Summary

Gellish
A Generic Extensible Ontological Language

- Design and Application of a Universal Data Structure -

Cover illustration:

Museum: Boymans van Beuningen, Rotterdam, The Netherlands
Pieter Brueghel de Oude, The Tower of Babel (Genesis 11:1-9)

At first everyone spoke the same language…
Then people said: “Let's build a city with a tower that reaches to the sky! We'll become famous.”
But the LORD said: “Come on! Let's confuse them by making them speak different languages, then they won't be able to understand each other…”
So the people had to stop building the city…
That is why the city was called Babel (confusing) -because there the LORD confused the language of the whole world…

Summary

Gellish
A Generic Extensible Ontological Language

- Design and Application of a Universal Data Structure -

Thesis
to obtain the degree of doctor
at the Delft University of Technology,
on the authority of the Rector Magnificus prof.dr.ir. J.T. Fokkema,
to be defended in public for a committee,
appointed by the 'College voor Promoties',

on Wednesday 14th of September 2005 at 10.30 hour
by Andries Simon Hendrik Paul VAN RENSSEN
mechanical engineer
born at Utrecht

This thesis is approved by the promotor:

Prof.dr.ir. J.L.G. Dietz

Composition of the committee:

Rector Magnificus, chairman

Prof.dr.ir. J.L.G. Dietz, Delft University of Technology, promotor

Prof.dr. J.S. Dhillon, Delft University of Technology

Prof.dr.ir. P.A. Kroes, Delft University of Technology

Prof.dr. R.A. Meersman, Free University, Brussels, Belgium

Prof.dr.ir. A.W.M. Meijers, Delft University of Technology

Prof.dr. B. Smith, State University of New York at Buffalo, United States of America

Ir. G.J. van Luijk, Technische Universiteit Delft,
previously director of Shell Global Solutions International

Summary

*The **problem statement** of this research is the question whether it is possible to provide a formal generic artificial language for an unambiguous description of reality, that is based on natural language, is defined in a formal ontology[1], and is practically applicable, at least for technical artifacts[2] such that it is suitable to express and exchange information in the form of electronic data in a structure that is system and natural language independent.*

The problem behind this statement is that information exchange between computers and integration of information that comes from different sources is currently hardly possible without the creation of costly data conversions and interface software. This is caused by that fact that software developers are trained to create a new data structure for each new application and by the fact that software users are used not to use a standard for the reference data that is part of the content of those data structures. Together this causes a 'Babylonian confusion of tongues' in information exchange between computer systems. This means that there is no common language for communication with and between computer applications. This hampers the unambiguous interpretable storage, integration and retrieval of information and often requires the development of costly data conversion procedures. A solution to this problem is therefore of considerable social and economic importance.

The research that is described in this document resulted in a solution to the root problem by the creation of an artificial language that can be used for an unambiguous and computer interpretable description of reality and imagination. The artificial language is a formal subset of natural languages with variants per natural language. That artificial language provides an extensible and generally applicable data structure that potentially eliminates the need to develop ad hoc data structures for many applications. The language is called Gellish[3], and its variants are Gellish English, Gellish Nederlands, etc. Gellish is defined in such a way that expressions in one language variant can automatically be translated in any other language variant for which a Gellish dictionary is available. Gellish is system independent and is both human and computer readable. Gellish might also provide a contribution to the technology of computerized natural language processing.

The Gellish language is developed through an analysis of commonalities and limitations of data models and of many issues in data modelling, in combination with a study of ontologies and generic semantics of languages.

[1] An ontology is the result of research on the nature and properties of things. An ontology can be presented as a graphical model, as is illustrated by the figures in this thesis.

[2] Artifacts are products that are made by human beings.

[3] Gellish is originally derived from 'Generic Engineering Language', however it is further developed into a language that is also applicable outside the engineering discipline.

The following highlights illustrate the development process of Gellish.

