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…

Nederlandse vertaling:

Ooit werd er op de hele aarde één enkele taal gesproken…
De mensen zeiden: laten we een stad bouwen met een toren die tot in de hemel reikt. Dat zal ons beroemd maken…
Maar God dacht: …laten wij …spraakverwarring onder hen teweegbrengen, zodat ze elkaar niet meer verstaan…
en de bouw van de stad werd gestaakt…
Zo komt het dat die stad Babel (verwarring) heet, want daar bracht God verwarring in de taal die op de hele aarde gesproken werd…

Gellish
A Generic Extensible Ontological Language

- Design and Application of a Universal Data Structure -

Proefschrift
ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof.dr.ir. J.T. Fokkema,
in het openbaar te verdedigen ten overstaan van een commissie,
door het College voor Promoties aangewezen,

op woensdag 14 september 2005 te 10.30 uur
door Andries Simon Hendrik Paul VAN RENSSEN
werktuigkundig ingenieur
geboren te Utrecht

Dit proefschrift is goedgekeurd door de promotor:

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

Samenstelling promotiecommissie:

Rector Magnificus, voorzitter

Prof.dr.ir. J.L.G. Dietz, Technische Universiteit Delft, promotor

Prof.dr. J.S. Dhillon, Technische Universiteit Delft

Prof.dr.ir. P.A. Kroes, Technische Universiteit Delft

Prof.dr. R.A. Meersman, Vrije Universiteit, Brussel, België

Prof.dr.ir. A.W.M. Meijers, Technische Universiteit Delft

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

Ir. G.J. van Luijk, Technische Universiteit Delft,
voormalig directeur Shell Global Solutions International

Published and distributed by: DUP Science

DUP Science is an imprint of
Delft University Press
P.O. Box 98
2600 MG Delft
The Netherlands
Telephone: +31 15 278 5678
Telefax: +31 15 278 5706
E-mail: info@library.tudelft.nl

ISBN 90-407-2597-4

Keywords: ontology, data modeling, standardization, product modeling, language, engineering, dictionary, taxonomy, data exchange, data storage, application interface

Printed in The Netherlands

Table of content

Samenvatting

van het proefschrift
Gellish
Een Generieke Uitbreidbare Ontologische Taal
Ir. Andries van Renssen

De probleemstelling van dit onderzoek is de vraag of het mogelijk is een formele generieke kunstmatige taal te definiëren die geschikt is voor een eenduidige beschrijving van de werkelijkheid, waarbij die taal gebaseerd is op een natuurlijke taal en gedefinieerd is als een formele ontologie[1]*, terwijl hij op z'n minst praktisch toepasbaar is voor technische artefacten*[2]*, zodanig dat hij geschikt is om informatie uit te drukken en uit te wisselen in de vorm van elektronische gegevens in een structuur die zowel systeem als natuurlijke taal onafhankelijk is.*

Het probleem achter deze probleemstelling is dat informatie-uitwisseling tussen computers en integratie van informatie die afkomstig is van verschillende bronnen momenteel nauwelijks mogelijk is zonder de ontwikkeling van kostbare conversies en interface software. Dit wordt veroorzaakt door het feit dat software ontwikkelaars getraind zijn om nieuwe gegevensstructuren te creëren voor iedere nieuwe applicatie en door het feit dat softwaregebruikers over het algemeen gewoon zijn om geen standaard te gebruiken voor de referentiegegevens die deel uitmaken van de inhoud die in die gegevensstructuren wordt opgeslagen. Deze feiten tezamen veroorzaken een Babylonische spraakverwarring tussen computersystemen als daartussen informatie uitgewisseld moet worden. Dit betekent dat er geen gemeenschappelijke taal is voor de communicatie met en tussen computersystemen. Dit belemmert een eenduidig interpreteerbare opslag, integratie en uitwisseling van informatie en veroorzaakt dat vaak kostbare procedures voor de conversie van gegevens noodzakelijk zijn. Oplossing van dit probleem is daarom van aanzienlijk maatschappelijk en economisch belang.

Het onderzoek dat in dit document beschreven wordt resulteerde in een oplossing voor het kernprobleem door de ontwikkeling van een kunstmatige taal die gebruikt kan worden voor de eenduidig en computer interpreteerbare beschrijving van de werkelijkheid en van de denkbeeldige dingen. Die kunstmatige taal is een subset van natuurlijke talen met varianten per natuurlijke taal. De kunstmatige taal biedt een uitbreidbare en algemeen toepasbare gegevensstructuur die in beginsel de noodzaak wegneemt om voor elke nieuwe applicatie een nieuwe gegevensstructuur (data model) te ontwikkelen. De voorgestelde taal is Gellish genoemd[3] en zijn varianten heten Gellish Engels, Gellish Nederlands, enz. Gellish is zo gedefinieerd dat uitdrukkingen in de ene taalvariant automatisch vertaald kunnen worden in iedere andere taalvariant waarvoor een Gellish woordenboek beschikbaar is. Gellish is systeemonafhankelijk en is leesbaar voor zowel mensen als computers. Gellish zou wellicht ook een bijdrage kunnen leveren aan de technologie om natuurlijke talen met behulp van computers te verwerken.

De Gellish taal is ontwikkeld via een analyse van de overeenkomsten en beperkingen van gegevens modellen en van veel vraagstukken die zich voordoen bij gegevens modelleren, tezamen met het bestuderen van ontologieën en de algemene semantiek van talen.

---

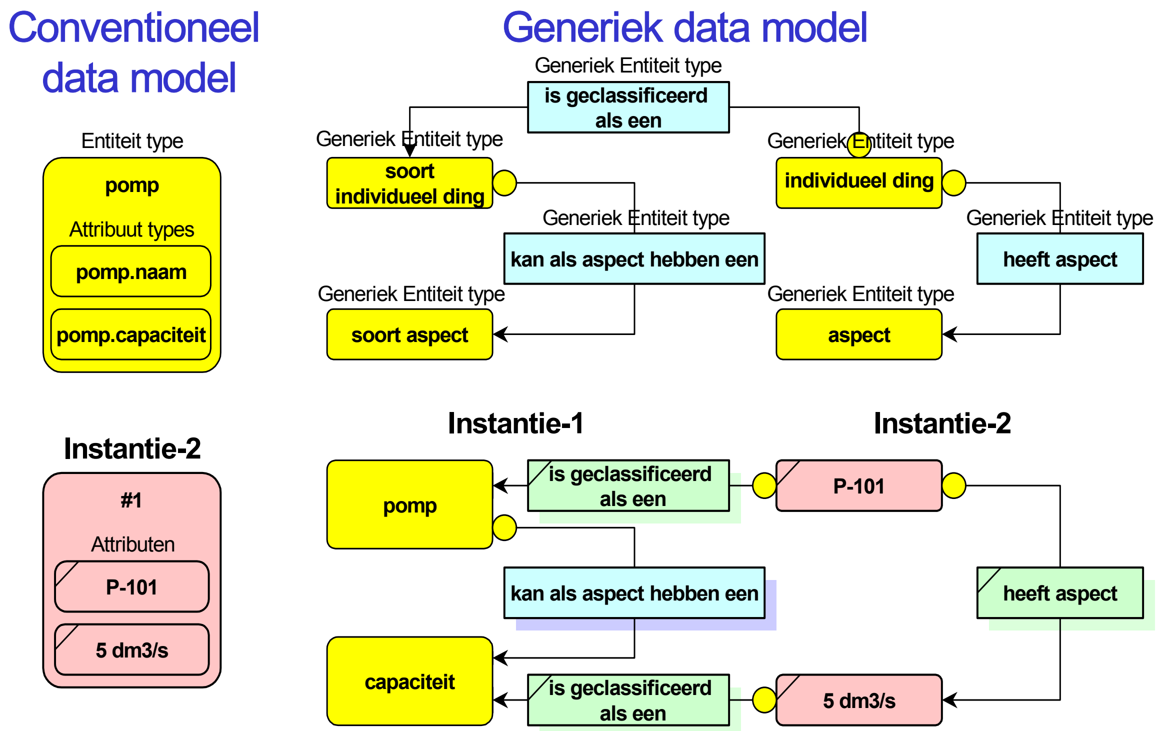
[1] Een ontologie is het resultaat is van onderzoek naar de aard en de eigenschappen van de dingen. Een ontologie kan weergegeven worden in de vorm van een grafisch model, zoals wordt geïllustreerd door figuren in dit proefschrift.

[2] Door mensen vervaardigde producten.

[3] Gellish is oorspronkelijk afgeleid van 'Generic Engineering Language', maar is verder ontwikkeld tot een taal die ook toepasbaar is buiten de engineering discipline.

De volgende hoogtepunten illustreren het ontwikkelingsproces van Gellish:

1. Allereerst werd een **generiek datamodel** ontwikkeld om de scope van datamodellen uit te breiden tot een algemene toepasbaarheid, zodat het veel verschillende conventionele datamodellen zou kunnen vervangen.
   Het verschil tussen een conventioneel datamodel en een generiek datamodel wordt geïllustreerd door Figuur 1.



Figuur 1, Generiek versus conventioneel data model

   Generieke data modellen omvatten onder andere een expliciete classificatierelatie (de ‘is geclassificeerd als een’ relaties in Figuur 1) tussen twee verschillende generieke entiteittypen. Zulke expliciete classificatierelaties vervangen in principe de impliciete instantiatierelaties tussen de instanties van attributen en de attribuuttypen bij conventioneel data modelleren. De entiteittypen in een generiek datamodel hebben in principe geen attributen omdat attribuuttypen zijn vervangen door entiteittypen, terwijl impliciete en expliciete relaties tussen die attribuuttypen zijn vervangen door expliciet gekwalificeerde relaties tussen entiteittypen. Dit proces resulteerde in een generiek data model met een taxonomie van relatietypen. Het generieke data model omvatte daardoor een strikte specialisatiehiërarchie (subtype/supertypehiërarchie) van alle concepten, eindigend met het meest generieke concept ‘iets’. Het resulterende generieke data model werd gestandaardiseerd op twee manieren in ISO standaards[4].

