not made explicit). Nevertheless it is possible to include such 'information objects' in Gellish knowledge models. However, normally those 'information objects' are not computer interpretable. Every knowledge model has a specific scope: it does not need to include all knowledge that is available about a particular kind of thing. There may exist various knowledge models about the same kind of thing. For example, each with knowledge collected from a different perspective. Those knowledge models may also include partly the same facts, so that they have overlapping scope, whereas a Gellish enabled software may be able to integrate such models and present the combined knowledge in an integrated way.

The facts in a knowledge model generally include the following kinds of relations:

- a relation between the knowledge model and the concept about which the model expresses knowledge;
- relations between concepts, one of which is the concept about which the knowledge is expressed;
- relations between a concept and an information object;
- relations between the model and the facts that are included in the knowledge model (that belong to its scope).

So a knowledge model forms a 'cloud' of relations in a bigger network, whereas the scope of the knowledge model (the inclusion relations) determine the cut by which the model can be separated from the total network of relations between concepts.

Note that the knowledge in a knowledge model is additional relative to the definition model. That means: the knowledge model complements the definitions with additional encyclopaedic knowledge, but has no impact on the definition of the concept! This means that a library of knowledge models (a knowledge library) in fact can be called a modelled encyclopaedia that contains more knowledge about the concepts than a dictionary. The concept in such an encyclopaedia does not provide an extended definition (or narrower definitions) than the dictionary.

For example:

A dictionary model expresses what a viaduct is. It contains only criteria that can be used to decide whether an object can be classified as a viaduct or not.

A knowledge model of a viaduct will contain a lot of generally valid knowledge about viaducts, but the definition of how a viaduct is defined does not change because of all that knowledge.

Standardised types (Specification object types)

In many cases organisations want to include specifications in knowledge models, such as typical compositions, standard requirements and typical costs figures. However, these requirements do not include generally valid knowledge about kinds of things, but they belong to the context dependent requirements and specifications that are discussed in the next part of the Architecture.

3.4.2 The general structure of a knowledge model

A knowledge model about a kind of physical object usually specifies the kind of components from which such a kind of object can be composed or typically is composed and also what kind of properties it can have or typically has and in which kind of activities or occurrences the kind of object can be involved or typically is involved.

This means that a knowledge model typically consists of the following kinds of relations:

- Conceptual composition relations. For example:
  car <can have as part a> wheel
- Conceptual possession of aspect relations. For example:
  car <can have as aspect a> colour
- Conceptual involvement in activity relations. For example:
  car <can be a subject in a> collision test

Furthermore the kind of object can also have relations with documents or kinds of documents, organisations, etc.

Knowledge models can contain knowledge from different perspectives. Therefore we can distinguish for example:

- Knowledge models that describe many possible compositions with a large variety of options for characteristics, possible specifications and possible requirements and constraints. Such knowledge models will be of value for trainees and experts in different disciplines. For example, for cost estimation, design, etc.
- Knowledge models that specify knowledge about particular subtypes. They are of value once subtypes are already selected, but a further specification of details is required. Therefore these knowledge models are about subtypes of the above mentioned concepts and contain more details. These models can therefore be called specification models.
- Domain specific knowledge models, which contain knowledge for a particular application domain. For example, reference models for cost estimation. Such models

may contain for example knowledge about the kinds of aspects that are relevant for the estimation of the costs involved when an object of such a type is designed, manufactured, installed, etc. Other examples are function oriented models and activity or process models.

A model library may also distinguish between models that are specifically intended for particular departments (organisation specific model), models that are discipline specific and models that are project specific.

This means that models may become very specific and its applicability may be limited to a particular context within which the model is made. Therefore it is important to pay special attention to the management of models in a model library.

3.4.3 Composition tree

Typical or possible compositions of kinds of objects are often presented as a tree structure: a composition tree. Such a composition tree (also called a composition hierarchy) is fundamentally different from a specialisation tree (a subtype-supertype hierarchy) as described above where the definition model was discussed.

A composition hierarchy describes the kinds of components from which a kind of thing can be composed and from which sub-components those components can be composed, etc.

An example of a decomposition tree of a potable water system can be as follows:

- potable water system
    - water treatment plant
        - filter
        - water storage tank
    - water distribution system
        - pump station
        - pipe line

This decomposition hierarchy can be expressed in Gellish as follows:

potable water system	<can have as part a>	water treatment plant
water treatment plant	<can have as part a>	filter
water treatment plant	<can have as part a>	water storage tank
potable water system	<can have as part a>	water distribution system
water distribution system	<can have as part a>	pump station
water distribution system	<can have as part a>	pipe line

3.5 Standard specification models and requirements models

The next part of the Architecture deals with standard specification models and requirements models. Standard specification models are models that reflect a standardised specification for a kind of thing. For example, standardisation organisations have published standards for a large number of standardised components. For example, standard nuts and bolts, standard structural steel beams, piping components, etc., etc. Also manufacturers publish their standard catalogue items, which are types of products that comply with a standard specification that is defined by the manufacturer. When such standards are applicable it means that a product that complies with the standard shall have the aspects that are defined in the standard. It is not optional, but required within that context. Finally, this kind of models may also be defined within organisations or projects to specify typical solutions and best practices.

Standard specification models are requirements models are therefore models in which the typical or possible compositions and aspects are replaced by required compositions and required aspects. In other words, the <can be> and <can have> relations are replaced by <shall be> and <shall have> relations. These kinds of models then reflect typical or standard design solutions or best practices and thus express requirements that are only valid in the context of a particular organisation that expressed those requirements or in the context of a particular standard that is applicable because an organisation declared it applicable. These requirements do not relate to generally valid knowledge about kinds of things, but are context dependent requirements and specifications for kinds of things.

In most cases those specifications are only valid for a particular subtype of a concept, but it may also be the case that specifications are only valid for a kind of thing in a particular kind of role. An example of the latter is when viaducts are distinguished because some rules apply when a viaduct is managed by a local government, whereas other rules apply when a viaduct of the same type is managed by the central government. In this case there are no really different types of viaducts, because for the same viaduct the management may change. This means that the difference is determined by the relation between the viaduct and the kind of managing party, whereas it remains the same type of viaduct. This means that some rules apply for a viaduct in the role of being managed by a local government, whereas other rules apply for a viaduct that is managed by a central party. In this case there are no subtypes of viaduct, but two kinds of roles: locally managed viaduct and centrally managed viaduct.

These models can be highly specialised and nevertheless they may still allow for variations and options. For example, a Volvo S40 and a V70 are types of cars, which models are specified to a high degree, but nevertheless there are still variations and options available.

A standard specification model can be used several times to derive individual objects from the specification. The whole purpose of such a kind of model is usually to be reused for the design or creation of individual objects. This may be only within a particular project, within a company or within a whole branch of industry.

Standard specification models are thus about knowledge that is context specific, which means that the knowledge is only applicable within the context of a particular project, company or when a standard is declared to be applicable (e.g. because of government regulations).

3.5.1 Relations between concepts and documents

Documents as well as models are 'information objects' that can have various kinds of presentation formats. For example the same drawing can be presented on paper, it can be presented in a Powerpoint file and it can be presented in a pdf-file. This illustrates why we should distinguish between the information (a document as an information object) and the various physical carriers of such information. Examples of kinds of information are: user manual, drawing, schema, 3-D model, knowledge model, etc. Examples of information carriers are: book, report, XML-file, DOC-file, MS-Access database, or even a Gellish database table in Excel.

The consequence of this distinction is that in a Gellish knowledge model or in a requirements model we should first relate a thing to a document (the information) and then relate the information to the physical information carrier. For example as follows:

road type ABC	<shall be compliant with>	specification document ABC
specification document ABC	<is presented on>	ABC.doc

3.6 Facility information models

The next part of the Architecture consists of facility information models.

Facility information models are models with facts about *individual* things, such as individual buildings, roads, process plants, ships, etc. They can be imaginary objects, such as designs that only exist in the minds of the designers, or realised objects from reality that may be natural object or that may be artefacts, which result from manufacturing and construction activities.

For example, during a capital investment project it may be that an information model is built that contains part or all of the information about the facility that is designed and possibly also measured data about the equipment that is installed. Such a project model then contains facts about individual objects. For example, the fact that road B23 has a length of 23.4 km or the facts that road B23 is described on drawing T-12345, which drawing is presented on file F-123.dwg and as well as on F-123.pdf. In tabular Gellish form:

B23	is classified as a	road	
B23	has as aspect	L23	
L23	is classified as a	length	
L23	has on scale a value of	23.4	km
B23	is described on	T-12345	
T-12345	is presented on	F-123.dwg	
T-12345	is presented on	F-123.pdf	

The above information means that a search for B23 will result in three relations: a relation that classifies B23 as a road, a relation to the length of the road and a relation to the drawing, which points to two files, whereas the search engine can launch an application to visualise the content of either one or the other file on a screen or print the drawing on paper.

To achieve that the various parties that participate in a project use unambiguous terminology, it is important that in their systems the same language is used. That can be achieved by specifying that the individual objects and all their aspects and relations shall be classified by well defined concepts, such as the concepts that are defined in the Gellish Dictionary or a possible company specific or domain specific extension of that dictionary. As a result a search for things that are classified for example as a vessel, will result in a list of all vessels (including objects that are classified as subtypes of vessel), from which list the required vessel can be selected.

A common dictionary and the classification relations that relate the individual objects to the concepts in that common dictionary are therefore essential parts of those information models.

The usage of standardised allowed values for characteristics and scales (units of measure) is also of high importance for the unambiguous exchange of information about individual objects and for the integration of data from different sources.

An individual object in the facility information model can also be related to a knowledge model or the standard specification model and requirements model, either because it is derived from that model, or it is compliant with that model, or because the individual model is generalised to become a standard solution within the project or company.

For example, if a local government uses the specification of a sewage as the basis for a standard solution of sewages, then the product model of that individual sewage can be generalised to become a standard specification models for sewage type A, whereas that model can be included in a model library as a standard solution for wider usage in other projects.