1. First a **generic data model** was developed to increase the scope of data models to a general applicability so that it can replace many different conventional data models.
   The difference between conventional data models and a generic data model is illustrated in Figure 1.

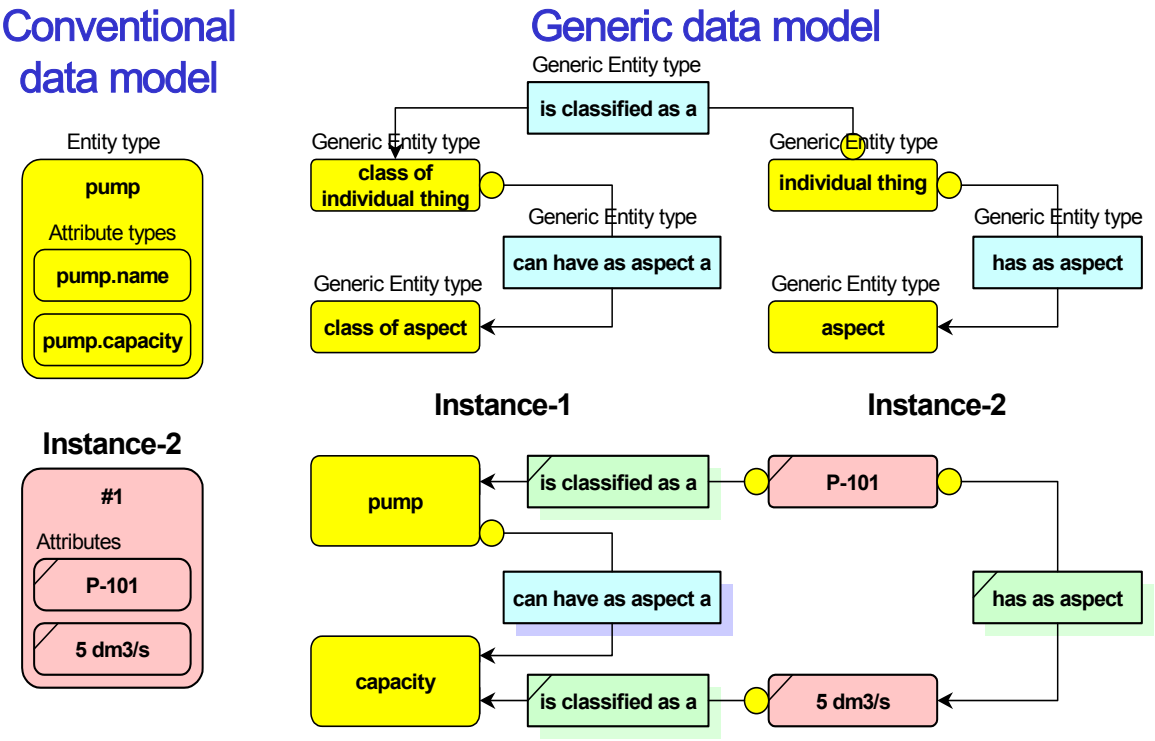

Figure 1, Conventional versus generic data models

Generic data models include among others an explicit classification relation (the 'is classified as a' relations in Figure 1) between two different entity types. That explicit classification relation in principle replaces the implied instantiation relation between the attribute instances and the attribute types in conventional data modelling. The entity types in a generic data model has no attributes, because attribute types are replaced by entity types whereas implicit and explicit relations between attribute types (which relations are unqualified in conventional data models) are replaced by explicitly qualified relations. This resulted in a generic data model with a taxonomy of relation types. The generic data model thus contained a strict specialization hierarchy (subtype/supertype hierarchy) of all concepts, ending with the most generic concept 'thing' or 'anything'. The resulting generic data model was standardised in two ways in ISO standards[4].