2. Tegelijkertijd werd een bijbehorende **ontologie** ontwikkeld, die gebaseerd is op een **taxonomie van concepten**, die bedoeld is om kennis over applicatiedomeinen op een flexibele en uitbreidbare manier vast te leggen. De auteur van dit document coördineerde het werk van een grote groep van specialisten in verschillende vakdisciplines, die georganiseerd waren in diverse groepen vakgenoten, om te komen tot overeenstemming over de definities, de taxonomie en de ontologie van de concepten in hun domein. Het resultaat werd vastgelegd in een database van concepten die onderling gerelateerd zijn en die van definities zijn voorzien. Daarbij werd de domeinkennis vastgelegd doormiddel van relaties tussen de concepten. De concepten en definities, evenals de kennis, werden oorspronkelijk gedefinieerd als instanties van de entiteittypen in het generieke data model.

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

3. Vervolgens werd ontdekt dat alle instanties van het generieke data model uitgedrukt konden worden in één enkele tabel. Daarom werd de **Gellish Tabel** ontwikkeld. Een voorbeeld van een Gellish Tabel is weergegeven in de onderstaande tabellen.

101	3	201	7
Linker object naam	**Relatie type naam**	**Rechter object naam**	**Een-heid**
auto	is een subtype van	voertuig	
wiel	is een subtype van	artefact	
wiel	kan een deel zijn van een	auto	
W1	is geclassificeerd als een	wiel	
C1	is geclassificeerd als een	auto	
W1	is een deel van	C1	
W1	heeft aspect	D1	
D1	is geclassificeerd als een	diameter	
D1	kan op schaal gekwantificeerd worden als	50	cm

Dezelfde tabel maar dan met unieke identificatie van feiten en objecten ziet er als volgt uit:

54	2	101	1	60	3	15	201	7
Taal van linker object	**Linker object UID**	**Linker object naam**	**Feit UID**	**Relatie type UID**	**Relatie type naam**	**Rechter object UID**	**Rechter object naam**	**Een-heid**
Nederlands	670024	auto	1	1146	is een subtype van	670122	voertuig	
Nederlands	130679	wiel	2	1146	is een subtype van	730063	artefact	
Nederlands	130679	wiel	3	1191	kan een deel zijn van een	670024	auto	
meertalig	10	W1	4	1225	is geclassificeerd als een	130679	wiel	
meertalig	11	C1	5	1225	is geclassificeerd als een	670024	auto	
meertalig	10	W1	6	1190	is een deel van	11	C1	
meertalig	10	W1	7	1727	heeft aspect	12	D1	
meertalig	12	D1	8	1225	is geclassificeerd als een	550188	diameter	
meertalig	12	D1	9	5279	kan op schaal gekwantificeerd worden als	920303	50	cm

Figuur 2, Voorbeeld van een Gellish Tabel

De informatie die in de bovenstaande tabellen staat zou in vrije vorm Gellish Nederlands weergegeven kunnen worden als:

- een auto is een soort voertuig en een wiel is een soort artefact, terwijl een wiel een deel kan zijn van een auto.
- wiel W1 is een deel van auto C1, terwijl de diameter D1 van wiel W1 gelijk is aan 50 cm.

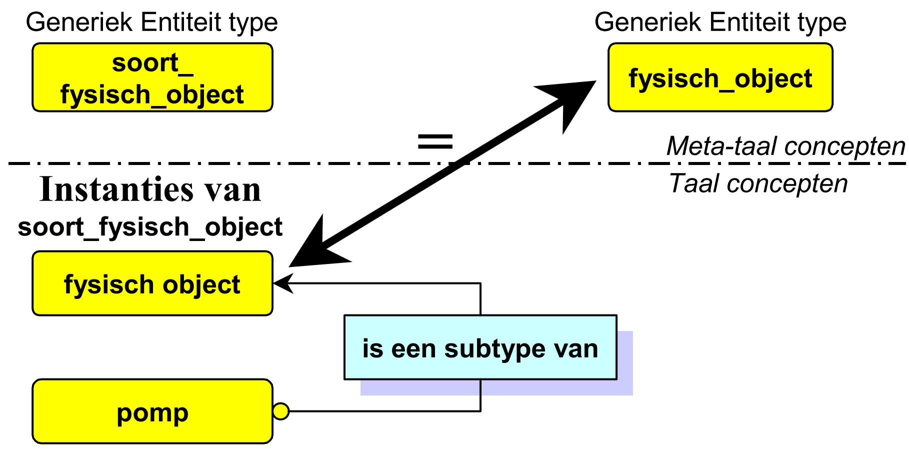
Verder onderzoek is nodig om de toepasbaarheid van deze vrije vorm van Gellish te onderzoeken. Dit proefschrift beperkt zich tot een tabelmatige weergave van Gellish expressies.

De eerste van de bovenstaande tabellen bevat het voor mensen leesbare gedeelte van een Gellish Tabel. Elke regel daarin is de uitdrukking van een *feit*. De tweede tabel is een illustratie van een meer uitgebreide versie van dezelfde Gellish Tabel. Daarin zijn ook de unieke identificaties van de gerelateerde dingen weergegeven, alsmede die van de relatietypen en van de uitgedrukte feiten. Daarnaast is ook de taal aangeduid waarin de feiten zijn uitgedrukt. Als die unieke identificaties zijn natuurlijke taal onafhankelijk. Dit betekent dat de feiten op een natuurlijke taal onafhankelijk manier zijn vastgelegd, waardoor is het mogelijk dat een computer met behulp van een Gellish woordenboek dezelfde tabel ook in andere talen kan genereren en weergeven. Dit wordt geïllustreerd door de bovenstaande Gellish Tabel te vergelijken met Figure 8 in de Engelstalige Samenvatting van dit proefschrift.

De Gellish Tabel is een implementatiemethode (syntaxis voor de Gellish taal) die geschikt is om allerlei soorten feiten in uit te drukken. Bijvoorbeeld:

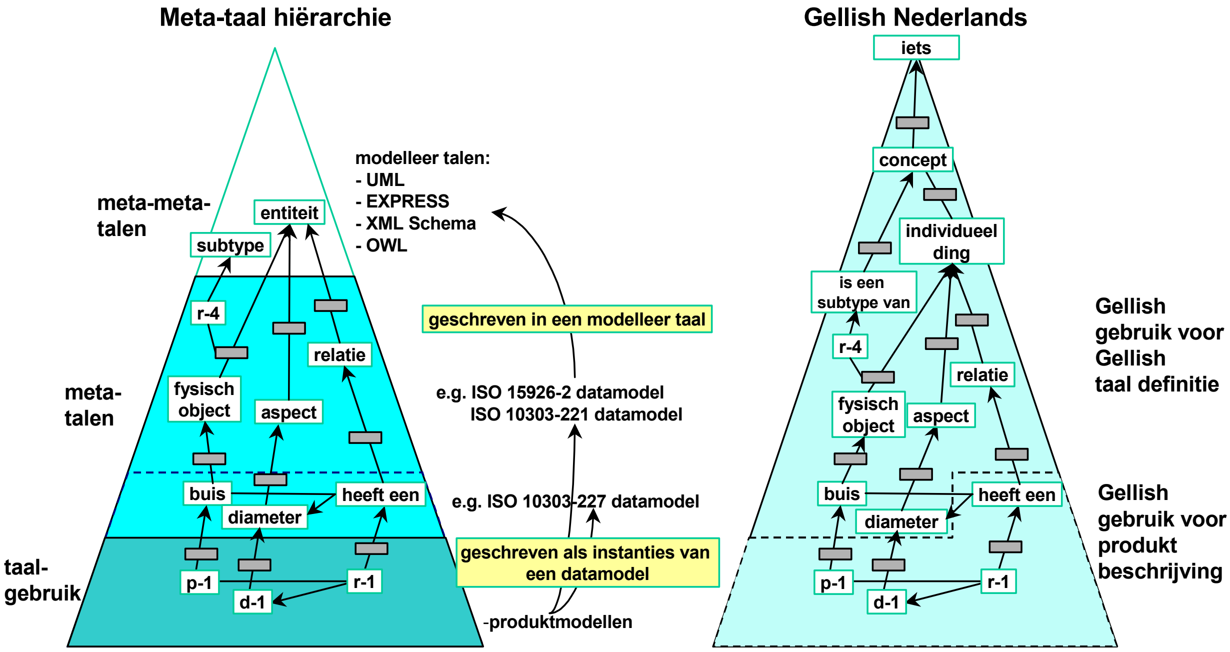
- Om *definities* van concepten op te slaan.
  - zie regel 1 en 2 in de bovenstaande tabel, waarin specialisatierelaties concepten uit een toepassingsdomein definiëren (hoewel details van de definities op die regels zijn weggelaten).
- Om *kennis* uit te drukken door de relaties tussen concepten vast te leggen.
  - zie regel 3 met een voorbeeld van een relatie tussen soorten dingen.
- Om *informatie* op te slaan over individuele dingen.
  Dit betreft feiten en gegevens (instanties), uitgedrukt als relaties tussen individuele dingen onderling – zie regel 6 en 7.
- Om de *classificatie van dingen* alsmede andersoortige relaties tussen individuele dingen en concepten vast te leggen.
  Dit soort feiten wordt uitgedrukt als relaties tussen individuele dingen en concepten (soorten) – zie regel 4, 5, 8 en 9.
- Het bleek dat één enkele tabel voldoende is om willekeurig welk feit of soort feit uit te drukken, inclusief feiten die de definitie van de Gellish taal zelf betreffen.
  De Gellish Tabel representeert een gestructureerde vorm van een subset van de grammatica van een natuurlijke taal. De kern ervan bestaat uit concepten (gerepresenteerd in de bovenstaande tabel door de linker en rechter objecten, met hun identificaties en namen) en Gellish zinsdelen (gerepresenteerd in de bovenstaande tabel door relatietypen, eveneens met hun identificaties en namen).

4. In een volgende fase werd ontdekt dat de generieke entiteittypen in het datamodel (die de concepten uit de metataal vormen) identiek zijn aan de concepten op de hogere niveaus in de ontologie. Die overeenkomst is geïllustreerd in Figuur 3.