4 Phases in the modeling of knowledge

The Gellish Modeling Method includes a staged approach to assist domain experts to specify knowledge in a structured way. That approach consists of a number of stages within the following phases:

1. **Create definitions**

   The kinds of things within the knowledge domain of an expert get a definition in the form of a definition model that consists of a fixed pattern of Gellish relation types.

2. **Add knowledge**

   Knowledge about these kinds of things is documented in the form of knowledge models. That is also done by using expressions using a limited set of predefined relation types.

3. **Add requirements**

   Requirements for kinds of things are added in the form of requirements models. These models also use special kinds of relation types.

The addition of knowledge may include standard specifications that represent standard design solutions, or typical solutions for which cost figures are available, or information objects, such as documents and drawings. Each kind of knowledge requires its own set of relation types. It is also possible that requirements are modelled in various ways. For example, models may contain requirements at a qualitative level, whereas it is also possible that models contain requirements that are quantitative.

5 The definition of concepts

To include new concepts in an extension of the Gellish Dictionary, it is required to first define those concepts according to the rules for a proper definition. This means that new concepts should be arranged at the proper place in the specialization hierarchy, as a subtype of one or more existing direct supertype concepts. In addition to that it should be defined which aspects and/or components distinguish the defined subtype from the concept that is its direct supertype and from the concepts that are also a subtype of that supertype (its neighbors). Those aspects and components are called the discriminators.

Figure 4 illustrates some definition models that consist of a model part and a presentation part with terms from the Gellish Dictionary and an extension with the concept called motorway, which is also denoted by a symbol. The model shows for example that a road is a subtype of a civil item, which is a subtype of physical objects, etc. It also illustrates that the concepts are related by other defining relation types with other concepts from the dictionary or its extension.

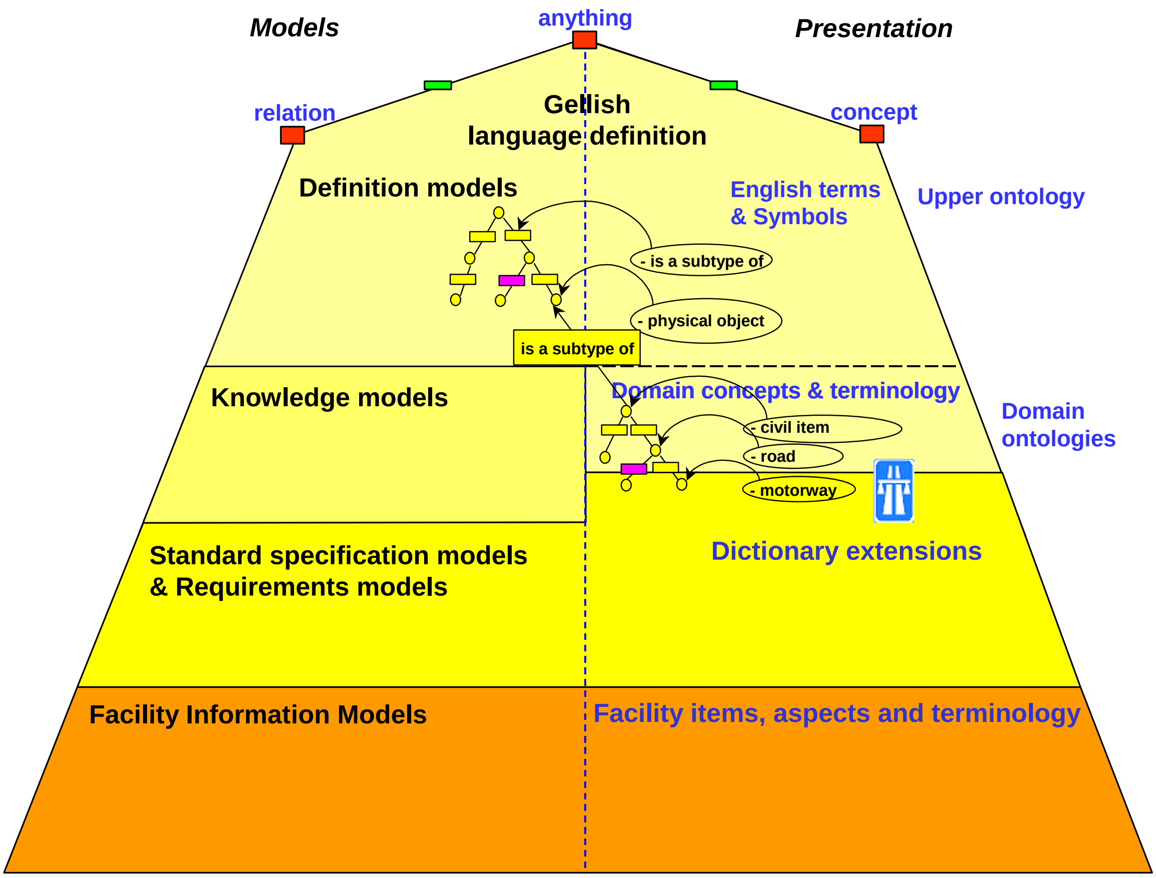

Figure 4, A definition model in Gellish

It is important to make a clear distinction between facts that are part of a definition model and facts that belong to knowledge models, because people who use the dictionary should not be confronted with facts that are not by definition the case. The creation of knowledge models is described in the next chapter.

Facts that determine the definition of a kind of thing are always context-independent and are always valid. Those defining facts are therefore not company specific (except for proprietary inventions) and not dependent on design choices.

On the other hand, facts that express specifications or requirements are by definition only in the context in which they are specified and are thus context sensitive. For example, the fact that a bridge may have pillars does not belong to the definition of a bridge in general, because there are bridges without pillars.

5.1 The stage-wise creation of a definition model

The general structure of a definition model of a physical object is presented in Figure 5.

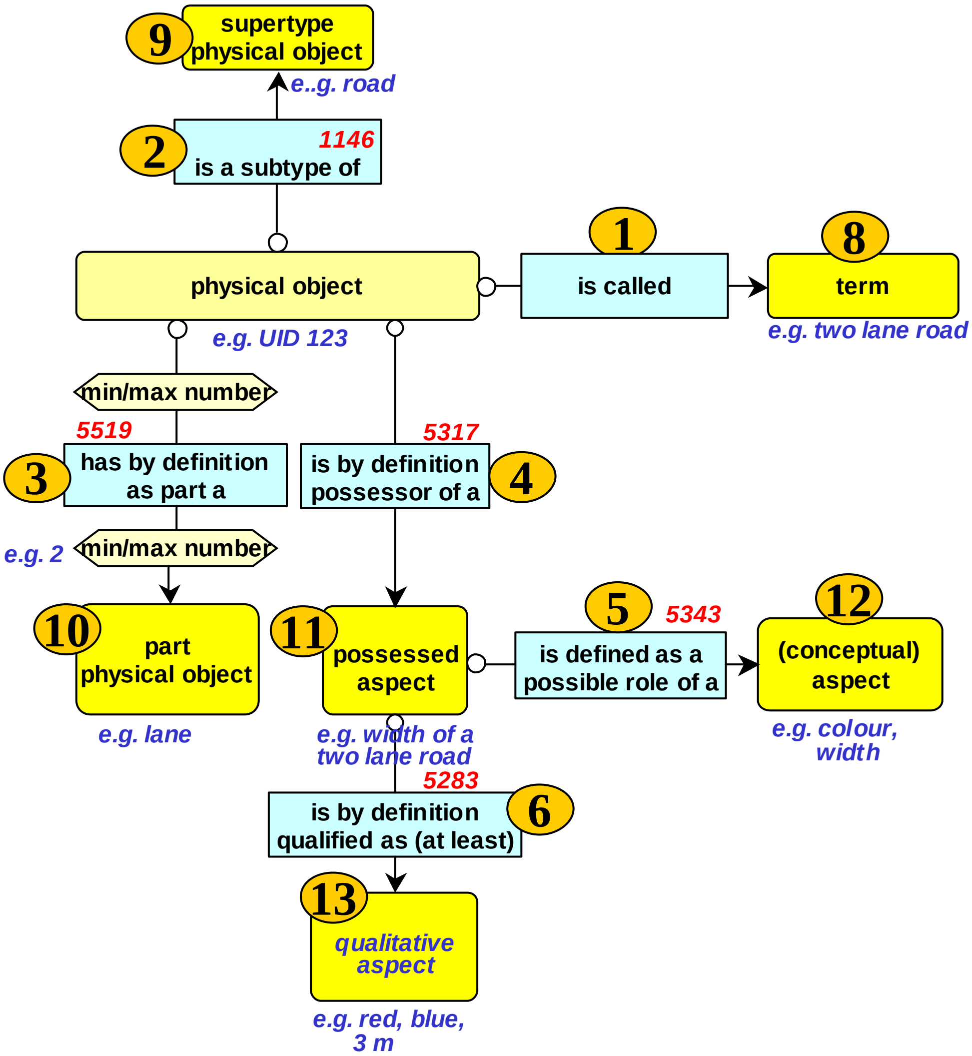

Figure 5, Definition model of a physical object

The figure shows how a physical object can be defined by various kinds of relations with various other concepts. For example, the concept with UID 123 has relations with the concepts 8 till 11, whereas concept 11 is defined by two relations with the concepts 12 and 13.

Those relations are called defining relations. They are of the kinds 1 till 6 in Figure 5.

This means that for the creation of a definition model for a physical object, the next six steps should be followed:

1. Allocate one or more names, synonyms, codes and/or abbreviations to the physical object.
2. Define the concept as being a specialisation (subtype) of one or more earlier defined supertype concepts.
3. Specify from which kinds of components the concept is by definition assembled (if applicable).[5]
4. Specify which aspects, such as characteristics, properties, qualities or functions, are by definition possessed by the concept.
5. Define of that aspect what its nature is.
6. Define of that aspect what its qualitative value is.

Detailed instructions for the creation of these facts using an accompanying template are given in the Workbook for the creation of definitions that belong to the Gellish Modeling Method.