2. At the same time an accompanying **ontology** was developed, based on a **taxonomy of concepts,** to capture application domain knowledge in a flexible, extensible way. The author of this document coordinated the work of many discipline engineers, organized in 'peer groups' to come to agreement about the definition, taxonomy and ontology of the concepts in their domain. The result was expressed as a database of concepts that are mutually related and that are accompanied by definitions, whereas domain knowledge was captured as relations between concepts. The concepts and knowledge were originally defined to be instances of the entity types of the generic data model.

[4] ISO 10303-221 and ISO 15926-2.

3. It was discovered that all instances of the generic data model could be expressed in a single table. Therefore the **Gellish Table** was developed. An example of a Gellish Table is presented in the following tables.

101	3	201	7
Left hand object name	**Relation type name**	**Right hand object name**	**UoM**
car	is a subtype of	vehicle	
wheel	is a subtype of	artifact	
wheel	can be a part of a	car	
W1	is classified as a	wheel	
C1	is classified as a	car	
W1	is a part of	C1	
W1	has aspect	D1	
D1	is classified as a	diameter	
D1	can be quantified on scale as	50	cm

54	2	101	1	60	3	15	201	7
Language of left hand object	**Left hand object UID**	**Left hand object name**	**Fact id**	**Relation type id**	**Relation type name**	**Right hand object UID**	**Right hand object name**	**UoM**
English	670024	car	1	1146	is a subtype of	670122	vehicle	
English	130679	wheel	2	1146	is a subtype of	730063	artifact	
English	130679	wheel	3	1191	can be a part of a	670024	car	
multi-lingual	10	W1	4	1225	is classified as a	130679	wheel	
multi-lingual	11	C1	5	1225	is classified as a	670024	car	
multi-lingual	10	W1	6	1190	is a part of	11	C1	
multi-lingual	10	W1	7	1727	has aspect	12	D1	
multi-lingual	12	D1	8	1225	is classified as a	550188	diameter	
multi-lingual	12	D1	9	5279	can be quantified on scale as	920303	50	cm

Figure 2, Example of a Gellish Table

The information that is recorded in the above tables could be presented in free form Gellish English for example as:

- a car is a kind of vehicle and a wheel is a kind of artefact, whereas a wheel can be a part of a car.
- wheel W1 is a part of car C1, whereas the diameter D1 of wheel W1 is 50 cm.

Further research is required to investigate the applicability of this free form Gellish. This thesis is limited to a representation of Gellish expressions in table form.

The first of the above tables contains the part of a Gellish Table that is human readable. Each line in that table is the expression of a *fact*. The second table is an illustration of a more extended version of the same Gellish Table, in which also the unique identifiers are presented as well as the indication of the language in which the *facts* are expressed. The unique identifiers are natural language independent. This means that the facts are recorded in a natural language independent way, so that it becomes possible that, by using a Gellish dictionary, a computer can generate and

present the same table also in other languages. This is illustrated when the above Gellish Table is compared with **Error! Reference source not found.** in the Dutch Summary of this thesis.

The Gellish Table is an implementation method (syntax for the Gellish language) that is suitable to express any kind of facts. For example:

- To store the *definitions* of concepts.
  – See line 1 and 2 in the above table, with specialization relations that define domain concepts (although details of those definitions on those lines are not shown).
- To express *knowledge* as relations between concepts.
  – See line 3, which contains an example of a relation between kinds of things.
- To store *information* about individual things.
  This regards facts and data (instances), expressed as relations between individual things.
  – See line 6 and 7.
- To express the *classification of things* as relations between individual things and concepts (kinds) as well as other kinds of relations between individual things and concepts.
  – See line 4, 5, 8 and 9.
- It appeared that a *single* table is suitable to express any kind of fact, including the definition of the language itself.
  The Gellish Table represents a structured form of a subset of natural language grammar. Its core consists of concepts (represented in the above table by the left hand objects and right hand objects) and Gellish phrases (represented in the above table by the relation types).

4. In a next phase it was discovered that the generic entity types in the data model (being the meta-language concepts) are identical to the higher-level concepts in the discipline ontology. This is illustrated in Figure 3.