Figuur 3, Identiteit tussen entiteit type en instantie

Figuur 3 illustreert dat bijvoorbeeld het generieke entiteittype ‘fysisch_object’ in het datamodel een metataal concept is dat hetzelfde ding blijkt te zijn als het gewone taal concept ‘fysisch object’ in de taxonomie. Dat concept is een supertype van alle soorten fysische objecten in de Gellish Tabel database, zoals bijvoorbeeld het concept ‘pomp’. Bovendien is dat concept (en elk subtype ervan) een instantie van het entiteittype ‘soort_fysisch_object’. Vanwege die identiteit werden de metataal concepten, ofwel de entiteittypen uit het generieke datamodel, toegevoegd aan de ontologie. Zij vormden vervolgens de hogere ontologie, waarna het **datamodel overbodig** werd, ofwel gereduceerd tot een mini 'opstart model' ('bootstrapping model') en werd het datamodel vervangen door een Gellish Tabel met de definitie van de hogere ontologie. De entiteittypen uit het datamodel die aan de ontologie werden toegevoegd betroffen ook relatietypen en typen rollen van diverse aard die nodig zijn in die soorten relaties, welke rollen gespeeld worden door specifieke soorten objecten. Met de toevoeging van soorten rollen werd een consistente specialisatiehiërarchie of taxonomie van soorten rollen gecreëerd. Het resultaat is dat Gellish het conventionele onderscheid elimineert tussen toepassingstaal (gebruikersgegevens) en meta-talen (datamodellen). Ook de concepten die voorkomen in meta-meta-talen waarin gewoonlijk datamodellen geschreven zijn (zoals EXPRESS, UML, XMLS of OWL) konden opgenomen worden in de Gellish ontologie. In tegenstelling met het conventionele onderscheid in meta-niveaus van talen zijn in Gellish alle concepten uit die drie niveaus geïntegreerd in één taal. Dat is geïllustreerd in Figuur 4.



Figuur 4, Gellish als één geïntegreerde taal

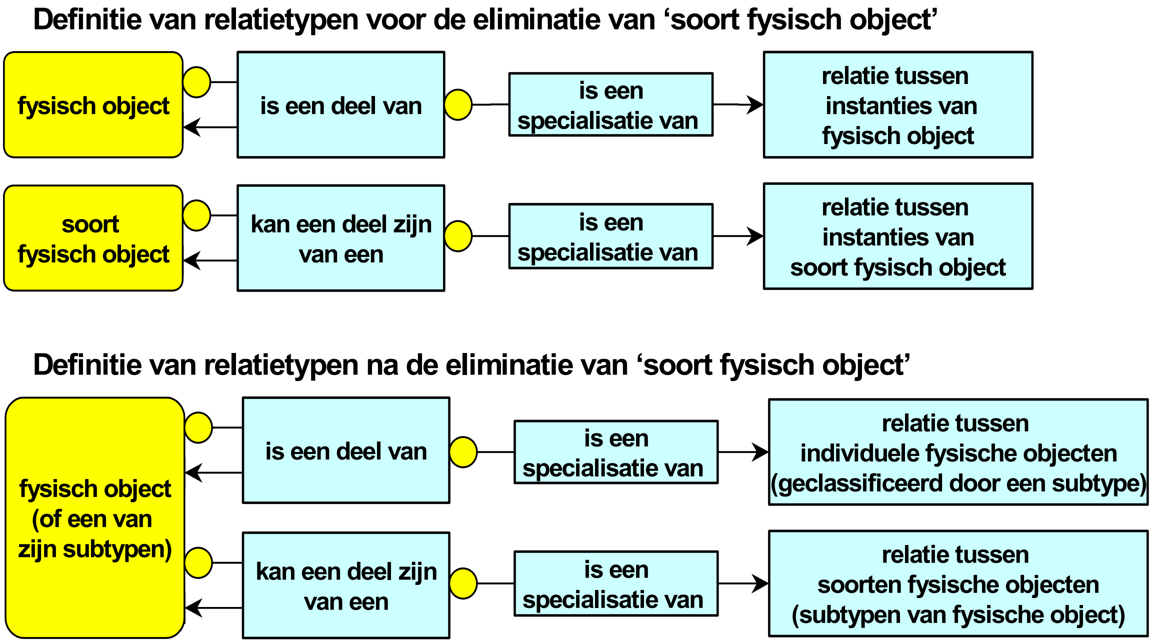
Het linker deel van Figuur 4 illustreert het onderscheid tussen concepten uit verschillende talen (product modellen of gebruikerstalen), datamodellen ofwel meta-talen en modelleer talen ofwel meta-meta-talen. Het rechterdeel illustreert dat alle concepten uit die verschillende niveaus geïntegreerd zijn in de ene Gellish taal.

5. **Algemeen toepasbare browser software**[5] werd ontwikkeld die Gellish Tabellen kan lezen en die de semantische correctheid van de inhoud kan verifiëren en weergeven. Die software bewees dat het mogelijk is om de kennis die in een Gellish Tabel met de hogere ontologie is vastgelegd te lezen en vervolgens te gebruiken om de domeinontologie van vakdisciplines te interpreteren. Bovendien bleek die software in staat om Gellish Tabellen met modellen met informatie over individuele producten en gebeurtenissen te interpreteren, te verifiëren en weer te geven.

6. De volgende stap was de ontdekking dat veel entiteittypen die geconverteerd waren tot concepten in de hogere ontologie semantisch **overbodige artefacten** waren die **weggelaten** konden worden uit de ontologie zonder verlies aan semantische uitdrukkingsmogelijkheden.
Dit is geïllustreerd in Figuur 5.

---

[5] De STEPlib Browser, die in feite een algemeen toepasbare Gellish Browser is.



Figuur 5**,** Eliminatie van semantisch overbodige concepten

Bijvoorbeeld, het concept 'soort fysisch object' in het bovenste deel van Figuur 5 was oorspronkelijk een generiek entiteittype dat bedoeld was om instanties van soorten fysische objecten te bevatten (te verzamelen en/of te classificeren), zoals de instanties 'auto' en 'wiel'. Dergelijke instanties van soorten worden gebruikt om de semantiek vast te leggen over mogelijkheden voor relaties tussen leden van die soorten. Zodra concepten zoals 'kan een deel zijn van een' in de Gellish taal gedefinieerd zijn, kunnen ze gebruikt worden om relaties tussen twee instanties van de entiteittype 'soort fysisch object' te classificeren. Bijvoorbeeld: de 'kan een deel zijn van een' relatie kan gebruikt worden om de kennis 'een wiel kan een deel zijn van een auto' uit te drukken op een manier die door computers te interpreteren is.
Maar het is ook mogelijk en zelfs beter om diezelfde 'kan een deel zijn van een' relatietype te definiëren als een relatie tussen het concept 'fysisch object' en zichzelf, zoals is weergegeven in het onderste deel van Figuur 5, waarbij de betekenis van het relatietype is gedefinieerd als een relatietype die uitdrukt dat een individueel object dat geclassificeerd is als een 'fysisch object' *of één van zijn subtypen* een deel kan zijn van een ander individueel object dat eveneens geclassificeerd is als een 'fysisch object' of als een subtype daarvan.
Deze ontdekking maakte de kunstmatige concepten, zoals 'soort fysisch object', overbodig en daarom werden dat soort concepten uit de ontologie verwijderd.

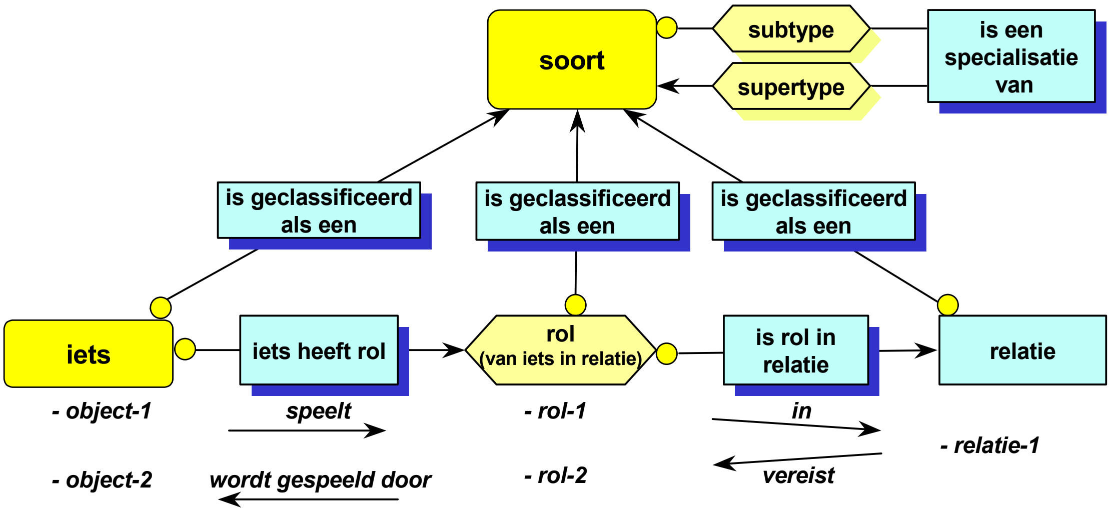
7. Bovendien, concepten die **semantisch overbodige subtypes** definieerden konden verwijderd worden uit de ontologie, te meer daar ze gewoonlijk ook niet gebruikt worden in natuurlijke talen. Bijvoorbeeld, de relatietype 'heeft eigenschap' kon vervangen worden door de algemenere relatietype 'heeft aspect'. Immers een dergelijk subtype dupliceert de semantiek die al vastligt in het feit dat de eigenschap die bezeten wordt per definitie al als subtype van 'eigenschap' geclassificeerd zal zijn en omdat het concept 'eigenschap' als een subtype van 'aspect' gedefinieerd is.

8. Anderzijds bleek het gewenst om **extra subtypen van relatietypen** toe te voegen aan de ontologie om de precieze semantiek van de soorten feiten vast te leggen die aanwezig bleek te zijn in de diverse applicatiedomeinen.