5.1.1 Elucidation of stage 1: Naming

Each concept shall have at least one name. Each individual thing usually has also a name, although Gellish allows that individual things do not have a name, but are denoted via other related objects. The allocation of names to things is formally specified by a naming relation between the UID that represents the thing and a character string that is a name for the thing. Such a name is allocated in a particular language and also within a language community that is the origin of the name. For example, the concept with UID 990152 has in English the name bank, whereas that name originates from the business language community. Names shall be unique within the language and language community context. So in English there may be another concept that is also called bank, but that second name must have its origin in another language community. Within the English language the usage of the same term for multiple things are called homonyms.

The prime name for a concept is allocated in Gellish together with the specialization relation (see next paragraph). The prime name for an individual thing is allocated in Gellish together with the classification relation For example, the name bank is allocated to concept 990152 and the name of the Westminster bank is allocated to its UID as follows in a Gellish Database table:

Language	Language community	UID of left hand object	Name of left hand object	Name of relation type	UID of right hand object	Name of right hand objects
English	business	990152	bank	is a specialization of	990008	organization
English	business	123456789	Westminster bank	is classified as a	990152	bank

Concepts may have more than one name. Such other names are called synonyms of the first name. Codes, abbreviations, translations, etc. are treated in the same way as names. They also originate from a language and language community context. The only exceptions are international terms such as codes (‘symbols’) for units of measure, numbers, etc. In Gellish

[5] Formally speaking: provide a specification of kinds of components when individual things of this kind are by definition composed of individual components of such kinds.

the language context is then called International. The following Gellish Database table presents some examples to allocate names, synonyms, abbreviations, codes and translated names to concepts.

Language	Language community	UID of left hand object	Name of left hand object	Name of relation type	UID of right hand object	Name of right hand objects
English	materials engineering	990152	water	is a specialization of	430143	inorganic substance
International	chemical engineering	990152	H2O	is a code for	990152	water
German	materials engineering	990152	Wasser	is a translation of	990152	water
English	electrical engineering	430465	alternating current	is a synonym of	430465	alternating electric current
English	electrical engineering	430465	AC	is an abbreviation of	430465	alternating current

In addition to the allocation of textual names, etc. it is possible to denote concepts by means of graphical objects, such as symbols, drawings or pictograms. Those graphical objects are information objects that are presented as (typical individual) physical objects ("ink on paper") that may be modeled, or that may be referenced, for example as an electronic file in some native format. The way in which that is done is described in the Gellish Modeling Method part 2.

5.1.2 Elucidation of stage 2: Specialise

Each concept that is added to a domain dictionary shall be defined as a specialization of a more general concept that is also included in the domain dictionary or in the Gellish Dictionary. This is illustrated with relation number 2 in Figure 5. Also the hierarchy in Figure 3 illustrates a number of specialization relations (subtype-supertype relations) that form a taxonomy of defined objects, such as motorway, road, civil item, physical object, etc.

The specification of a specialisation relation means that the defined concept is a further specialisation (subtype) of the earlier defined supertype concept. That supertype concept shall be either exists already in the Gellish Dictionary or one of its supertypes shall exist already in the Gellish Dictionary.

As all concepts are specialisations of more generic concepts, this means that they are all arranged in a kind of tree structure that gives insight in how they all relate to each other[6]. The further specialized concepts are located at further branches in the tree structure. This is illustrated on the vertical axis of Figure 6: the concepts that are located lower on the vertical axis are more specialized.

[6] As a concept may have more than one supertype (and will have more than one subtype) it is not really a tree structure, but a hierarchal network structure.

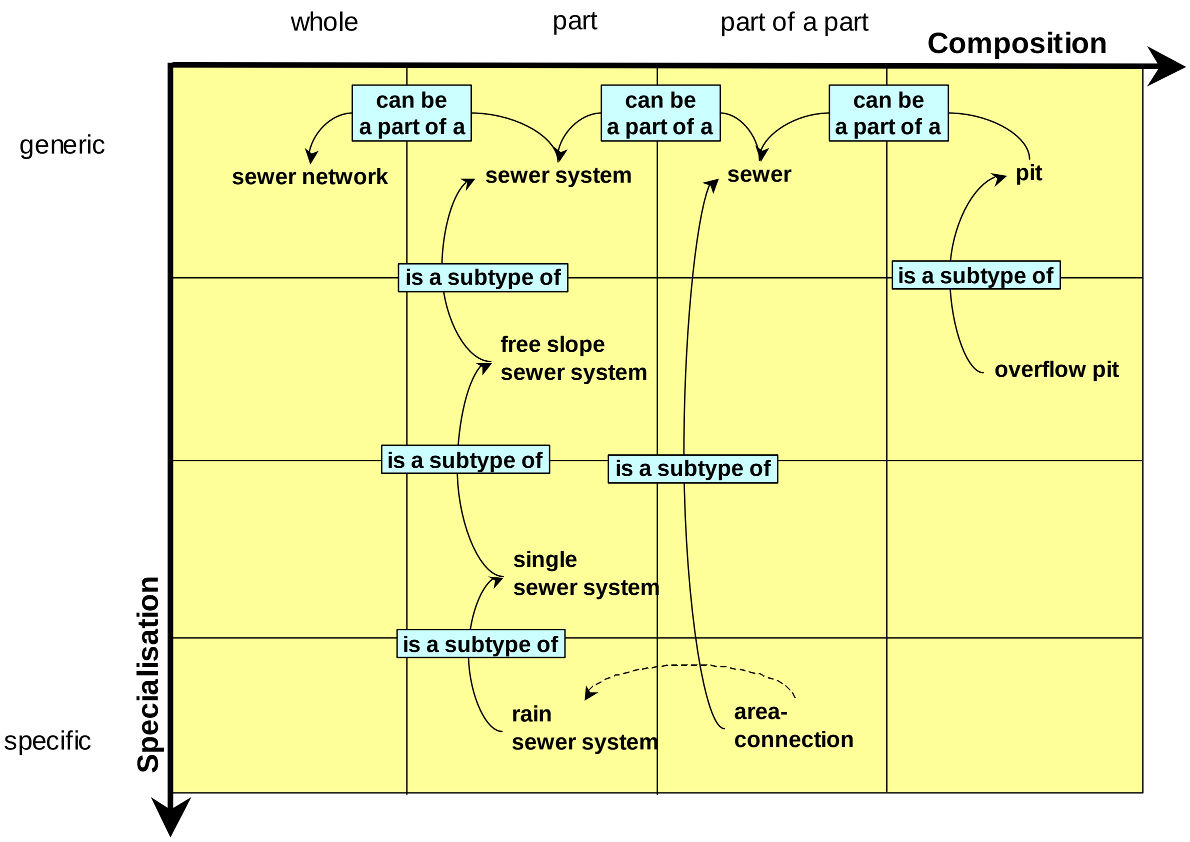

Figure 6, Basic structure for object definitions: specialisation and composition

As is mentioned earlier, the hierarchical structure of specializations of concepts in a dictionary is called a specialization hierarchy or specialization tree.

When all concepts in a particular application domain are added to the domain dictionary, then all the objects that are used in projects in that domain can be classified by the concepts in the dictionary. This means that the project uses a common dictionary with unambiguous definitions for all parties that participate in the project. Furthermore those concepts can be used to find all objects of the same kind and they can be used to attach and find knowledge about those concepts.

5.1.2.1 The inheritance principle

A specialization relation between concepts implies that everything that is the case for the supertype concept is also the case for the subtype concept. For example, the fact that a car is a specialization of a vehicle means that everything that is the case for the concept vehicle is also the case for the concept car. This implies that a more specialised (subtype) concept "inherits" all knowledge about its supertype concept that is located at the higher level in the specialization hierarchy. Because that is the case for each concept in the hierarchy, this means that a concept inherits all aspects of all concepts that are arranged higher in the specialization hierarchy. In other words, it inherits everything from all its supertypes (all its generalizations).

Note that this inheritance mechanism only operates via the specialization relations. Knowledge is not inherited via composition relations. So, if a component is a part of an assembly, then what is true for an assembly is not inherited to the component.

For example, a path inherits knowledge that is available about the concept road, because a path is a subtype of road. Further specializations of path are foot path and bike path. This means that both inherit knowledge from path as well as from road.

So, a concept inherits always via the specialization relations. In other words, via the <is a specialization of> relations.

5.1.2.2 Definitions and qualification of aspects

An aspect is phenomenon that describes something about a physical object or activity but is not the object itself. For example, length, pressure, colour, material of construction, duration, etc. Aspects are defined largely in the same way as physical objects. They also have one or more names and are defined as subtypes of higher level aspects. However they don't have parts and don't possess other aspects, although aspects may be correlated.

Aspects can be distinguished in conceptual aspects and qualitative aspects. A qualitative aspect is a numeric or textually defined value of an aspect. The values: '3 m', red, high, stainless steel, 5 seconds, are examples of qualitative aspects that are qualifications of the conceptual aspects, length, colour, etc.

Qualitative aspects are defined by a special kind of specialization relation, called a qualification relation. So, instead of an <is a specialization of> relation a qualitative aspect is defined for example as follows:

red   is a qualification of   colour

5.1.3 Elucidation of stage 3: Add defining composition

Often a concept is by definition composed of particular kinds of components. Then it is required to specify such composition relations with its parts.

For composition relations we distinguish between parts from which the assembly is *by definition* composed and parts that are optional. If a concept has by definition (always) a particular kind of object as part, then that is a defining fact. Such a fact is expressed by a defining composition relation and that fact then belongs to the definition model of that concept.

This is expressed as follows:

car   <has by definition as part a>   wheel

as is illustrated by relation number 3 in Figure 5.

Note: Optional composition relations (denoted by the phrase <can have as part a>) or composition relations that are required by a particular organisation (denoted by the phrase <shall have as part a>) do not belong to a definition model, but respectively to a knowledge model and a requirements model.

Cardinalities

The example above that specifies that "a car has by definition as part a wheel" illustrates that it is required to specify also the minimum and maximum number of parts of the same kind that are by definition (at the same time) part of the assembly. These numbers are called the minimum and maximum (simultaneous) cardinalities. If the cardinalities are not specified then the allowed number of parts is greater than zero, but further undefined. In this example, a car shall have at least three wheels, but the maximum number is unspecified. This is specified as (3, n).