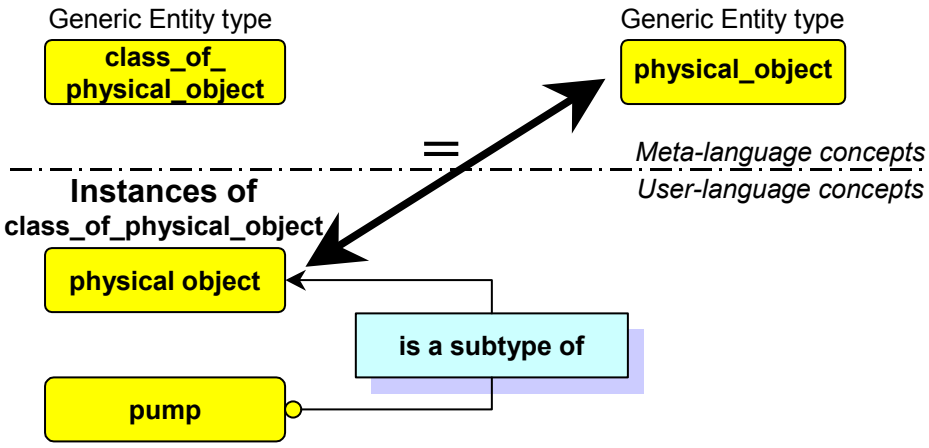

Figure 3, Identity between entity type and instance

Figure 3 illustrates that for example the generic entity type ‘physical_object’ in the data model, is a meta-language concept that appeared to be the same thing as the concept ‘physical object’ in the taxonomy. Furthermore, that concept is a supertype of all classes of physical objects in the Gellish Table database, whereas that concept and each subtype is an instance of the entity type ‘class_of_physical_object’. Because of that identity, the meta-language concepts, being the generic data model concepts (entity types) were added to the ontology as its upper ontology and the **data model was eliminated** or reduced to a ‘bootstrapping’ mini data model and the data model was replaced by a Gellish Table with the definition of the upper ontology. The data model entity types that were added to the ontology included also relation types and the roles of various kinds that are required by relations and that are played by objects. With the addition of the relation types and role types, a corresponding consistent specialization hierarchy or taxonomy of relations and roles was created. The result is that Gellish eliminates the conventional distinction between application language (user data) and meta-languages (data models). The concepts that occur in

meta-meta-languages in which data models are usually written (such as EXPRESS, UML, XMLS or OWL) could be included in the Gellish ontology. In contrast with the conventional distinction in meta-levels of languages, Gellish integrates all concepts in those three levels in one language. This is illustrated in Figure 4.