Gedurende de bovenomschreven ontwikkelingen werden de resultaten in overeenstemming gebracht met de concepten die resulteerden uit een analyse van diverse ontologieën en van de natuurlijke taal. Natuurlijke talen zijn waarschijnlijk niet door mensen ontworpen, maar over het algemeen nemen we hun bestaan aan als gegeven natuurverschijnsel en bestuderen we hun structuur en de betekenis van de achterliggende concepten. Dergelijke onderzoeken maken het waarschijnlijk dat de diverse talen overeenkomstige semantische concepten kennen (Wierzbicka, 1996). Een van de conclusies van dit onderzoek is dat die overeenkomstige semantiek wordt gevormd door concepten en relaties tussen concepten waarvoor in diverse talen en culturen verschillende termen gebruikt worden. Gellish is een taal die is opgebouwd uit de elementen uit die **semantische overeenkomsten** tussen talen. De volgende overeenkomstige elementen zijn in de Gellish taal verwerkt:

- **Concepten**.
  Als mensen feitelijke informatie uitwisselen lijken ze gebruik te maken van dezelfde concepten, onafhankelijk van de taal die ze gebruiken, hoewel ze naar die concepten verwijzen doormiddel van verschillende woorden in de diverse talen. Daarom maak Gellish een expliciet onderscheid tussen de taalonafhankelijke concepten en de termen of zinsdelen weermee aan die concepten gerefereerd wordt in de diverse contexten of taalgemeenschappen. Elk concept wordt aangeduid in Gellish door een unieke identificatie (UID) die onafhankelijk is van elke natuurlijke taal. Daarnaast wordt ook in Gellish aan die concepten gerefereerd door middel van termen en zinsdelen uit de natuurlijke talen. Dit gebeurt door middel van een symbool of reeks tekens of door middel van een patroon van symbolen, zowel geschreven als gesproken in een van de diverse talen. Gellish omvat daarom een Nederlands woordenboek, een Engels woordenboek, enz. die als een taxonomie van concepten gestructureerd zijn.

- Een **basale semantische structuur.**
  Er lijkt een **basale semantische structuur** van talen te bestaan die onafhankelijk is van de natuurlijke talen. Een dergelijke structuur is weergegeven in Figuur 6.



Figuur 6, Basale semantische structuur

Die structuur is geïdentificeerd en opgenomen als de basale semantische structuur van Gellish voor de uitdrukking van ieder willekeurig feit. In die structuur worden feiten uitgedrukt als relaties tussen dingen, waarbij die relaties twee of meer rollen vereisen. Alleen dingen van een bepaalde soort kunnen rollen vervullen van de vereiste soorten. Gegeneraliseerde feiten, of kennisgebieden, worden in die structuur uitgedrukt als relaties tussen concepten, of nauwkeuriger uitgedrukt: ze worden uitgedrukt door relaties die rollen van een bepaalde soort vereisen, terwijl zulke rollen gespeeld kunnen worden door instanties van de gerelateerde concepten.

- **Feiten en relaties.**
  Een feit is: dat wat het geval is, onafhankelijk van een eventuele uitdrukking daarvan in een taal. Het concept ‘feit’ is een concept dat gebruikt kan worden om dingen te classificeren als ‘zijnde het geval’. Feiten lijken in talen uitgedrukt te worden als relaties tussen dingen. Daarom gebeurt dat ook in Gellish. Gellish is voor een belangrijk deel gedefinieerd door **soorten relaties** te identificeren die onafhankelijk zijn van de taal en die gebruikt kunnen worden om uitdrukkingen van feiten van dezelfde soorten te classificeren. Een analyse van de soorten relaties die gebruikt worden in ontologieën, in de fysica, in engineering en in bedrijfsprocessen bracht aan het licht dat er een beperkt aantal soorten relaties lijkt te bestaan, waarmee ontologieën, technische artefacten en andere objecten en hun gedrag of gebeurtenissen waarin zij betrokken zijn, kunnen worden

beschreven op een voor computers interpreteerbare manier.
De identificatie van die relatietypen resulteerde in een taxonomie van soorten relaties (of relatietypen) die is opgenomen in de definitie van Gellish. Die relatietypen worden eveneens aangeduid door taalonafhankelijke unieke identificaties (UID's) en per natuurlijke taal wordt er aan gerefereerd door verschillende zinsdelen (frasen). De semantiek van expressies in Gellish wordt gecompleteerd door het gebruik van concepten en relatietypen voor de classificatie van individuele dingen en van relaties.
Veel relatietypen zijn binair, maar gebeurtenissen en correlaties zijn voorbeelden van hogere orde relaties waarin een aantal dingen betrokken zijn. De betrokkenheid van elk van dergelijke betrokken dingen kan uitgedrukt worden als een binaire *elementaire* relatie. Dat betekent dat dergelijke hogere orde relaties in Gellish uitgedrukt kunnen worden als een verzameling van n binaire relaties. De bijzondere rol die elk van die betrokken dingen speelt in zo'n hogere orde relatie kan in zo'n elementaire relatie vastgelegd worden door middel van de keuze van een bepaalde subtype van de elementaire betrokkenheidrelatie.

- **Taxonomie van concepten.**
  Er werd een subtype/supertype hiërarchie of taxonomie van concepten ontwikkeld die ook relatietypen omvat. Het bleek dat er een grote mate van overeenstemming bereikt kon worden tussen de diverse domeinexperts uit verschillende landen en taalgemeenschappen, over die taxonomie en over de definities. Dit resulteerde in een Gellish Engels woordenboek / taxonomie van concepten (die dus ook relatietypen omvat) met een aantal vertalingen van termen en zinsdelen. Dat woordenboek ofwel die taxonomie kan naar believen uitgebreid worden met nieuwe concepten die zowel vertrouwelijk kunnen blijven als dat ze kunnen worden voorgedragen als uitbreidingen van de definitie van de Gellish taal.

- **Ontologie.**
  De hierboven genoemde elementen zijn geïntegreerd in een samenhangend hiërarchisch netwerk dat de definitie van de Gellish taal omvat. Door daarin ook andere kennis op te nemen ontstond een '**kennisbibliotheek**' ('knowledge base') en werd de ontologie gecompleteerd.

De definitie van de Gellish taal is gepubliceerd als 'open source' data en is vrij toegankelijk en te kopiëren op basis van een 'open source licentie'.

Mogelijke toepassingen van de Gellish taal omvatten, maar zijn niet beperkt tot:

- Het gebruik van de Gellish kennisbibliotheek en woordenboek als basis voor een bedrijfsspecifieke 'data dictionary' of kennisbibliotheek.
  Een dergelijke kennisbibliotheek kan bijvoorbeeld dienen als informatiebron in ontwerpsystemen of als referentie voor de harmonisatie van de inhoud van verschillende systemen. Bijvoorbeeld als verschillende systemen vervangen moeten worden door minder al of niet gelijksoortige systemen. Ook kan de Gellish kennisbibliotheek de basis vormen voor een elektronisch intelligent woordenboek annex encyclopedie. De kennisbibliotheek kan uitgebouwd worden doordat extra algemeen geldige of bedrijfsspecifieke kennis in Gellish wordt vastgelegd en wordt toegevoegd aan de bestaande Gellish kennisbibliotheek.

- Het ontwikkelen van productmodellen die systeemonafhankelijk en computer interpreteerbaar zijn.

  Dit betekent dat ontwerpinformatie over individuele producten of producttypen wordt vastgelegd in productmodellen die zijn uitgedrukt in Gellish. Bijvoorbeeld ontwerpinformatie over onderdelen en samenstellingen, zoals apparaten en installaties, wegen, gebouwen, schepen, auto's, vliegtuigen, fabrieken, enz. Dit maakt het mogelijk dat delen van of gehele productmodellen op een systeemonafhankelijke manier uitgewisseld worden tussen verschillende partijen. Ook wordt het eenvoudiger om verschillende beschrijvingen met hun bijbehorende documenten samen te voegen tot één geïntegreerd product model, ook al zijn de delen afkomstig uit verschillende bronsystemen. Het feit dat het gebruik van Gellish impliceert dat gestandaardiseerde begrippen worden gebruikt betekent dat de consistentie van de gegevens wordt vergroot en dat de kwaliteit van de gegevens eenvoudiger en computerondersteund geverifieerd kan worden. Daardoor kan een duidelijke kwaliteitsverbetering bereikt worden. Bovendien wordt het eenvoudiger om generieke applicaties te ontwikkelen die het mogelijk maken om de informatie over verschillende of grote installaties in hun samenhang te raadplegen of om zulke gegevens onderling te vergelijken en te rapporteren. Bijvoorbeeld door het vergelijken van de prestaties van apparatuur op verschillende locaties waarbij de informatie is opgeslagen in systemen met verschillende gegevensstructuren.

- Het beschrijven van het gedrag van producten en de rol van personen en organisaties.
  Dit betreft het in Gellish beschrijven van processen en gebeurtenissen, zowel mechanische als fysische-, chemische- en besturingsprocessen en de rol die personen en organisaties daarin vervullen. Dit soort beschrijvingen zijn beter toegankelijk voor onderhoud, uitwisseling tussen systemen en voor het elektronisch raadplegen ervan. Ook wordt het eenvoudiger om procesbeschrijvingen en productbeschrijvingen te integreren.
- Het vastleggen van algemene en bijzondere specificaties in productcatalogi.
  Dit betekent dat specificaties voor gestandaardiseerde producten in Gellish worden vastgelegd. Bijvoorbeeld vereisten die beschreven zijn in standaard specificaties, zoals gepubliceerd door standaardisatieorganisaties. Het kunnen ook specificaties zijn van typen producten, zoals producttypen uit productcatalogi van leveranciers of van inkoopspecificaties. Zulke beschrijvingen maken het mogelijk dat computersoftware kan assisteren bij het onderling vergelijken van producttypen of het selecteren van producten op basis van specificaties ook al zijn die beschrijvingen van verschillende partijen afkomstig. Dit verschaft aanbieders van producten voor e-business een wijze van vastleggen van productinformatie die eenduidig, neutraal en door een computer te interpreteren is.
- Het beschrijven van procedures en bedrijfsprocessen.
  Zulke in Gellish vastgelegde informatie maakt het eenvoudiger om de beschrijving van bedrijfsprocessen te verbeteren en ze te integreren met andere beschreven processen. Dit geldt in het bijzonder als de beschrijving van de bedrijfsprocessen gedaan wordt door gebruik te maken van een systematische methodiek, zoals de DEMO methodiek. Ook zouden software ‘agents’ ontwikkeld kunnen worden die aangestuurd worden door de kennis die in die procesbeschrijvingen is vastgelegd, waardoor ze automatisch kunnen reageren op binnenkomende berichten.
- Het in Gellish vastleggen van informatie over reële individuele dingen en gebeurtenissen, zoals metingen en waarnemingen.
  Zo’n systeemonafhankelijke vastlegging vereenvoudigt bijvoorbeeld het koppelen van de tijdafhankelijke waarnemingen aan productmodellen die de waargenomen objecten beschrijven en verhoogt de helderheid over de definities van de gemeten grootheden.
- Verbetering van de nauwkeurigheid van de respons van zoekmachines op internet of andere document databases door zowel bij de vastlegging als bij het zoekproces (interpretatie van de vragen) gebruikt te maken van de samenhangen tussen begrippen (keywords of key-facts) die is vastgelegd is in de taxonomie / ontologie van Gellish.