5.1.4 The distinction between specialization relations and composition relations.

When modeling knowledge it is not as easy as it seems to be at first sight to make the proper distinction between specialisation relations and composition relations (stage 2 and stage 3).

It is required to understand the route that shall be followed to model the available knowledge in a logical and systematic way. Starting from the definition of an assembly concept at a generic level, specifying its subtypes as well as its kinds of parts and the kinds of parts of its subtypes, up to the specification of the parts of its parts and the subtypes of its parts. In other words, the route from generic to specific and from assemblies to parts.

The structure of the method to model the knowledge from generic to specific, as described above, is depicted in the example of the sewer system in Figure 7. It should be note however that the composition relations in that example are optional compositions as in a knowledge model and they are not defining relations as in a definition model of a concept.

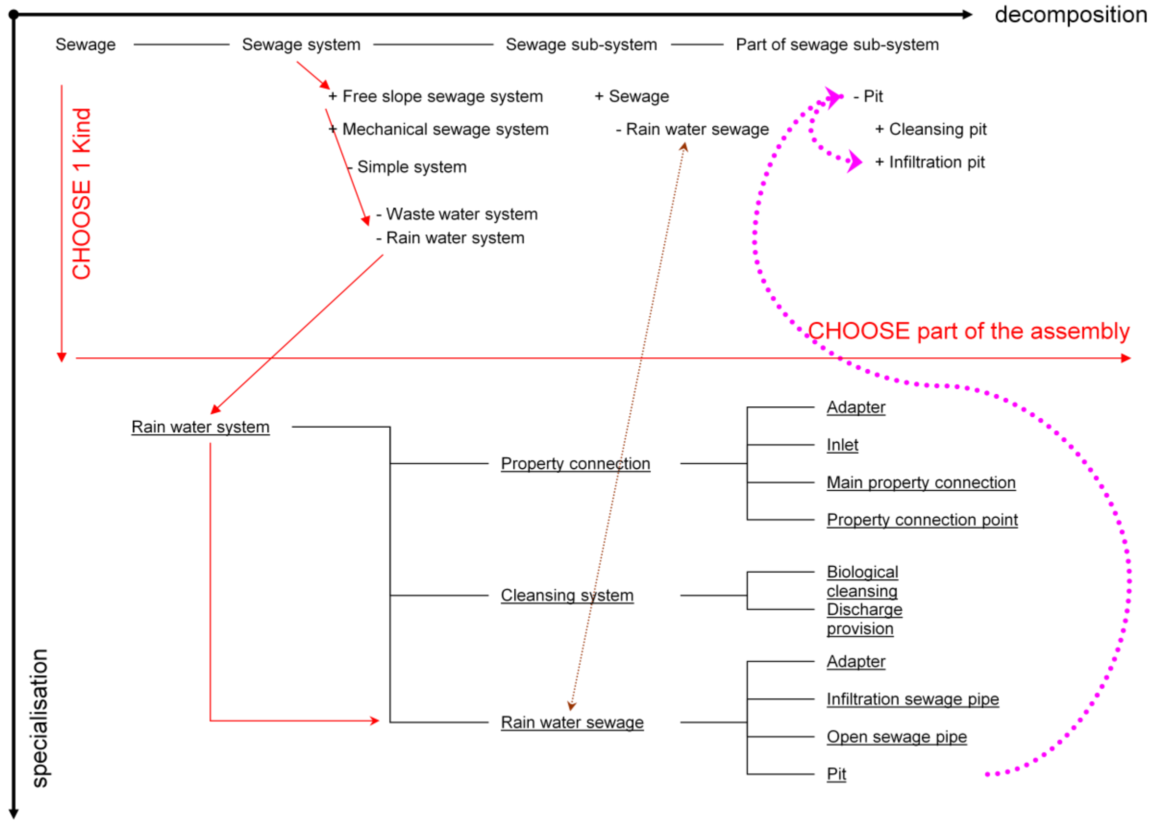

Figure 7, Composition of a sewer system

Vertical arrows in Figure 7 indicate specialization relations and horizontal arrows indicate composition relations.

Examples of the composition relations are expressed as follows:

a sewer network <can have as part a> (1,n) sewer system

a sewer system <can have as part a> (1,n) sewer subsystem

a sewer subsystem <can have as part a> (1,n) sewer subsystem component

On each level in the composition hierarchy, such as sewer network, sewer system, sewer subsystem and sewer subsystem component, it is possible to define specializations (subtypes). The framework for knowledge about sewer networks is therefore created through the alternating specification of specialization relations and composition relations. This shall be done as follows:

1. Start with a *kind of thing* at the highest assembly (compositions) level. (In the example, we don't have subtypes of sewer network).
2. Specify from which kinds of *components* such a kind of assembly can be composed.
3. Specify for each of those components which kinds of *subtypes* exist.
4. Specify for each kind of subtype of which kind of *components* it can be composed.
5. Etcetera.

5.1.5 Elucidation of stage 4: Add defining possessed aspects

The next stage implies the specification of the defining aspects, by which the concept is further defined.

In a similar way as with the components we make a distinction between defining aspects and optional aspects. Defining aspects of a kind of object are aspects that are always possessed and by such an object and always have the defined value if we talk about objects of such a kind.

Thus, a definition of a concept includes only defining aspects. Those aspects are called discriminating aspects, because they are the discriminators that help to distinguish between the objects that they define. The definition model of a concept is extended with expressions of the possession of such defining aspects. This is illustrated with relation 4 in Figure 5. An example of such a specification is Gellish is:

reciprocating pump	is by definition	possessor of a	operation principle of a reciprocating pump

After the specification of the defining aspects it is possible that further knowledge about the concept is added by the specification of additional relations, using other relation types.

5.1.6 Elucidation of stage 5: Define the nature of the possessed aspect

A discriminating (defining) possessed aspect of a concept shall be defined by specifying what its nature is. Each discriminating possessed aspect is a role of an aspect and is therefore by definition defined as a possible role of a particular kind of aspect. For example, the 'with of a two lane road' is by definition a possible role of a width.

5.1.7 Elucidation of stage 6: Define the value of the possessed aspect

The value of a discriminating aspect shall also be defined. This is done by specifying that the possessed aspect is by definition qualified by a qualitative aspect.

A qualitative aspects is an aspect that is qualified by a value. For example '3 meter' is a qualitative aspect with a value 3 on a meter scale. Such a qualitative aspect is a qualification of the general aspect length. Length is therefore also called an example of a conceptual aspect.

Another example is the colour red. Red is a qualitative aspect of the conceptual aspect colour. So qualitative aspects may be quantifications on a scale or may be qualified by textual 'values'. A number of qualitative aspects that are qualifications of the same conceptual aspect are often collected in collections of allowed values, such as in pick lists in user interfaces of software systems.

The Gellish Dictionary already contains a large number of qualitative aspects, but nevertheless it will be frequently required to define additional qualitative aspects.

So, the definition of a qualitative aspect implies that the aspect value shall be defined as a qualification of a more general aspect. Examples of such definitions are:

reciprocating	is a qualification of	operation principle
3 m	is a qualification of	distance

5.1.8 Example of a definition of the concept two lane road

Figure 8 is an illustration of an example of a definition model for the concept 'two lane road' with as UID 123.

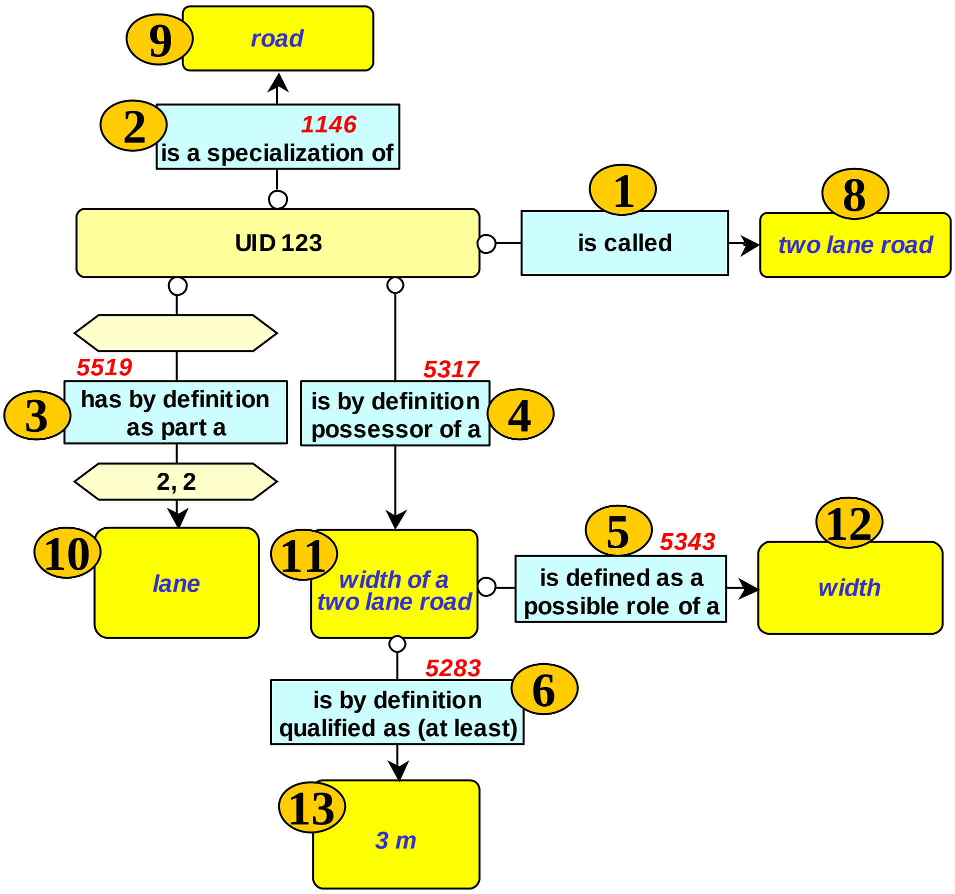

Figure 8, Definition model of the concept two lane road

The concept two lane road in Figure 8 is defined trough the following steps:

1. the concept is given a name: two lane road (fact 1)
2. it is defined that the concept is a specialization of the concept road (fact 2)
3. it is defined that this kind of road has by definition two lanes (fact 3)
4. it is defined that this kind of road has by definition an aspect called 'width of a two lane road'

5. for that aspect it is defined that its nature is that it is by definition a possible role of a width.(fact 5)
6. for that aspect it is defined that is by definition qualified as at least 3 m (fact 6)

Note that it is assumed that the concepts 9, 10, 12 and 13 already exist in the Gellish Dictionary. If that is not the case, then they shall be defined by a similar procedure.