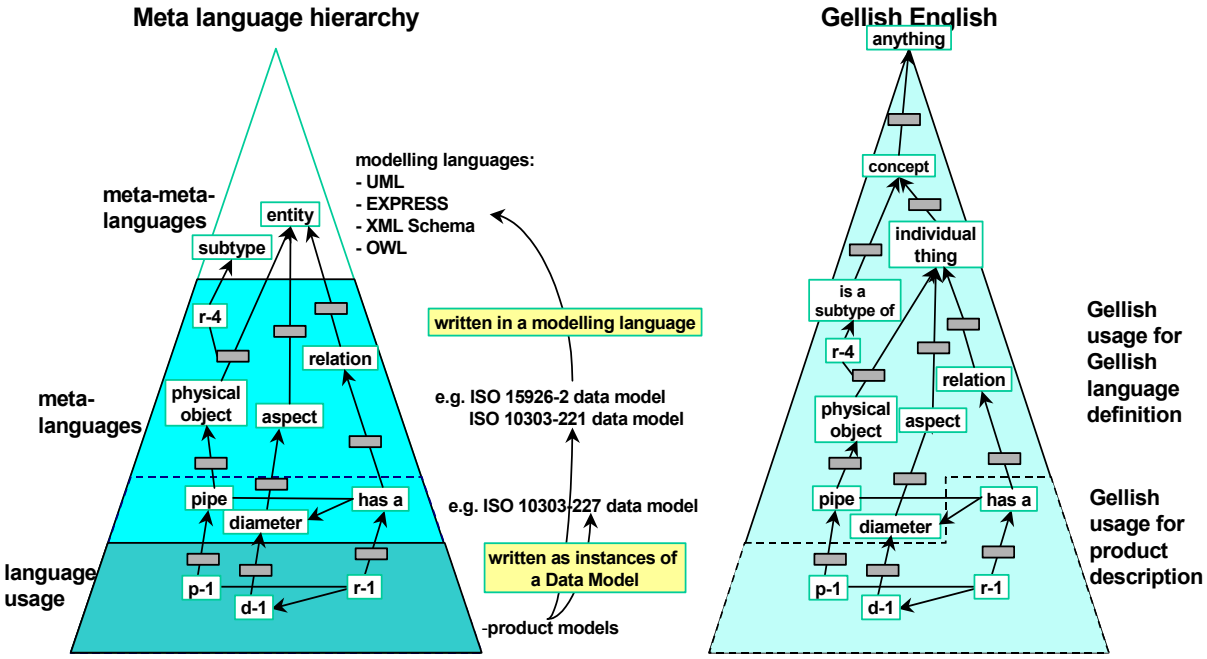

Figure 4, Gellish as one integrated language

The left hand part of Figure 4 illustrates the distinction between concepts from various languages (product models or user languages), data models or meta-languages and modelling languages or meta-meta-languages. The right hand part illustrates that all the concepts from the various levels are integrated in a single Gellish language.

5. **Generally applicable browser software**[5] was developed that can read Gellish Tables and that can verify the semantic correctness of the content. That software proofed that it is possible to read the upper ontology knowledge that is contained in a Gellish Table and that subsequently can use that knowledge to interpret a discipline ontology. Furthermore, that software was also able to interpret Gellish Tables with models with information about individual products and occurrences, to verify their correctness and to display those models.

6. The next step was the discovery that many entity types that were converted into upper ontology concepts were semantically **superfluous artefacts** that could be **removed** from the ontology without a loss of semantic expressiveness. This is illustrated in Figure 5.

[5] The STEPlib Browser, which is actually a general Gellish Browser.