Diverse **voorbeelden** van toepassingen van Gellish Engels zijn beschreven in dit document, zoals:

- Een deel van de kennis, vereisten en ontwerp van een smeeroliesysteem van een compressor.
- Een specificatie van een item uit een product catalogus.
- Een deel van een algemeen bedrijfsproces voor communicatie over bedrijfstransacties.

Tenslotte kan vermeld worden dat de semantiek van de hogere ontologie is samengevat in bijlage A en B. Die bijlagen zijn afkomstig uit in de tabel ‘Gellish Nederlands’ van het ‘hogere ontologie’ deel van Gellish (zie de TOPini file met de Gellish Hogere Ontologie op https://sourceforge.net/projects/Gellish).

Summary

of the thesis
Gellish
A Generic Extensible Ontological Language
Ir. Andries van Renssen

*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[6], and is practically applicable, at least for technical artifacts[7] 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[8], 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.

[6] 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.

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

[8] 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 7.

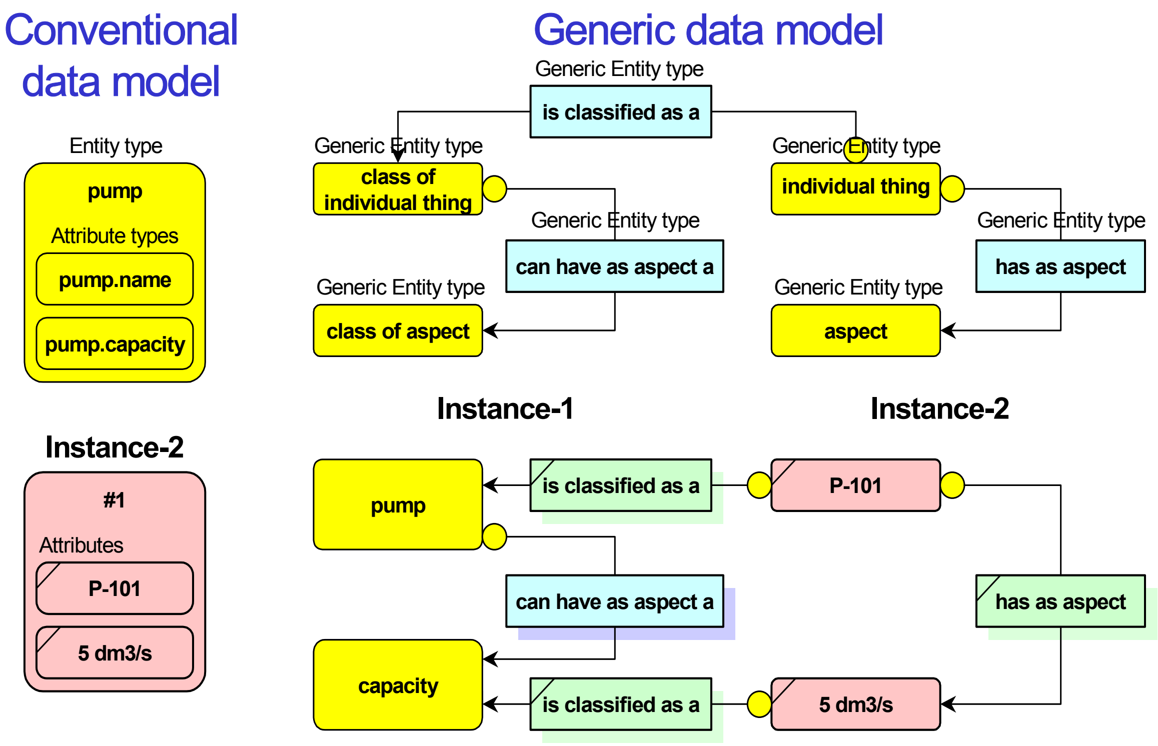


Figure 7, Conventional versus generic data models

   Generic data models include among others an explicit classification relation (the 'is classified as a' relations in Figure 7) 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[9].

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.

[9] 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 8, 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.

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 9.

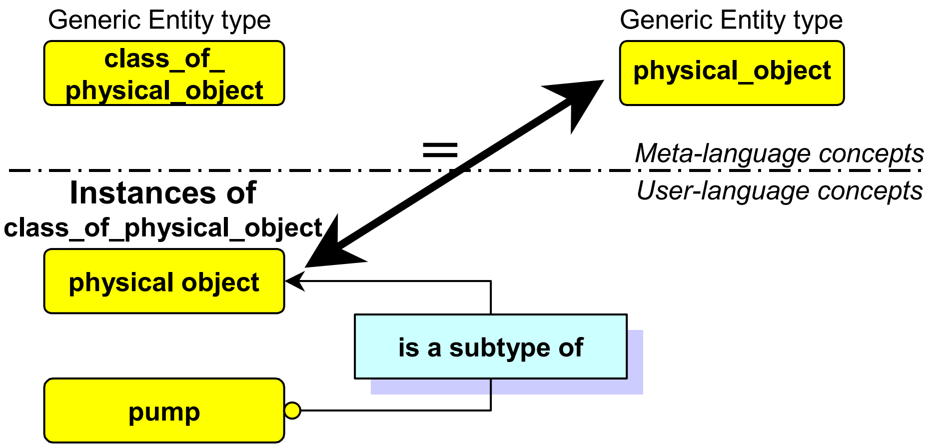


Figure 9, Identity between entity type and instance

Figure 9 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 10.

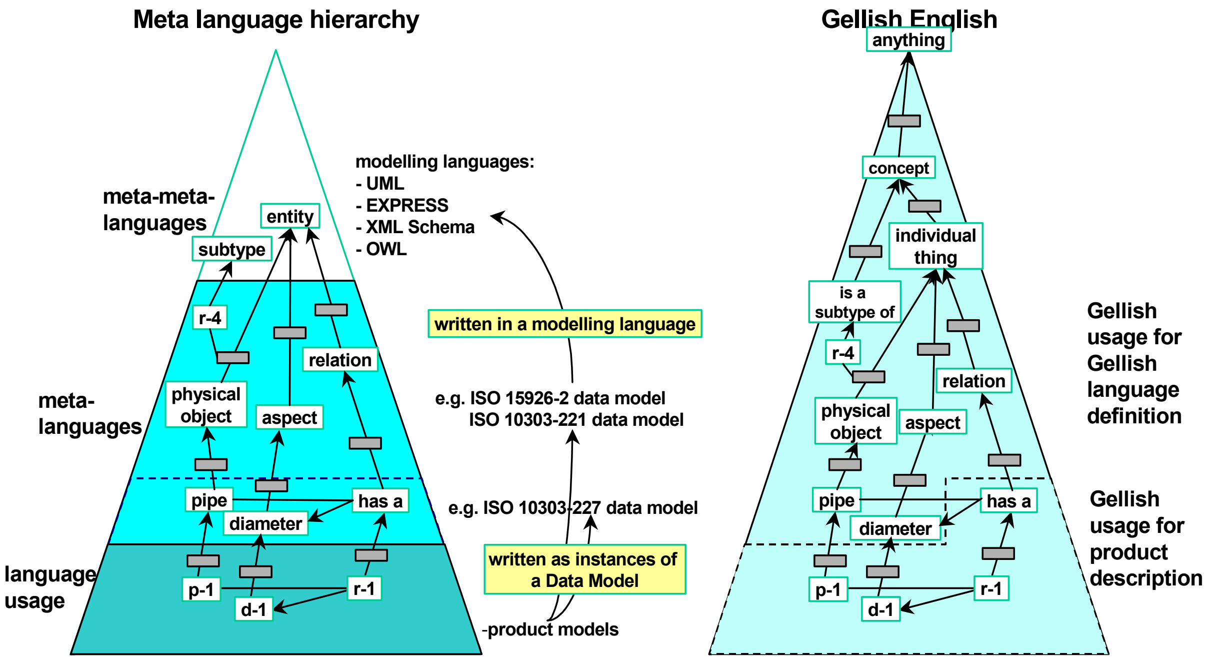


Figure 10, Gellish as one integrated language

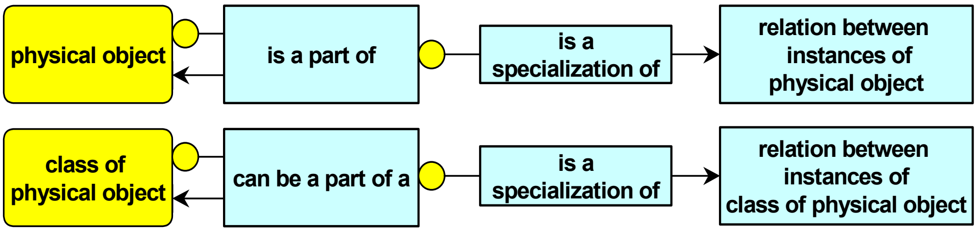
The left hand part of Figure 10 illustrates the distinction between concepts from various languages (product models or user languages), data models or meta-languages and modelling languages or meta-meta-languages. The right hand part illustrates that all the concepts from the various levels are integrated in a single Gellish language.

5. **Generally applicable browser software**[10] 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.

[10] The STEPlib Browser, which is actually a general Gellish Browser. See www.steplib.com.

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 11.

Definition van relation types before the elimination of 'class of physical object'



Definition van relation types after the elimination of 'class of physical object'

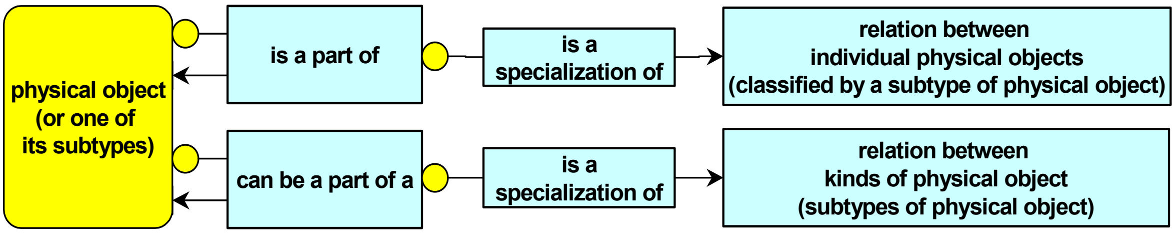


Figure 11, Elimination of superfluous concepts

For example, the concept 'class of physical object' in the upper part of Figure 11 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 lower part of Figure 11, 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 12.