The recording of the above six facts in a Gellish Database table is the way in which the concept two lane road is included in the domain Dictionary.

5.2 Usage of templates

Definitions and knowledge can be created directly in Gellish, but there is also a simplified method available that can be used without in-depth knowledge of the Gellish language. The latter method is illustrated in Figure 9.

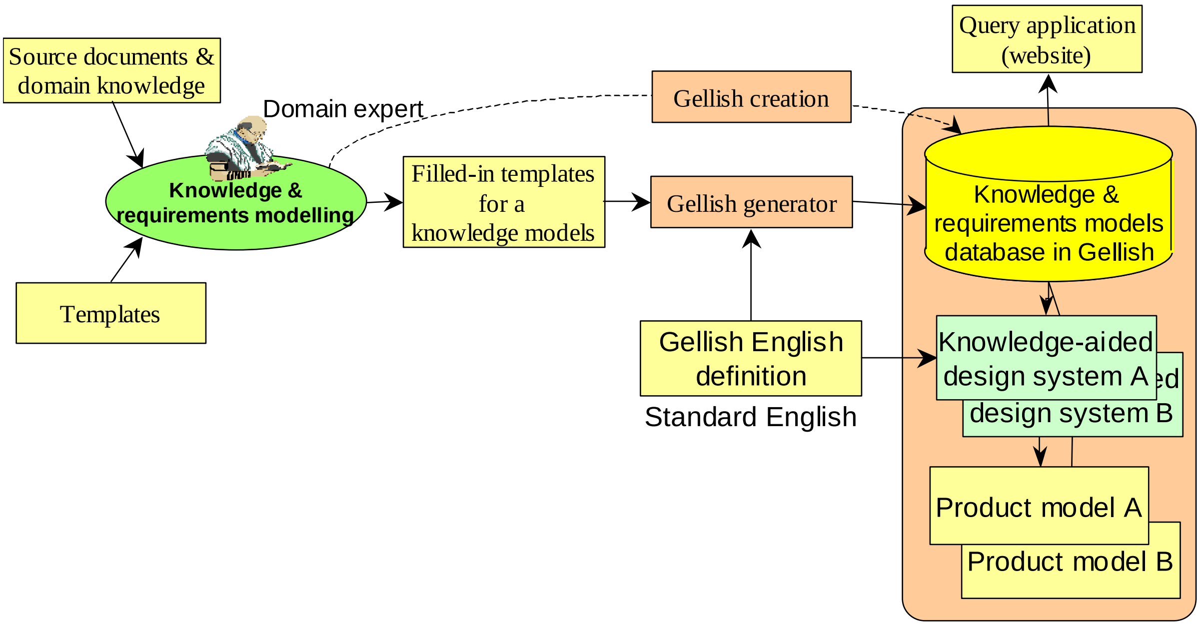

Figure 9, The process to create and use a definition model or a knowledge model

The simplified method provides the discipline specialist with a template with fill-in-the-blanks options as an aid to model his discipline domain knowledge. Such a template is always the same, it is independent of the subject that is modelled, whether it is domain knowledge about a road, a bridge, a sewage or process equipment: the method to document definitions and knowledge is always the same.

Each of the earlier described stages in the modeling process results in models that are characteristic for that stage. Therefore, each stage requires its own templates. This means that there are definition templates and various kinds of knowledge templates, as well as requirements templates. When a domain expert has completed filling-in some templates, then a Gellish generator is used to create a database with knowledge models. That generator is a computer program that is able to interpret the template and that uses the Gellish language definition to create a correct Gellish database.

The stages in the Gellish Modeling Method with the accompanying templates guide the discipline specialist through the various parts of the architecture while modeling his or her knowledge about the kinds of objects.

In this way it is relatively simple to develop usable knowledge models about kinds of objects that belong to the knowledge domain of the specialist, without the need for the specialist to become a specialist in Gellish.

A template has the form of a table with predefined columns, as is illustrated in Figure 10.

Template for the definition of a physical object

Context:	[Select a context]
Top model:	
Start date:	20-2-2006
Latest update:	11-7-2006
Status:	proposed
Author:	Gellish-extractor
Organisation:	Gellish@Work
Language:	English
Remarks:	

Object	**Level 1**	**Level 2**	**Level 3**	**Level 4**	**Type of discriminator**	**Value of discriminator**
sewage sub-system						
	sewage				application (sub-system)	Transport of sewage water
		sewage water			technology (system)	Transport by gravitation
					kind of sewage water	Waste water
					kind of sewage water	Miscellaneous sewage water
		rain water			technology (system)	Transport by gravitation
					application (sewage)	Absorb of water
					kind of sewage water	Rain water
	connection				application (sub-system)	Supply of sewage water
		property connection			application (connection)	Supply of waste water and/or rain water from a property
		pot hole			application (connection)	Supply of downflow rain water from a pot hole
	provision				application (sub-system)	Collect, salvage, cleaning and drainage of sewage water
		over flow provision			application (provision)	Salvage and drainage of sewage water
		infiltration provision			application (provision)	Collection and release of rain water
			pumping-station			

Figure 10, Template for the definition of a kind of physical object

Figure 10 is an example of a specialization tree template. It illustrates how a specialisation tree (subtype-supertype hierarchy) is specified by filling-in a template. The tree should be read from left to right to go from subtype to supertype. The content of this template will result in Gellish expressions, such as:

waste water sewage   is a specialization of   sewage

sewage   is a specialization of   sewage subsystem.

Another template is used to specify a composition hierarchy (also called an assembly structure or an asset breakdown) of a kind of object. For example, when a road needs to be described, then the composition hierarchy indicates the kinds of components from which a road is composed. Such is template is intended to create part-whole relations, or in other words, it results in expressions that use the Gellish relation type ‘can have as part a’. For example in the fact:

road   can have as part a   lane

The next stage includes that the discriminators need to be determined. This means that the aspects or defining components that determine the definition of each kind of object should be specified. A template to specify those discriminators will result in Gellish expressions, such as:

bridge   has by definition as part a   road surface

pipe   has by definition as aspect a   internal diameter

6 Creation of knowledge models

The third part of the Architecture of the Gellish Modeling Method is the creation of knowledge models and their inclusion in a knowledge library or object library. This part is concerned with knowledge about kinds of things that are defined already by using the definition models that are described in the previous chapter. This means that this part is concerned with additional knowledge that enrich the knowledge that is contained in the definitions of the concepts, but that do not change those definitions.

Knowledge libraries contain knowledge models that rely on and extend the concepts that are defined with the definition models.

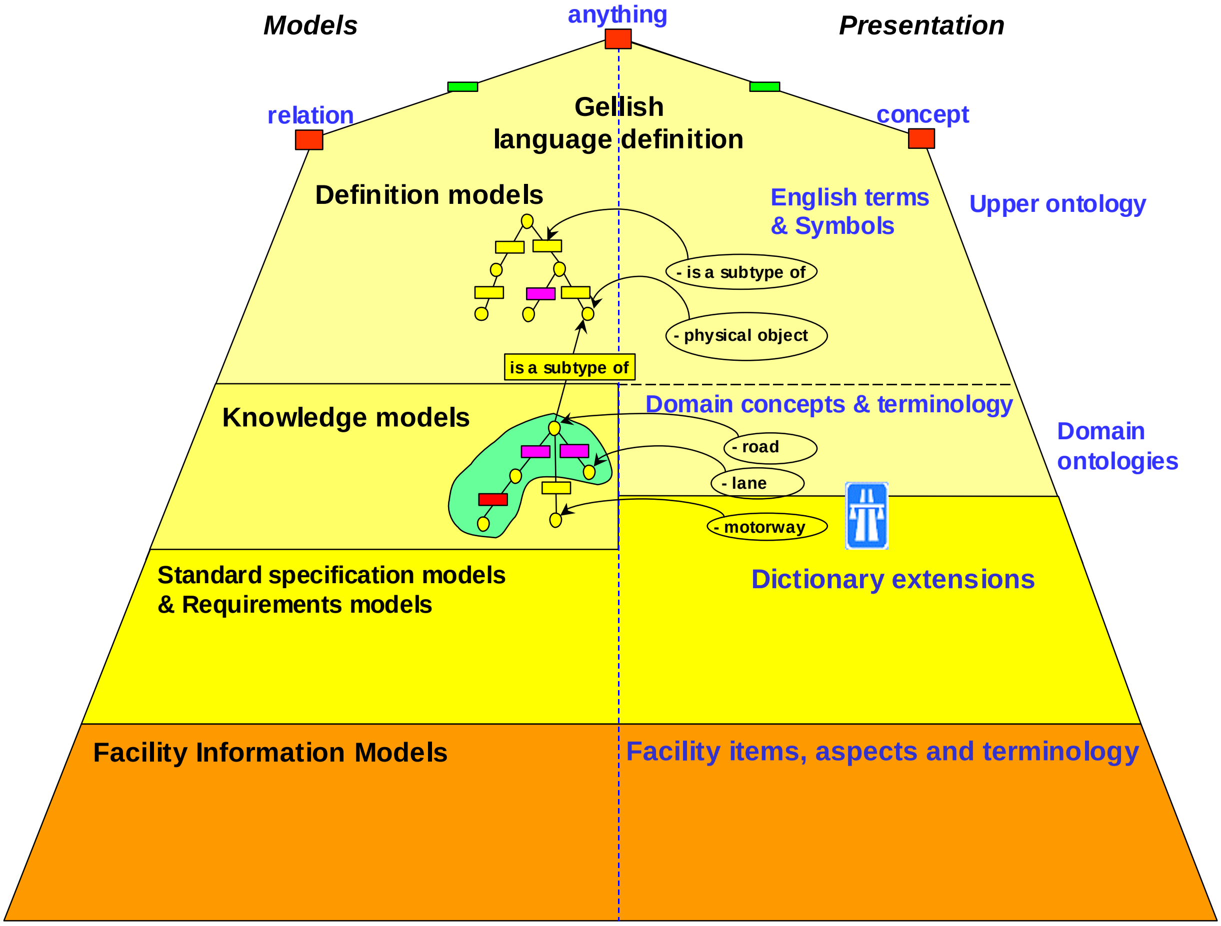

Figure 11, Knowledge models related to Definition models

Figure 11 illustrates that a knowledge model about a road (indicated by the collection of facts in the green 'cloud') consists of a cloud around the concept road that itself is in the dictionary and thus is an element in a definition model. So by a search on the concept road it is possible to find one or more knowledge models about a road. In that way the knowledge about a road is disclosed.