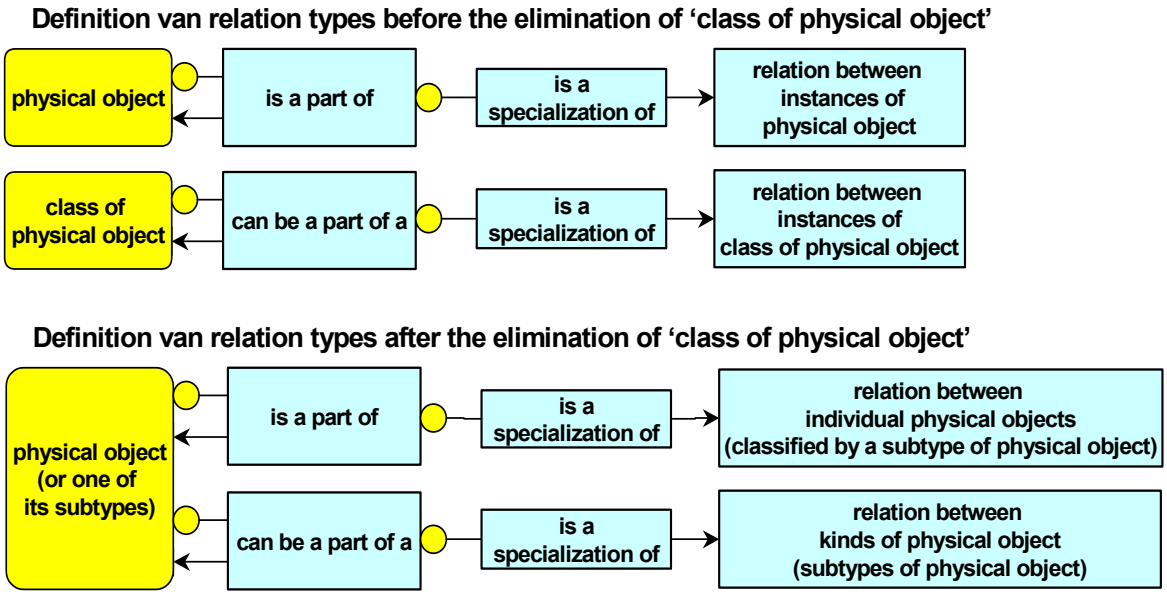

Figure 5, Elimination of superfluous concepts

For example, the concept 'class of physical object' in the left hand part of Figure 5 was originally a generic entity type, intended to contain (or to collect and/or classify) instances of classes of physical objects, such as 'car' and 'wheel'. Such instances of concepts are used to define the semantics of possibilities for relations between members of classes. Once the relation types, such as 'can be a part of a', are defined in the Gellish language, they can be used to classify relations between instances of 'class of physical object'. For example, the 'can be a part of a' relation can be used to express the knowledge that 'a wheel can be a part of a car' in a computer interpretable way.

However, it was discovered that it is also possible and even better to define that same 'can be a part of a' relation type as a relation between the concept 'physical object' and itself, as is indicated in the right hand part of Figure 5, in which case the meaning of the relation type is defined as a relation type that expresses that an individual thing that is classified as a 'physical object' *or one of its subtypes* can be a part of another individual thing that is also classified as a 'physical object' or one of its subtypes.

This discovery made the artificial concepts, such as 'class of physical object', superfluous and therefore that kind of concepts were removed from the ontology.

7. Furthermore, it appeared that **semantically unnecessary subtypes** could be eliminated from the ontology, especially as they do not appear in natural languages either. For example, the relation type 'has property' could be replaced by the more general relation type 'has aspect', because such a subtype duplicates the semantics that is already contained in the fact that by definition the property that is possessed already will be classified as a subtype of 'property' and because the concept 'property' is defined as a subtype of 'aspect'.

8. On the other hand **additional subtypes of relation types** appeared to be required to be added to the ontology to capture the precise semantics of kinds of facts that appeared to be present in the various application domains.

During the above development the resulting ontology was aligned with the concepts that resulted from an analysis of various ontologies and of natural language.

Natural languages are probably not designed by human beings, but we generally take their existence for granted and mainly analyse their structure and the underlying concepts. Such research reveal that the various languages seem to be using common semantic concepts (Wierzbicka, 1996). This research led to the conclusion that that common semantics is formed by concepts and relationships of a limited number of kinds between concepts, whereas for those concepts and kinds of relationships different terms are used in different cultures and languages. Gellish is a language that is built on the elements

from those **semantic commonalities** between languages. The following common elements are captured in the Gellish language:

- **Concepts.**
  When human beings communicate, they seem to use the same concepts, irrespective of the language they use, although they refer to those concepts by different names in the various languages. Therefore, Gellish makes an explicit distinction between the language independent concepts and the terms or phrases with which the concepts are referred to in different contexts or language communities. Each concept is referred to in Gellish by a unique identifier (UID) that is independent of any natural language. Furthermore, the Gellish language refers to those concepts also through terms and phrases from those natural languages. This is done through the use of a symbol or string or pattern of symbols, either written or spoken in the various applied languages. Therefore, Gellish includes a Dutch dictionary, an English dictionary, etc., which dictionaries are structured as a taxonomy of concepts.

- **A basic semantic structure.**
  There seems to be a **basic semantic structure** of languages, which is a commonality that is independent of the natural languages. Such a structure is presented in Figure 6.