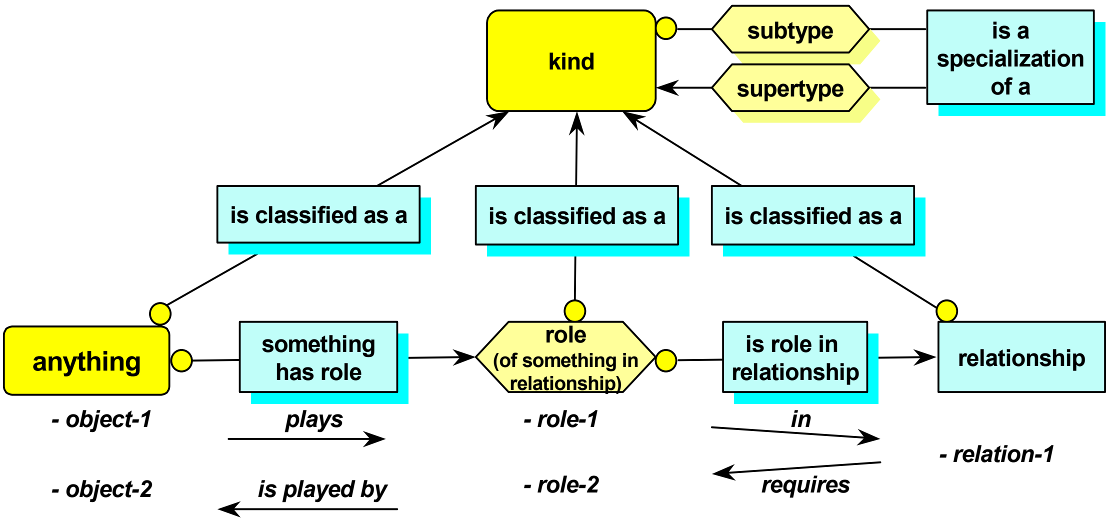


Figure 12, Basic semantic structure

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 document, 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.

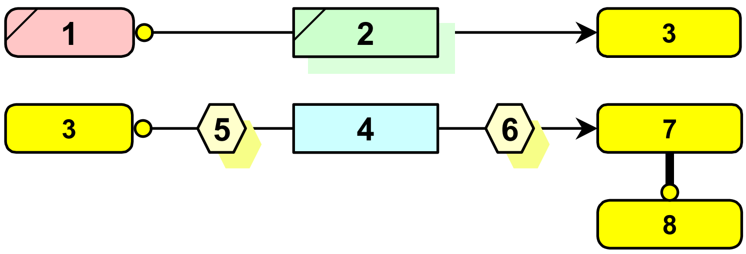
Finally, it should be mentioned that the semantics of the upper ontology is summarized in Appendix A and B. Those appendices are derived from the table 'Gellish English' of the 'upper ontology' part of Gellish. (See the TOPini file with the Gellish Upper Ontology on https://sourceforge.net/projects/Gellish).

Nomenclature

This document uses some symbols and colours in its illustrations. They are not meant as a new notation technique, nor are they normative for the definition of Gellish. Each semantic concept in Gellish is defined by explicit relations. Those relations are typically presented in a Gellish Table form, which is directly computer interpretable. So, the figures and colours are for illustration only.

The figures have a colour/grey scale coding, although they can be interpreted while ignoring the colours/grey scales, because equivalent shape aspects and texts are sufficient for an unambiguous interpretation. Actually the colours are superfluous and only applied to add clarity for those who can see the colours.

The figure below illustrates some basic distinctions between concepts in Gellish. Each concept is indicated by a number.



1 = A box with rounded corners represents a high level concept or a totality or aspect and can be an individual thing (which is red and has a line in the top left corner) or a concept (then it yellow and has no line).

1 & 2: A line in the top left corner of a box indicates that the box represents an individual thing.

2 = A rectangular box represents a relation or a relation type.
Furthermore:
- A green colour and a line in the top left corner indicate that the relation expresses a fact about an individual thing, being either a relation between individual things or a relation between an individual thing and a kind of thing.
- An arrow passing through a rectangular box represents a relation that is an expression of a fact.
- A shaded rectangular box represents a relation and a classification relation between that relation and a kind of relation.
- An expression in a shaded box is a name of a kind of relation.
- An expression requires in natural language a left hand object and a right hand object.
  The circle at one end of the arrow indicates the left hand object in the expression.
  The arrow point indicates the right hand object in the expression.

For example, if the expression in box 2 is: 'is classified as a',
then relation 2 indicates that 1 is related to 3 by relation 2,
whereas relation 2 is classified as a classification relation (an 'is classified as a' relation).

3 = A box with rounded corners without a line in the top left corner is yellow and represents a particular concept (kind of thing).

4 = A rectangular box without a line in the top left corner is blue and represents a relation between two concepts. This can be either a kind of relation that can classify relations between members of the related kinds of things (light blue) or it can be a kind of relation that can classify relations between subtypes of the related kinds of things (medium blue) or it can be a ternary or higher relation (e.g. an occurrence or correlation or a kind of occurrence or correlation) (dark blue).

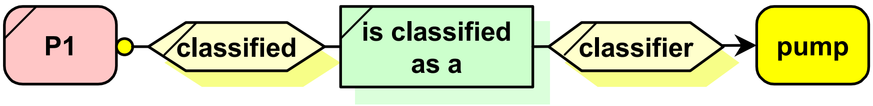
5 = A light yellow hexagonal box (or a text) in an arrow at the side of the circle represents a first role (role-1) that is played by the left hand object. For example, the role that is played by object 3 in relation 4.
If the hexagonal box is shaded, then the expression in the box indicates the kind of role.
Often the roles are not graphically represented as their type can be derived from the definition of the relation type.

6 = A light yellow hexagonal box (or a text) in an arrow at the side of the arrow point represents a second role (role-2) that is played by the right hand object. For example, the role that is played by object 7 in relation 4. The expression in the box indicates the kind of role.

7 = A thick line with a circle at one end is an equivalent of a specialization relation.
- The circle indicates the subtype (8) and the other (7) is the supertype.
- So (8) is a particular concept (kind of thing) that is a subtype of (7) (see example 2 below).
- The inverse means: (7) is the supertype of (8).

These rules are illustrated in the three example figures below.

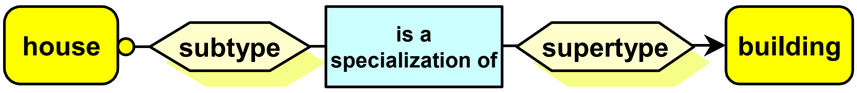
The following figure illustrates the classification relation: P1 is classified as a pump.



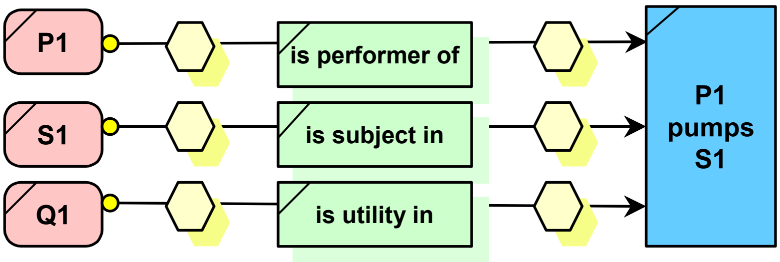
In detail this means:

- P1 has a relation #2 with pump
- relation #2 is classified as a ‘is classified as a’ relation
- P1 has a role-1 in relation #2
- role-1 is classified as ‘classified’
- pump has a role-2 in relation #2
- role-2 is classified as a ‘classifier’

The following figure illustrates the specialization relation: house is a specialization of building.



The following figure illustrates the ternary relation: P1 is pumping S1 using Q1 (while roles are implicit).



1 Introduction

1.1 Objectives and approach

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, and is practically applicable, at least for technical artefacts 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.*

This problem statement is derived from an urgent business issue that there is no common language or general ontology available for data communication between systems. Existing languages and ontologies only provide partial solutions in limited application areas. This hampers organizational cooperation, system interoperability, data integration and knowledge management and sharing.

The prime requirements for a solution of this problem are: the solution should include a generally applicable kernel that is extended with domain specific semantics and is extensible to other specialized application areas, whereas it should be relatively easy to implement.

Because of the generic nature of the problem, it is a scientific challenge to bridge the gap between philosophical generic ontologies, information science principles and practical applications in information technology.

In this research the problem is analyzed from various perspectives, because a solution of the investigated problem requires an integration of knowledge from different disciplines. However, the latest homo universalis (generalist) in history seems to have been Leonardo da Vinci. Since then we only know of specialists. The cohesion between the disciplines is nowadays itself a specialist subject area, studied by philosophers. Therefore, it seems appropriate that the subject is described in separate parts, each from its own perspective.

The study of the structure of reality is the typical subject of **philosophy**, in which the totality and the cohesion of the aspects of reality is studied. Especially the ontological research tries to develop models for that. Ideally this results in one agreed and consistent model. In general, in those models, reality is considered to be one whole, in which whole various aspects and parts are distinguished, which decompose the whole to a certain degree, after which the aspects and parts are put together in a map or model. Therefore, the ontological approach mainly results in models that contain a generic top structure of reality. Because of that we can characterize the philosophical approach in this research as a ‘top-down’ approach.

The various specialist discipline area’s often study details of reality. This results in many detailed specialist models. Generalization and integration of those models is therefore called a ‘bottom-up’ approach. For example, this approach is applied in **natural sciences**, where parts of reality are studied in detail, whereas models of the studied details are produced. But in addition to that, those sciences try to integrate those partial structures in a consistent total structure, especially by describing a set of consistent basic physical laws that apply to the complete reality. Also the **technical sciences** and within that the various technological disciplines develop structures of relative small parts of reality. It is a characteristic of technology that designs of new technical objects, called artifacts, are made. Such imaginary artifacts can be considered as models of the envisaged (imagined, future) parts of reality. Only after it is decided to create a real object on the basis of a design it is in a fabrication process that a part of reality is transformed such that a real object is created and added to reality.
The philosophical ontological models of reality also cover only a limited number of aspects of reality, whereas in general that generic top-structure is not integrated with the detailed ‘bottom-up’ models of natural sciences and technology. As a consequence there are many models of parts of reality and many problems when it is tried to integrate those parts in one whole. A practical example of such a problem is that the decomposition of complex technical artefacts varies per discipline, as well as the fact that the various disciples use different standards, forms, systems, formats, names of concepts, etc. *The Gellish language demonstrates that it is possible to develop a widely applicable ‘model’ or*

'language' with a generic top structure and a consistent and coherent detailed structure, including also a consistent methodology, and that can be practically applied for electronic data exchange. This document also demonstrates that the Gellish language can be used as framework and integrator for several specialized models and that it can be further extended to cover other application areas.