The same is the case for example for a motorway.

Furthermore, a concept inherits the knowledge that is available for its supertypes. So the concept motorway inherits the knowledge about roads in general, which makes that knowledge also applicable for usage, for example for the design of a motorway.

The same holds for any other kind of object in a knowledge library. For example, pumps, vessels, trains, cars, sewer systems, etc. and all their components. All the subtypes of such kinds of objects have their own knowledge models and inherit the more general knowledge from their supertypes.

So, in the Gellish Modeling Method the knowledge that is expressed for the supertype kinds of objects does not need to be captured again for each of their subtype kinds of objects.

This means that the specialization hierarchy of concepts with their definitions act as connectors for knowledge models, as coat hangers for coats, whereas the concepts can be used to select the knowledge models for usage for example during a design process.

Each application area has its own needs for knowledge. For example, people who formulate specifications of requirements or create facility information models or cost estimators, all have their own knowledge requirements. So each of them may need other knowledge models about the same kind of object for their own purpose.

6.1.1 Example of a knowledge model

A collection of facts that is applicable for example for the concept road is called a knowledge model of a road. This is illustrated in Figure 12.

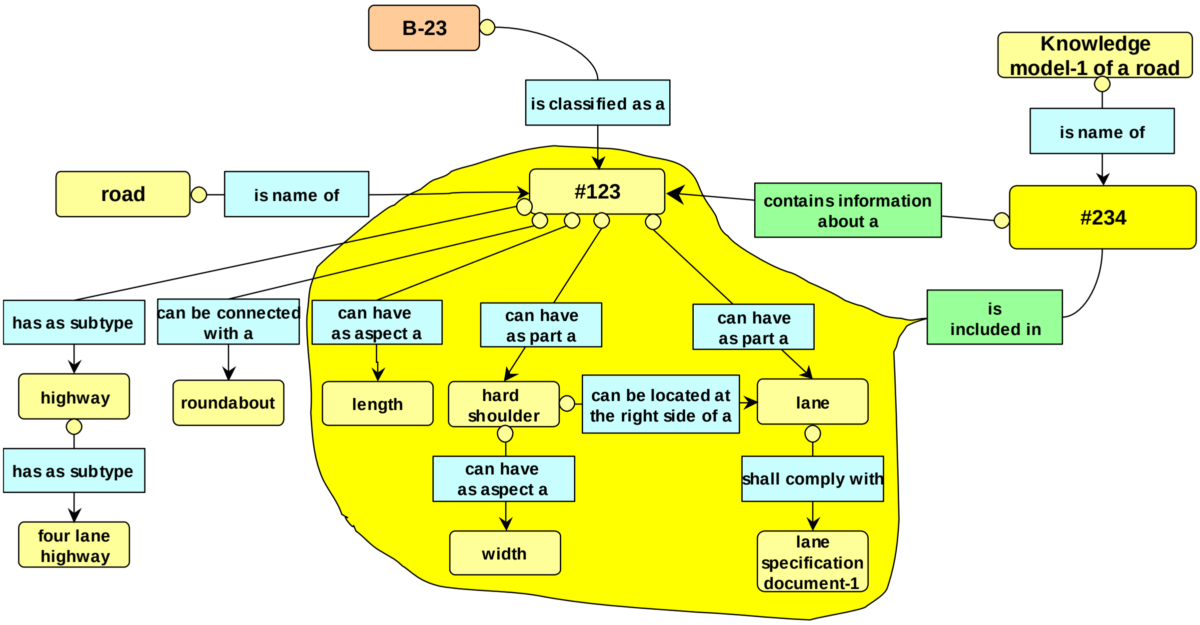

Figure 12, Example of a knowledge model of a road

The collection of facts that describe knowledge about a motorway in general is called a knowledge model of a motorway. The collection of facts that describe requirements that must be satisfied by a motorway is called a requirements model of a motorway. The requirements models are discussed in the next chapter.

Once a knowledge model is used it is also possible to zoom-in either on knowledge models about one or more subtypes or on knowledge models about the kinds of components from which such a kind of object can be composed, such as lanes, entries, exits, viaducts, roundabouts, etcetera.

6.2 Stepwise creation of knowledge models

In principle, knowledge models consist of many kinds of facts. The most important types are facts about compositions and facts about the possession of aspects (characteristics, properties, qualities and performance). Others may concern with relations with activities in which they may be involved (such as testing or functions they may fulfill), fluids that they may contain or organizations that may supply them, etc.

This means that the creation of a knowledge model consists of the following three main steps:

1. Specify possible compositions
2. Specify possible aspects
3. Specify possible relations with other kinds of objects

6.2.1 Elucidation of step 1: Specify possible compositions

It may be that objects of a particular kind are by definition composed of a particular kind of component. In that case such a composition fact is a defining composition that should already be included in the definition model of the concept.

Apart from these defining components there are kinds of objects that may or may not be a part of a whole assembly of a kind. These are the possible or optional components. The facts about those possible components are included in knowledge models.

A fact about a possible composition captures that a kind of object can have one or more objects of a particular other kind as parts.

For example: a bridge <can have as part a> pavement.

It is also possible that more knowledge about a composition of such a kind of object should be specified. For example, next to the fact that a bridge can have a pillar as a part, it may also be specified that a particular kind of bridge shall have a minimum number of pillars. For example: a bridge <can have as part a> 0,1 or more pillars.

Note that the specification of the composition of a kind of object (in other words: from which kinds of components a kind of object can be composed) is different from the specification of the subtypes of the kind of object. So the composition hierarchy is something else than the specialization hierarchy. This distinction is earlier described in paragraph 5.1.4.

6.2.2 Elucidation of step 2: Specify possible aspects

A knowledge model also contains facts about describing (optional) aspects that are not part of the definition.

This includes facts about aspects of the whole, but also aspects of the parts. For example: the fact that a road normally has a width[7] is a fact about a whole road. Such a fact is expressed by a relation between the kind of object and the kind of aspect (characteristic). The example fact is expressed as follows:

road <can have as aspect a> width

When a design is made of an object of a kind, some of the aspects will be quantified by a numeric value, which is usually a value on a scale. That is the case for example for length, width, thickness, etc. Other characteristics are of a qualitative nature and will be qualified by a (standard) textual 'value'. For example, material of construction, such as a kind of steel, colour, shape, operating principle, fabrication method, etcetera.

[7] In fact a road always has a width, but it may be the case that an information model about a road does not include that width or the width (value) of a designed road may not yet be known.

The Gellish Dictionary includes definitions of such standard quantitative and qualitative values of kinds of aspects. Domain specific dictionary extensions are required wherever additional standard values are required.

It is possible to include a number of standard values in a collection of allowed values. This enables to specify in a knowledge model that for a particular kind of aspect (or for a kind of aspect in a particular role) only values that are selected from the collection are allowed.

It is possible that an organization specifies its own list of allowed values, as is described in the next chapter.

6.2.3 Elucidation of step 3: Specify other possible relations

Next to facts about the composition and about aspects of a kind of object there are a large number of other kinds of specifications and kinds of knowledge that can be specified. For example, a kind of object can have a particular role in a kind of activity, such as being subjected to a particular kind of test. These kinds of facts are not about aspects that are inherent to the kind of object, but they concern relations between a kind of object and a kind of activity.

Other examples are possible connections between objects of particular kinds or relations between kinds of objects and kinds of documents.

Details of the modeling of such additional possible relations are given in the Gellish Modeling Method part 3.

7 The creation of requirements models

The third layer in the Architecture of the Gellish Modeling Method is the part that specifies how requirements models and standard specification models are created. Requirements models and standard specification models are both models about kinds of things.

Requirements models are models that do not specify what can be the case (as is done in knowledge models), but they specify what shall be the case in a particular context, when particular standards are declared to be applicable. The way in which a requirements model is created is very similar to the way in which a knowledge model is created. The main difference is the use of other relation types. For example, the fact that a particular type of pipe shall be made of stainless steel is specified as follows:

pipe type A   shall be made of   stainless steel

Requirements models about types of objects should also be distinguished from Facility Information Models that specify requirements for individual things as are described in the next part of the Architecture.

Figure 13 illustrates how a requirements model or standard specifications model relates to a dictionary and to knowledge models. For example, a requirements model may be for a motorway or for a particular type of motorway that is a subtype of motorway. Such a model consists of a ‘cloud’ of facts that might for example specify the requirements that a government has published for motorways. That government may have published those requirements in a requirements library on a website for usage via Internet by any of its suppliers.