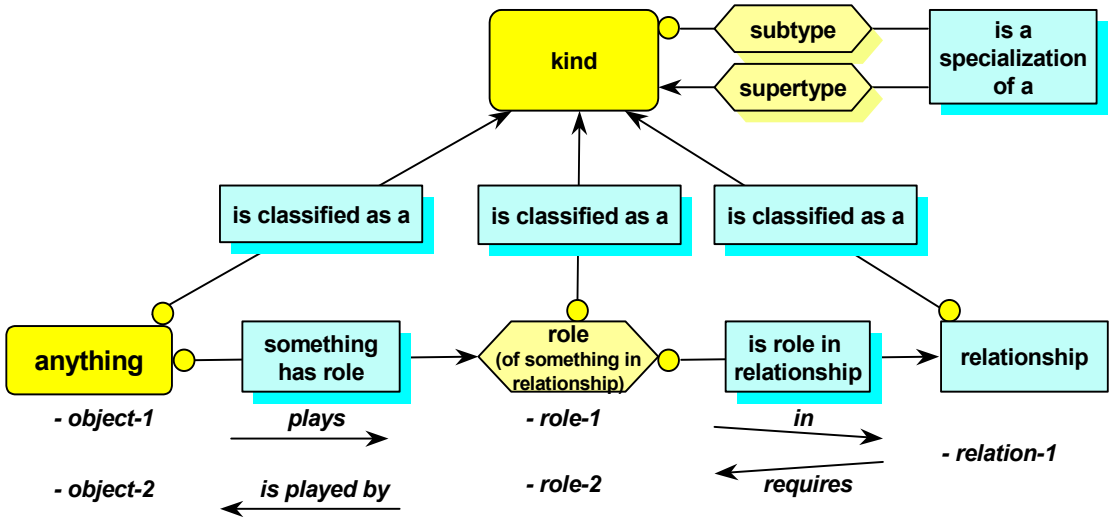

Figure 6, Basic semantic structure

That structure is identified and captured by its inclusion as the basic semantic structure of Gellish for the expression of any fact. In that structure, facts are expressed as relations between things, whereas the relations require two or more roles and only things of a particular kind can play roles of the required kinds. Common facts, or pieces of knowledge, are expressed as (common) relations between concepts; or expressed more precisely: common relations that conceptually require roles of a kind, which roles can be played by members of the related concepts.

- **Facts and relations.**
  A fact is: that which is the case, independent of language. The concept ‘fact’ is a concept that can be used to classify things as ‘being the case’.
  Facts seem to be expressed in languages as relations between things. Therefore that is also the case in Gellish. Gellish is defined to a large extent by the identification of **kinds of relations** that are independent of language and that can be used to classify expressions of facts of corresponding kinds. An analysis of the kinds of relations that are used in ontology, in physics, in engineering and in business processes, revealed that there is a limited number of relation types with which ontologies, technical artefacts and other objects and their behaviour or occurrences can be described in a computer interpretable way. The identification of those relation types resulted in a taxonomy of kinds of relations (or relation types) that is included in the definition of Gellish. Those relation types are also referred to by language independent unique identifiers (UID’s) and per natural language they are referred to by different ‘phrases’ (partial sentences). The semantics

of expressions in Gellish is completed by the use of the concepts and relation types for the classification of individual things and relations.
Many relation types are binary, but occurrences and correlations are examples of n-ary relations in which a number of things are involved. Each of those involvements can be expressed as a binary *elementary* relation. This implies that those higher order relations can be expressed in Gellish as a collection of n binary elementary involvement relations. In each involvement relation, the particular role of the involved thing can be made explicit by using a specific subtype of the elementary involvement relation.

- **Taxonomy of concepts.**
  A subtype/supertype hierarchy or taxonomy of common concepts was developed, which include also the relation types. It appeared that a high degree of agreement could be achieved between domain experts from various languages and countries, about that taxonomy and about the definitions. This resulted in a Gellish English Dictionary / Taxonomy of concepts (also including relation types) with a number of translations of terms and phrases. That dictionary / taxonomy can be extended as and when required with new concepts that can be kept proprietary or can be proposed for addition to the standard Gellish language definition.