Technologists describe a design making only limited use of natural language. Instead, a design is primarily described in drawings, tables and filled-in standard forms. This means that technical disciplines have created their own expression capabilities. In other words, technologists have created their own '**languages'**. Those languages are in fact natural language extensions that do not only consist of a set of technical terms, but in addition to that they have their own structures. The standards that are developed and applied in the technical disciplines are in fact effort to formalise parts of those languages. Therefore they could be called artificial languages. For that reason, this research will also pay attention to the linguistic aspects of the description of the design and recording of observations on artefacts.

This document describes the results of an analysis of the structural elements of those artificial languages that are used in practice to describe technical artefacts and their behaviour. This analysis resulted in the discovery and development of a set of (universal) semantic concepts that can be expressed in any natural language (so that the concepts themselves are language independent) and to the discovery of a semantic system of kinds of facts (in analogy with the 'periodic system of elements' in chemistry). That semantic system consists of a hierarchically structured collection of kinds of facts or relationships that forms a basic semantic set of linguistic elements. On that basis an 'ontological language' is developed, called Gellish, which is demonstrated to be suitable to describe all kind of technical artefacts and their behaviour. It is called an 'ontological language', because the language is defined in a formal ontology. To some extent Gellish can be described as a structured subset of a natural language, as it has a multi language dictionary with normal natural language terms and a grammatical structure that enables the expression of natural language like sentences. The use of various natural language specific dictionaries results in various natural language specific versions of Gellish, such as Gellish English, Gellish Nederlands, etc. On the other hand, the Gellish English dictionary, which uses natural English terminology, defines **natural language independent concepts** and a natural language independent grammar and semantics, so that the various language specific Gellish versions share the same concepts and grammar, and can therefore be mutually translated by computer software. The Gellish language has the capability to be used to describe other parts of reality than artefacts as well, such as geographic objects and physical phenomena. The language has the potential to be further extended to still wider application areas by extending the basic semantic concepts and the dictionary.

The application of **information technology** results in an increasing exchange of data via the *internet*. To a large extent that exchange uses the 'hypertext mark-up language' (HTML) and increasingly it uses its successor XML or it uses 'attached files' in all kind of formats, among which the standardised system independent STEPfile format. However, these formats only define the *form* of the data and not its meaning or *semantics*. To enable the interpretation of the information by computers it is necessary that an application language or ontology becomes available in which the names and meaning of the concepts in the user's application domain is included and not just the meaning of the IT concepts of data structures. Such a language would enable the exchange of data between computers in a standardised user language. Addition of knowledge about valid kinds of relations between concepts would allow validation of 'sentences' to verify whether they comply with that knowledge. By standardisation of 'interfaces' between systems on the use of such a language it would become possible to integrate data from different sources. This would result in significant cost savings. Furthermore it would become possible that computer programs react automatically on requests of users, without the interpretation of the information by human beings. Such a communication is described by Berners-Lee et al (2001) in the article "The Semantic Web", where it is mentioned that such software 'agents' can only be developed when an appropriate ontology is added to the above mentioned formats. Such an ontology is described in this document.

Nowadays the designs and descriptions of technical artefacts are captured electronically in databases and are exchanged via electronic files. However, it appears that is not possible to exchange data between systems without an extensive conversion process. On the contrary, for each new system and every new interface it appears to be necessary to design a new database and a new interface data structure. This is caused by the methodologies that are commonly applied in information technology. Because software developers apply methodologies to develop the structure of databases and interfaces that result in the fact that each database and every interface has its own (data)structure. Detailed evaluation of what the definition of a data structure is, reveals that it is in fact a definition of a specialised language for a limited application area. As a consequence, each database or interface designer in fact defines his *own language* for the representation of the model of the part of reality for which the database or interface is intended in the way he has understood it through a study of the 'universe of discourse'[11].

Different artefacts and parts of artefacts are designed in practice by using various kinds of software systems. As a result of this situation those designs are captured in different 'languages' and as a consequence they cannot be integrated nor exchanged with other systems without an extensive conversion exercise. Therefore, data integration is a fundamental problem in information technology and it is recognised that for every data communication between systems a data conversion is necessary.

A main objective of this research is to describe a new methodology in the information technology that addresses this problem and that could lead to the use of one single data structure (language) for the description of all kind of designs and business processes and for the description of a large variety of parts of reality and their operation.

That generic data structure is mainly detailed for and applied on the description of technical artefacts and their behaviour.

Because of the various aspects, as indicated above, this research will address the subject from the following perspectives:

- A philosophical perspective.
- An information technological perspective.
- A linguistic perspective.
- A technological and business process perspective.

and within the latter category:

- Various disciplines perspectives, such as mechanical, process engineering, control engineering, civil engineering, procurement perspective, etc.

From each of these angles, I will describe a model of the real world and of a realistic but imaginary world. Especially from the perspective of the technological view point I will provide a number of practical examples. Many more examples are published in the "Gellish Application Handbook" (Andries van Renssen, 2005), which can be regarded as an amendment to this document. Formulated more precisely: this document defines one model or language, but it describes it from various perspectives. The dictionary of that language defines concepts and expresses knowledge about *kinds* of things and processes, so that the language is suitable to describe real as well as imaginary things and processes. The Gellish dictionary and grammar is focussed on the part of reality and imagination that is dealt with in technology, but it is not limited to that part, because attention is also given to the totality of reality and imagination. It should be kept in mind that all this is about one and the same model.

Such a model has different names in different disciplines and different conventions are used in those disciplines to document such a model:

[11] Each data model, being a design of a data structure for a database, is in fact a design of a special language. That is evident from the fact that a data model is a coding system with semantic assumptions about the meaning of strings whenever they occur at specific positions in a structure that is compliant to the rules of the coding system.

- In **philosophy** such a model is called an **ontology** that tries to describe in general terms the structure of reality and imagination or a part of it. An ontology is generally documented as a collection of propositions expressed in a natural or artificial language.
- In **information technology** such a model is called a **data model** or **schema** that describes the structure of 'data' about a part of the reality or processes in it. Such a 'data structure' is a collection of relationships between 'objects'. A data model is generally documented in an artificial language. To support understanding it is often also documented in a graphical schema that shows the objects and their relationships. Data models in information technology can be used to model nearly anything, so they include models of any kind of object and its behaviour, including also models of business processes.
- In **linguistics** such a model is called (the definition of) an **artificial language**, with as aspects a grammar and a vocabulary with its semantics. The definition of the structure of such an artificial language has an 'ontological commitment', which means that the language definition is such that it enables the expression of meaningful propositions about the structure of reality and imagination.
- In modern **technology** this kind of model is called an integration of a **product model** and a **process model**. A product model describes the structure and the behaviour of a particular type of 'product' or of an aspect of a product. As technology mainly deals with artefacts, the term 'product' should be interpreted in this view as wide as possible, covering any artefact. This indicates that a technological perspective limits the view to a part of reality and imagination, whereas the total Gellish model covers any thing in reality or imagination. Examples of partial product models that model only an aspect of reality are the '3D models' that are widely used in mechanical engineering and that describe only the spatial (shape) aspects of things.
  Process models describe only the behaviour of things, especially of systems. Examples are process models that can simulate a physical process, such as chemical reactions or fluid flow that can happen in or between objects over time. Other examples of process models are business processes, such as transaction processes.

Each discipline uses its own terminology and method to describe 'its' models. As a result use of a model in another environment usually needs 'translation' of the model description and that hampers in practice the cross fertilisation and the integration of models.
This document demonstrates that it is possible to define a language that is applicable to various disciplines and that can be used to express and integrate various models from those different perspectives. *This is demonstrated by presenting a knowledge model that defines the Gellish language and by presenting a knowledge base that covers various disciplines and that is expressed in the Gellish language.*
The single model is described from different perspectives, which also provides the 'translations'. The model is a further development of a data model in which development I participated as a core development team member and that is documented in information technology standards of the International Standardisation Organisation (ISO)[12].

The terminology used in Gellish is compared in this document with the terminology used in the most important technologies that are currently applied in Information Technology (IT). In the past IT mainly used hierarchical models. Nowadays mainly 'relational data models' are developed, whereas newer technologies apply object oriented data models (according to the 'object-oriented' or O-O paradigm). The Gellish could be called an 'association-oriented' (A-O) data model.
Advantages and disadvantages of the A-O data model are described in comparison with earlier technologies and it is indicated how 'conventional' models can be translated to an A-O data model and how Gellish can be implemented and practically applied.

For the structure of the top of (the hierarchy of) the Gellish knowledge base or ontology a lot came from philosophical considerations. In this sense information technology learned from philosophy.

[12] See ISO 10303-221 and ISO 15926.

A **secondary objective** of this research is to describe how **philosophy** can use this methodology from information technology to facilitate ontological discussions through documenting ontologies in Gellish which then provides a common methodology for the documentation of ontologies. This may improve the understanding and insight in ontologies and may facilitate their comparison, improvement and integration.
Furthermore, this research has as objective to indicate that philosophy of technology could provide an important contribution to the foundation of information technology and the emergence of an ontologically sound and universally applicable data model.

The use of this methodology in philosophy is illustrated by the description of parts of ontologies as ontological (partial) models. This is done by the expression of these ontologies in Gellish. Several elements from those ontological models are compared with and integrated in the Gellish ontology. For example, the ontological model of Stafleu, which is a further development of a part of the ontology of Herman Dooyeweerd (the Philosophy of the Cosmonomic Idea) and parts of the ontological essays of Peter Simons, which are (partly) based on the ontology of Edmund Husserl are discussed, adopted and integrated in the Gellish ontology.

The main focus of this research is in the information technology aspects, because I try to arrive at a model that is practically applicable in information technology, primarily to support the storage and exchange of information about the design and fabrication of, trade in, and use of technical artefacts. A second focus is on philosophy, because I try to arrive at a model that is widely applicable, and therefore has a generic nature, as an integration of various specific partial models.