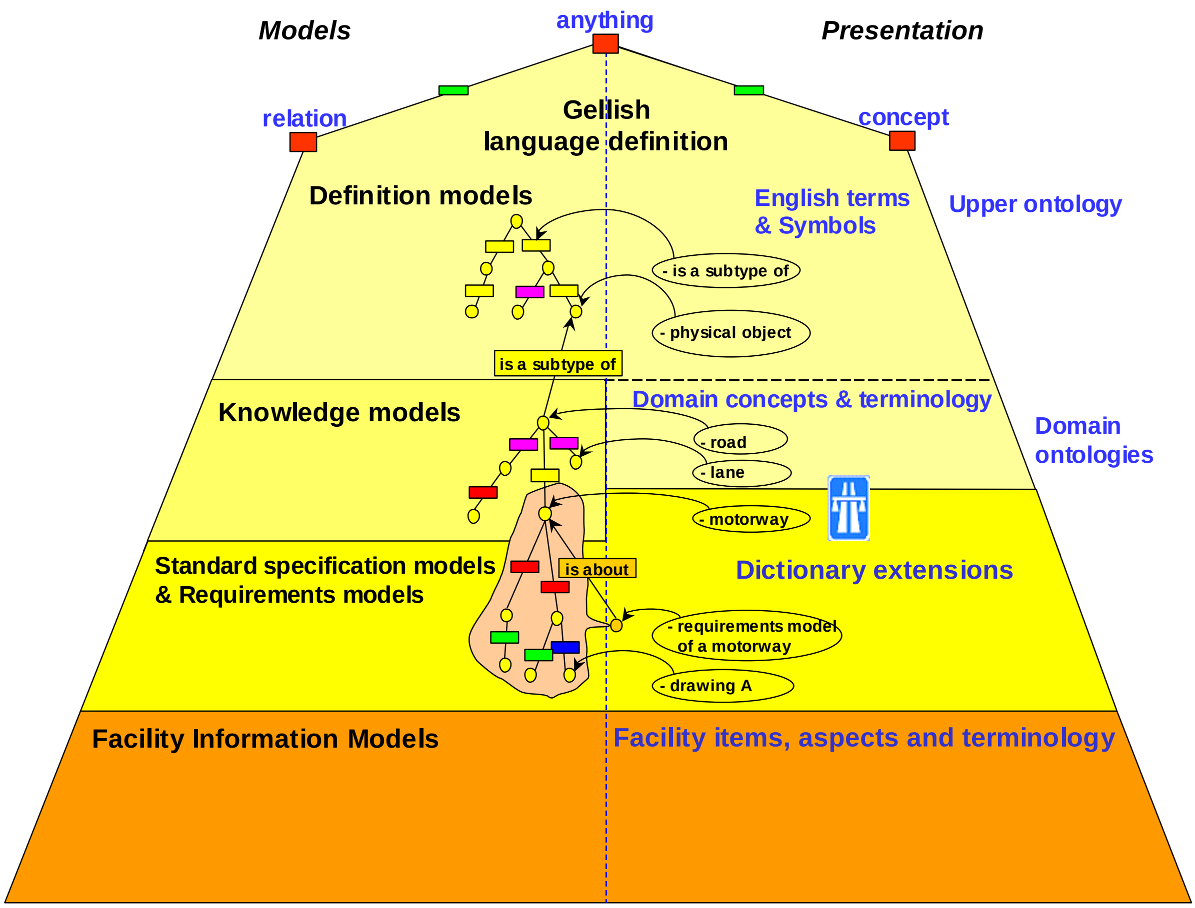

Figure 13, Requirements models in relation to knowledge models and definition models

Requirements models and standard specification models are valuable in many application domains such as the application area's that are described below.

During investment projects designs are made that specify imaginary individual objects and their operations, whereas during fabrication and construction real world individual objects are created. It is often considered valuable to reuse knowledge that is gained during such projects. For example, some designs may be applied several times in the same project or in other projects. This makes it attractive to convert such standard designs in company specific standard specification models and to include them in project or company specific libraries.

Standard specifications may also be developed in a wider context by industries and industry sectors and published as industry standards or national or international standards. Examples are standards for nuts, bolts, pipes, cables, etc. etc.

Another kind of standard specifications consists of the standardized types of products that companies offer on the market. Those companies thus specify standard kinds of products. Such product catalogues are also a kind of library of standard specifications.

Yet another kind of standard specifications is formed by standard buying specification that specify required aspects of kinds of products and components that are bought by those companies. Companies may develop such standard buying specifications in order to standardize on the use of particular types of products and thus to reduce their variety of products in stock.

Requirements models can be specified by directly using the definitions of the concepts in the Gellish Dictionary, extended with concepts in domain dictionaries. They can also be specified as additions to knowledge models in a knowledge library. It may also be the case that they are specifications of subtypes of the kinds of objects that are used in the definition models and knowledge models. For example, a standardized type of pipe that is made of stainless steel, has a diameter of 6 inch and has a number of other standard specifications is a subtype of the general concept pipe. That subtype inherits the general knowledge about pipe, so that a computer can point to other information that is also applicable for the specialized subtype.

Models may be created that are a mixture of knowledge and requirements. In other words, they may be a mixture of options and obligations.

An example of a standard specification model is a model of type of viaduct V-1 that is applied several times on the M1 and other motorways in the UK. For example, viaduct 26 on the M1 is classified as a V-1 viaduct. There are several requirements models created for this kind of viaduct, for example one about the construction (the composition), one about functional requirements and one about cost factors.

8 Creation of Facility Information Models

The last part of the Architecture of the Gellish Modeling Method describes the creation of Facility Information Models. When the facility is a building, then the model can also be called a Building Information Model, when the facility is a process plant then the model can be called a Plant Information Model, etc.

Figure 14 illustrates how Facility Information Models are related by a classification relation to the concepts in the Dictionary and to the knowledge models and the requirements models.

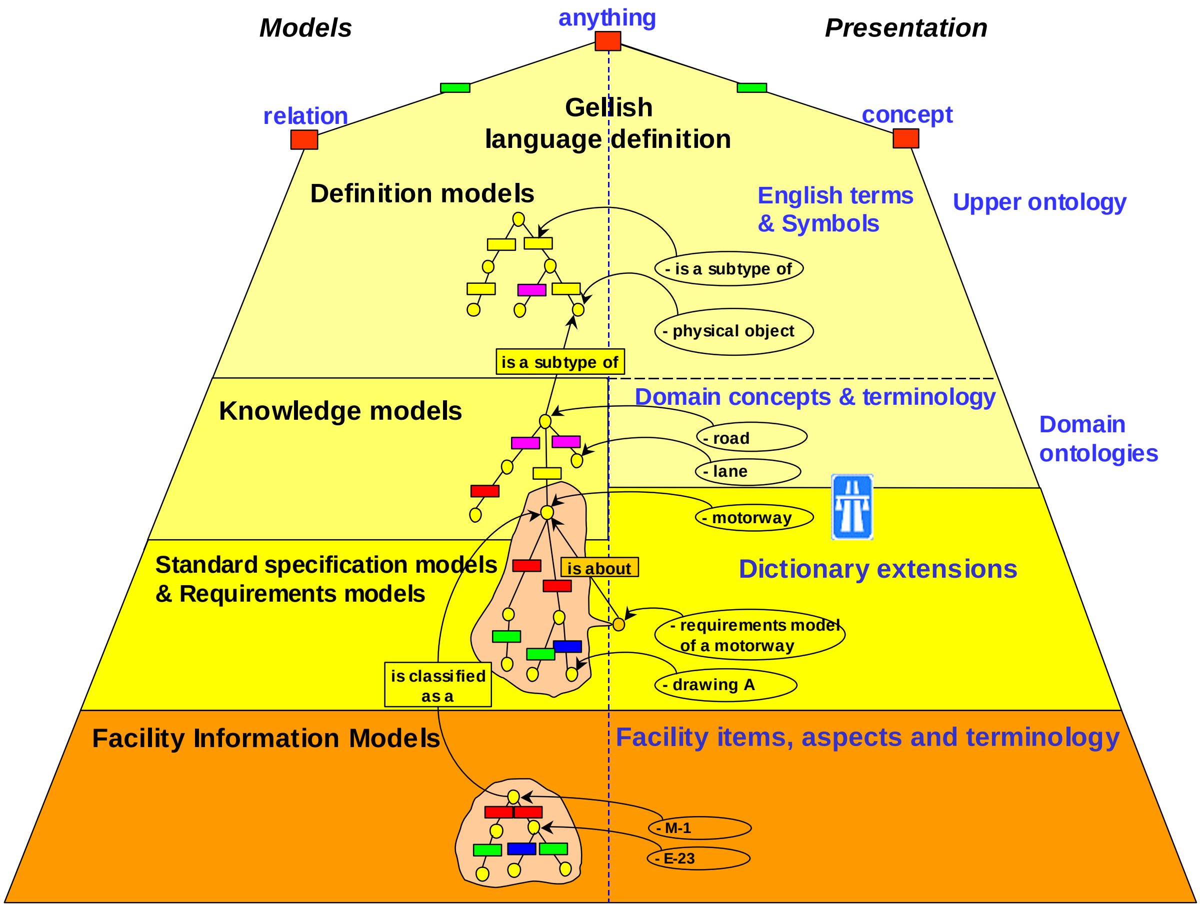

Figure 14, Facility Information Models in the Architecture

Facility Information Models are models that consist of facts that describe information about individual objects. Information about individual object should be distinguished from knowledge, which is about kinds of objects. Examples of Facility Information Models are models about a particular road such as the M1, the Empire State Building, the Queen Elisabeth, or the Pernis Refinery. A specification of information about individual pieces of equipment such as pump P-1301 results in Object Information Models or System Information Models. Investment projects usually specify requirements for complete facilities that consist of assemblies and arrangements of several or many pieces of equipment and systems that are integrated in one big whole. Therefore, the Gellish Modeling Method also describes how all such Information Models can be integrated in one Facility Information Model for a whole facility.