- **Ontology.**
  The above-mentioned elements are integrated in a coherent hierarchical network, which includes the definition of the Gellish language. By inclusion of additional knowledge a **knowledge base** was developed which completed the ontology.

The definition of the Gellish language is published as 'open source' data and is publicly available and can be downloaded on the basis of an open source license.

Possible applications of the Gellish language include, but are not limited to:

- The use of the Gellish dictionary and knowledge base as a basis for a company specific data dictionary or knowledge base.
  Such a knowledge base can be used for example as an information source in design systems or as a reference data set for the harmonization of the content of various systems. For example, when various systems have to be replaced by fewer systems that may or may not be of the same type. It is also possible that the Gellish dictionary and knowledge base is used as a basis for an intelligent electronic dictionary or encyclopaedia. The knowledge base can be extended by the expression of additional public domain knowledge or proprietary knowledge is expressed in Gellish and is added to the existing Gellish knowledge base.

- The development of system independent computer interpretable product models.
  This implies that design information about individual products or product types is recorded as product models that are expressed in Gellish. For example design information about parts and assemblies, such as equipment, tools and structures, roads, buildings, ships, cars, airplanes, facilities, etc. This enables that parts or complete product models are exchanged in a system independent way between various parties. Furthermore, it becomes simpler to combine several product models and related documents into one integrated overall product model, even if the contributions stem from different source systems. The fact that the use of Gellish implies the use of standardised concepts means that the consistency of the data is increased and that it becomes simpler to use computer software to support the verification of the quality of the data. This means that a significant quality increase can be achieved. Furthermore, it becomes simpler to develop generic applications that enable to retrieve, compare and report information about different or complex installations, such as comparison of performance data of equipment on different sites that is stored in systems with different data structures.

- The description of behaviour of products, persons and organizations.
  This means that processes and occurrences are described in Gellish, including the description of mechanical, as well as physical, chemical and control processes and the roles that people and organizations play in those processes. Such descriptions are easier to maintain, to exchange

between systems and to search. In addition to that it becomes easier to integrate process descriptions with product descriptions.

- The expression of general and particular requirements in product catalogues.
  This means that specifications for standardised products are described in Gellish. For example, requirements that are expressed in standard specifications as published by standardization institutes. But also specifications of types of products, such as product types that are described in product catalogues of suppliers or of buyer specifications. Such specifications enable that software applications can assist in the mutual comparison of product types or in the selection of product on the basis of specifications, even if those descriptions stem from different sources. This provides suppliers of products for e-business a way to record product information that is unambiguous, neutral and computer interpretable.

- The description of procedures and business processes.
  The expression of such information in Gellish simplifies the maintenance of the business process descriptions and enables their integration and comparison with other process descriptions, especially when the process descriptions make use of a systematic methodology, such as the DEMO methodology. It also becomes possible to develop software 'agents' that are controlled by the knowledge that is contained in those process descriptions, so that they can automatically react on incoming messages.

- The description of information in Gellish about real individual things and occurrences, such as measurements and observations.
  Such system independent recording simplifies for example the integration of time dependent observations with product models that describes the observed objects and increases the clarity about the definitions of the measured variables.

- Improvement of the accuracy of the response of search engines on Internet or other document repositories. This can be achieved by using the relations between the concepts (keywords or key-facts) that are contained in the taxonomy / ontology of Gellish. This knowledge can be used during recording of information as well as during the retrieval process through improved interpretation of the queries.

Various **examples** of applications of Gellish English are presented in this thesis, including:

- A part of the knowledge, requirements and design of a lubrication oil system for a compressor.
- The specification of a catalogue item from a product catalogue.
- A part of a generic business process for communication about business transactions.

Finally, it should be mentioned that the semantics of the upper ontology is summarized in the table 'Gellish English' of the 'upper ontology' part of Gellish. (See the TOPini file with the Gellish Upper Ontology on https://sourceforge.net/projects/Gellish).