There are two kinds of Facility Information Models: models about designs and models about (observable) real world physical objects. A model that contains design information is an information model about an imaginary individual object, because a designed object only exists in peoples minds. A model can also be about a fabricated and constructed physical object. Then the information in the model consists of observed or measured information. Facility information models may also consist of combinations of design information and real world information. For example, a designed pump may be referenced by the tag name P-1301. After construction and during operations there may be several pumps installed over time on that location, for example initially pump ABC 123 of manufacturer A was installed, which pump was later replaced by pump DEF 234 from manufacturer B. This illustrates that in this case we have information about three different objects. A Facility Information Model that contains both kinds of information will include relations between those objects. For example:

ABC 123 is installed for P-1301

8.1 Modeling of and individual facility

The creation of a Facility Information Model typically consists of the following steps:

1. Create a facility breakdown structure.

   The creation of a facility breakdown structure implies to decompose the facility in parts and to decompose the parts into parts of parts, etcetera. Such a facility break down structure is usually terminated with components that are ordered as a whole and that are often selected from supplier catalogues. However the suppliers of components will usually further decompose those parts up to the level of components that are delivered by their sub-suppliers, or may specify spare parts for components of the assemblies that they sell.

2. Classify the facility and its components.

   Classification is the specification of a relation between an individual thing and a concept (a class) that is selected from the Gellish Dictionary or its extension. The classification of a whole facility and all its components means that the parts of the facility, such as units, systems are classified as well as the parts of the parts to the required level of detail. The classification process is further clarified in the next paragraph.

3. Describe and classify documents about the facility and its components and relate them to files in directories.

   A document consists of information that can be expressed in various physical forms, such as ink on paper and electronic files in various formats. For example in pdf format or doc format or both. To define a document it should be classified and related to one of more physical forms, which physical objects may be collected in collections, such as in binders and/or directories.

4. Relate documents to the facility and to its components.

   After documents and facility components are defined, the documents should be related to the facility and its components about which they provide information, so that the documents can be searched via the facility breakdown structure.

5. Add fabricated items and relate them to design items.

   When fabricated items are produced and installed according to a design, it becomes possible to define the individual fabricated items and classify them. This classification can be for example by the manufacturers model or by the type of equipment. After that the fabricated items can be related to the (imaginary) design items. When in the course of time the installed items are replaced by other installed items, the new installed item shall be related to the same design item.

6. Add occurrences, such as processes, activities (functions) and events.

   Often it is also required to specify occurrences, being either processes or activities or events in which the physical objects are involved. Such an involvement can be as performer of the process, as being subjected to an activity or event or in any other role. This implies that the occurrence or kind of occurrence shall be defined and classified and that the involvement relation shall be specified. As part of the specification of processes it is also possible to specify fluids (streams and batches) and their roles in the processes, usually as input or output.

7. Add aspects to items and processes and activities.

   In addition to the information about the objects and occurrences that is included in the documents it is possible to specify data about them in the database. This means that the aspects have to be defined, classified and quantified and related to the components, fluids or occurrence by which they are possessed. For example, diameters, capacities, shapes, flow rates, densities, etc.

8. Add other objects and relation types.

   Finally other data and relations can be added, such as information about people and their roles in occurrences, connections between items, etc.

8.2 Classification

Classification is the process of specifying a classification relation between an individual thing and a concept that is defined in a definition model and that may appear in a knowledge models or requirements model.

Each individual physical object in a Facility Information Model shall be classified by a *kind of object*. Also all aspects, activities, functions (roles), etc. shall be classified by the concepts (kinds of things) that are defined in the Gellish Dictionary or a dictionary extension.

This is required, because it is the intention that knowledge and requirements about those *kinds* of things are related to those concepts (kinds of things) in the dictionary, so that the classification relations determine which knowledge and requirements are applicable for the individual things and which are not applicable.

Therefore, it is important that individual facilities and components of facilities are classified by kinds of objects in the Gellish Dictionary or a dictionary extension. Such a classification is the prime means to search and find knowledge about objects, because knowledge is usually expressed by means of relations between those kinds of objects and relations between those kinds of objects and the documents that contain knowledge about those kinds of objects.

For a similar reason it is important to classify individual occurrences, fluids, and aspects by concepts in the Gellish Dictionary or a dictionary extension.

The most important reason to classify things that way is that by such a classification all parties that are involved in the total life cycle of a facility will use unambiguous definitions, requirements, etc. so that information exchange between the parties and interpretation of requirements becomes less ambiguous.

In order to be able to classify individual things it is obvious that the concepts that need to be used for classification should have been defined in the first place. This gives evidence to the need to develop and maintain a common Gellish Dictionary and possible domain specific extensions.

For example, a Motorway Information Model for motorway M1 that is expressed in Gellish would means that a part of the model for example of a particular viaduct could contain the following facts:

V5	<is classified as a>	viaduct type 3
s-5	<is classified as a>	road surface layer
s-5	<is a part of>	V5

When the concepts ‘viaduct type 3’ and ‘road surface layer’ exist and are defined for example in a Domain Dictionary this means that the knowledge about those concepts and about the concepts that are related to them, such as their parts, are directly accessible.

This example illustrates how the objects V5 and s-5 are classified by concepts from a domain dictionary by means of a classification relation. In addition to that, the third line illustrates a Gellish expression of a fact in an information model about motorway M1.

9 Usage of knowledge from a knowledge library

The knowledge in a knowledge library can be used by importing knowledge in systems and in systems that are intended for knowledge aided design or knowledge aided verification of designs or verification of inspected real objects. However, the knowledge in a knowledge library can also be used directly by users, varying from trainees to experts, as is illustrated by the following two examples of a designer and a cost estimator.

9.1 A designer

A designer who is involved for example in the design of a particular (individual) steel bridge has to know which general requirements are applicable for the quality of a steel bridge. One of the aspects he has to take into account is the lifecycle cost of the bridge. Another aspect is that the material from which the bridge will be made should be compliant with specified rules for availability, reliability, maintainability and safety, for example according to the RAMS model.

Assume that he has to ensure that the costs of the design remain within an agreed range. He might save on kind of material, but questions whether some corrosion is allowed for a steel bridge. This means that he wants to search for knowledge, requirements and risks about a ‘steel bridge’ and especially about the process ‘corrosion’.

Assume that he discovers that corrosion in not allowed at all on a steel bridge. Then as a next search he may look for material or protection material that provides sufficient protection against corrosion. In other words materials that satisfy the specified requirements.

But assume that some corrosion is allowed on a steel bridge. Then the designer should know what is meant with ‘some corrosion’ or a ‘reasonable amount of corrosion’. He might need pictures that are collected in his company’s knowledge library under the heading ‘degradation of bridges’. Then a knowledge library that is designed to support such a design process will contain a knowledge models about the concept ‘bridge’ in which pictures of various kinds of degradation are included and thus are readily available for the designer.

9.2 A cost estimator

A cost estimator who is involved in an early phase of a design of the above mentioned steel bridge will ask himself: what are the costs of such a steel bridge?

The first thing he (or she) should know is: what is the precise definition of a steel bridge. The cost estimator will first use a particular piece of software to consult the Domain Dictionary for that.

From that he understands that a steel bridge is by definition a bridge of which the deck is hanging on a steel arc.

The next things he should know are:

- Which design options are available for a steel bridge
- What are the components from which a steel bridge is composed.
- What properties and specifications are applicable for the whole and for its parts.

Finally he has to determine what the costs per components of the assembly are.

In de knowledge library that is developed to support such a cost estimator he will find a knowledge model that is specifically created as a cost reference model. Such a model has a generic part with facts about kinds of solutions (types of steel bridges) and facts about typical compositions for those kinds of solutions (kinds of components from which those kinds of steel bridges are composed), whereas those collections of facts are specifically created for cost estimation and budgeting.

This will result in a survey of costs for various kinds of steel bridges.

Only after that he will be able to create a cost estimate for a steel bridge that is created according to some design solution.

10 What if a concept is missing in the dictionary

Assume an architect is given the order to design a general solution for the composition of ecoducts for the city of Frankfurt, whereas she makes use of a particular Domain Dictionary. She receives a specification of the requirements of the city of Frankfurt to which an ecoduct should comply. This means that she should model all her knowledge about ecoducts in a composition model. That composition model should make use of a definition model of an ecoduct, because the definition model of an ecoduct belongs to the common terminology that is used by all parties.

However, it appears that the concept 'ecoduct' is not included in the Domain Dictionary that is used by the Architect.

Now she has two options:

- To arrange that the concept ecoduct is added to the Domain Dictionary.
- To arrange that the concept ecoduct is added to the general open source Gellish English Dictionary.

In both cases she can arrange that the new definition model is either developed by the maintenance organisation of his Domain Dictionary by the Architect's own organisation. The difference is whether the new definition model is approved and included in a local Domain Dictionary or is approved and included in the international public domain general Gellish Dictionary. In the latter case all other parties that are involved in this and in future projects will automatically share the common definition.

Whoever will define the concept, he or she has to ensure such a quality of the definition that there is reasonable consensus in the domain community about the definition. He or she shall also define the concept as a subtype of an existing supertype concept (possibly being 'bridge') and shall verify whether all facts about the subtype are indeed valid when the concept ecoduct is defined as a subtype of that supertype in the specialisation hierarchy of concepts. In other words: the defining party has to convince himself (and others) that the concept ecoduct indeed inherits all defining aspects from its supertype concept ('bridge').

Finally it is important to record the references to sources of information that are used to create the facts in the definition model.

An extensive description of this modeling method for defintions, including a template to facilitate the creation of new definition, together with instructions for filling in that template, are included in the document Gellish Modeling Method part 2, Creation of Domain Dictionaries.

11 List of used concepts

The most important concepts that are used in this document are included in the following list of concepts.

The concepts are presented in the form of a Gellish Database table. Each concept has a Unique Identifier (UID) and a name. Each line in the table expresses a fact that is denoted by a Unique Identifier of that fact.

For each concept there is a line on which is specified that the concept is a subtype of a more general concept. That is done by using the relation type <is a specialisation of> (UID 1146). On that same line a textual definition and an example is added.

Furthermore, there are lines on which synonyms are specified by means of the relation type <is a synonym of>.

Note: in case of a synonym the unique identifier of the left hand object and of the right hand objects are identical, because the two names denote the same concept.

(see Appendix in Excel form)