The philosophical analysis of the structure of reality and imagination has delivered highly qualified ontological models. It would be a shame not to incorporate those results in information technological models. Because of the generic, top-down approach in philosophy, the philosophical discussion precedes the technological discussions on the subject in this research.

The third perspective, the linguistic perspective, stems from the requirement to incorporate as much as possible from the semantic richness of the natural languages in the expression capabilities of the Gellish language and to achieve automatic translation of expressions, making use of similarities in **semantic structures** in the various natural languages. This resulted in the identification of a hierarchy of ‘semantic concepts’ that define the semantics necessary to express facts of various kinds. The resulting Gellish language is a structured subset of the semantics and grammar that is common to natural languages. This document presents Gellish English, but also Gellish Dutch, Gellish German, Gellish Japanese, etc. are or can be developed.

Finally there is an economic incentive of standardisation of the methodology for the storage and exchange of information about objects and their usage, such as in E-business applications. This was the business requirement that led to this research and that is still the primary aimed application of the result. This clarifies the reasoning behind the choice to develop Gellish in a detailed way for the technological disciplines perspectives.

1.2 Related initiatives

The business requirements to exchange information between computers in a ‘neutral’ system independent way was recognised early in the development history of Information Technology. It resulted for example in the OSI reference model (ISO/IEC 7498-1 (1994)), which distinguishes seven layers in a communication process, ranging from the physical layer at the bottom, which specifies a physical connection, such as a cable, between a sender and a receiver to the top ‘application layer’, which specifies the human terminology and semantics of the exchanged messages. Information technology started to standardise the lower level layers and left the top layer to the ‘users’ of the applications. However those users focussed on isolated applications and customised each application with its own terminology, without any significant standardization on a scale wider than one or only a few applications. This research is completely dedicated to that highest, application layer and the semantics of human language.

The fact that the application layer was left to the users of the applications, meant that the IT world did not bother about standardization of the content of the applications and the user community was not aware of the Babylonian confusion of tongues that was created. The high costs of all the resulting 'dedicated interfaces' made a number of people aware of the need for a data exchange standard. This resulted around 1984 in the start of the development of the family of ISO STEP standards (ISO 10303), as a STandard for the Exchange of Product model data, starting with 2D and 3D shape models and extended with other data about products. However, the complexity of the problem caused a very long development time and resulted in a family of closely related, but slightly incompatible standards, each for a different application area. Around 1992 the new technology of generic data modelling was introduced, which resulted in the early years of the 21st century in the generic data models as standardised in ISO 10303-221 and ISO 15926-2, together with a 'reference data library', being standardised as ISO 15926-4. The Gellish language as presented in this document is a further development of those standards. In the mean time modelling languages were developed, such as EXPRESS (which is defined in ISO 10303-11)[13] and the Unified Modelling Language (UML) (Rumbaugh et al., 1998), whereas the development of Knowledge Representation methods resulted in the emergence of Knowledge Representation Languages such as the Knowledge Interchange Format (KIF) (Genesereth and Fikes, 1992) and Ontolingua (Gruber, 1992). Furthermore, developments around the Semantic Web, especially through the work of the World Wide Web Consortium (W3C), resulted in the emergence of Ontology 'mark-up languages', such as RDFS (Brickley and Guha, 2003), OIL (Horrocks et al., 2000) and OWL (Dean and Schreiber, 2003).

OWL is a semantic mark-up language for publishing and sharing ontologies on the World Wide Web. OWL is derived from DAML+OIL Web Ontology Language and builds on the Resource Description Framework (RDF) and the XML syntax. In contrast with Gellish, OWL is not an ontology, but it contains some basic concepts, called 'modelling primitives' that enable to describe an ontology. This means that Gellish has a rich set of semantic constructs, whereas OWL is a relative simple language although it has the capability to be used to define a full language. Secondly, the OWL concepts are themselves not part of an ontology, nor are those concepts integrated in a specialization hierarchy of concepts as is done with the basic concepts of Gellish. This means that the OWL concepts remain distinct from any ontology of concepts that is described in OWL. Finally the names and definitions of concepts in OWL is typical 'IT speak' with artificial terms, invented by the OWL developers. This means that it will be difficult to integrate and harmonise those OWL concepts with existing philosophical concepts and ontologies. Similar observations hold for other languages, such as TELOS, see Mylopoulos (1990).

An excellent overview of these developments is given by Asuncion Gomez-Perez et al in 'Ontological Engineering' (Asuncion Gomez-Perez, 2004), although unfortunately they do not mention the above ISO standards.

The basic concepts of Gellish have some similarity with the Object Role Modelling method for designing and querying database models at the conceptual level, see Terry Halpin (1996). But Gellish goes much further than ORM, primarily because ORM does not standardise its role types (which actually appear to be equivalent with relationship types, because ORM does not distinguish between a role played by an object and the relationship in which the role is played) as Gellish does. This means that ORM is a limited language that enables to describe a full language. Secondly ORM is not an ontology as Gellish is and finally ORM is intended to model on a conceptual level only, whereas Gellish is for modelling on a conceptual as well as on an individual object (or instance) level.

These above mentioned developments assume that application information, such as a product model of pump P-101, is expressed as instances of a meta-language, being a data model, and that that meta-language is written in a meta-meta-language, being a modelling language. The modelling languages, such as EXPRESS, UML, OWL and XML Schema, are meta-meta-languages that comprise concepts such as 'entity', 'attribute', etc. and that are intended to be used to define meta-languages, also called data models or product modelling languages. For that purpose, those modelling languages enable to define entity types, attribute types (slots), etc., such as 'physical_object', 'class_of_physical_object', 'capacity', etc. Thus, the meta-languages or product modelling languages, such as AP221, AP227, ISO

[13] ISO standards are copyright protected and are thus not free available in the public domain. They have to be bought from ISO or from a national standard body, such as NEN, DIN, BSI, etc.

15926-2, etc. consist of entity types and collections of attribute types arranged in those entity types and they are intended to be used as templates for filling-in ‘instances’ that describe products and processes and their properties. These languages *enable* to define objects, functions and relations and ‘application layer’ concepts, however they themselves are not application layer languages. They define only a few tenth or hundreds of concepts that form a basic semantic set of classes, relations (sometimes called properties) and related concepts and do not comprise a dictionary or taxonomy. Their concepts are even explicitly distinguished from natural language concepts by using non-natural naming conventions for concepts. For example, they use names of concepts such as ‘owl:onProperty’ or ‘rdfs:member’ etc. This is illustrated in Figure 13 that compares three different ways to defined the same concepts: three times a definition of the concepts: approval, approver and approved. Once in EXPRESS, once in XML Schema and once in Gellish English.

in EXPRESS:

```
ENTITY approval
    SUBTYPE OF (relationship);
        approved  :  relationship;
        approver   :  possible_individual;
END_ENTITY;
```

in XML Schema:

```
<xs:complexType name="approval">
    <xs:complexContent>
        <xs:restriction base="relationship">
                <xs:attribute name="approved" type="any-relationship" use="required"/>
                <xs:attribute name="approver" type="any-possible_individual" use="required"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
```

in Gellish English:

approval	is a subtype of	relationship	that specifies the individual that approves the relationship.
approval	requires as role-1 a	approved	
approval	requires as role-2 a	approver	
relationship	can have a role as a	approved	
possible individual	can have a role as a	approver	
approved	is a subtype of	related	by being subject to approval by an approver.
approver	is a subtype of	relator	by being performer of an approval act.

Figure 13, Definition of a new concept in EXPRESS, in XML and in Gellish English

The new concepts that are defined in Figure 13 are taken from the definition of a ‘product modelling language’, defined in the ISO 15926-2 data model (with an equivalent definition in ISO 10303-221 (AP221)). This ‘product modelling language’ is a meta-language and the concepts therefore are defined as meta-language concepts. Any actual fact is then expressed as ‘instance’ of a meta language concept. The definitions of those data models are first expressed in the EXPRESS meta-meta-language. The definitions in data models in EXPRESS use the concepts ‘ENTITY’, ‘SUBTYPE OF’, etc. from that EXPRESS language. However, those concepts themselves (‘entity’, subtype of, etc.) do not belong to the ‘product modelling language’, but they belong only to the meta-meta-language and are therefore not defined in ISO 15926-2 nor in AP221, but in ISO 10303-11. A similar story holds for the definition of new concepts that are expressed in XML Schema. Those definitions use concepts such as ‘</xs:complexType>’ from the XML Schema meta-meta-language. The concepts in EXPRESS differ from the concepts in XML Schema, although semantically they are sometimes close to or even synonyms of each other. For example, the concept ‘subtype’ appears in EXPRESS as well as in XML Schema. However, there is no translation (no mapping) available between these meta-languages, for example between the EXPRESS and XML Schema concepts.

Sometimes, similar concepts appear on both levels, the meta-meta level and the meta-level. For example the concept ‘subtype’ as defined in EXPRESS also appears in ISO 15926-2 and in AP221. However, in EXPRESS the term ‘SUBTYPE_OF’ is a name for a concept that is a relationship. This relationship concept is called 'SPECIALIZATION_OF_CLASS' in the ISO standards. Apart from that, there is another concept in these ISO standards that has the name 'SUBTYPE', that is defined as a role that is played by a class and that is required by that 'specialization_of_class relationship! This illustrates how confusing it can be to make a distinctions between languages, meta-languages and meta-meta-languages.

A similar distinction exists between data model concepts (the product modelling language concepts) and user data or product model instances. For example, the concept 'has aspect' (a 'possession of aspect' relation) is defined to be an '**entity type**' of e.g. a version of the ISO 15926-2 data model as well as the AP221 data model. The concepts 'possessor' and 'possessed' are defined in those data models as '**attribute types**' (without supertypes). However, the concepts 'has aspect', 'possessor' and 'possessed' are not instances of a data model and are also not part of a product model. They are actually defined as some of the concepts that define a product modelling language (the AP221 data model or the 15926-2 data model) that is a meta-language for the product model. A product model that is written in such a 'product modelling language', is defined as 'instances' of the data model entity types and in that way the product model *uses* the concepts such as 'has aspect', etc. However, other product model concepts, such as the concepts pipe and diameter, are not defined in the AP221 and 15926-2 product modelling languages. Those product model concepts (domain concepts) are defined as instances of some of the meta-concepts (entity types) of those data models. This is illustrated in Figure 14.

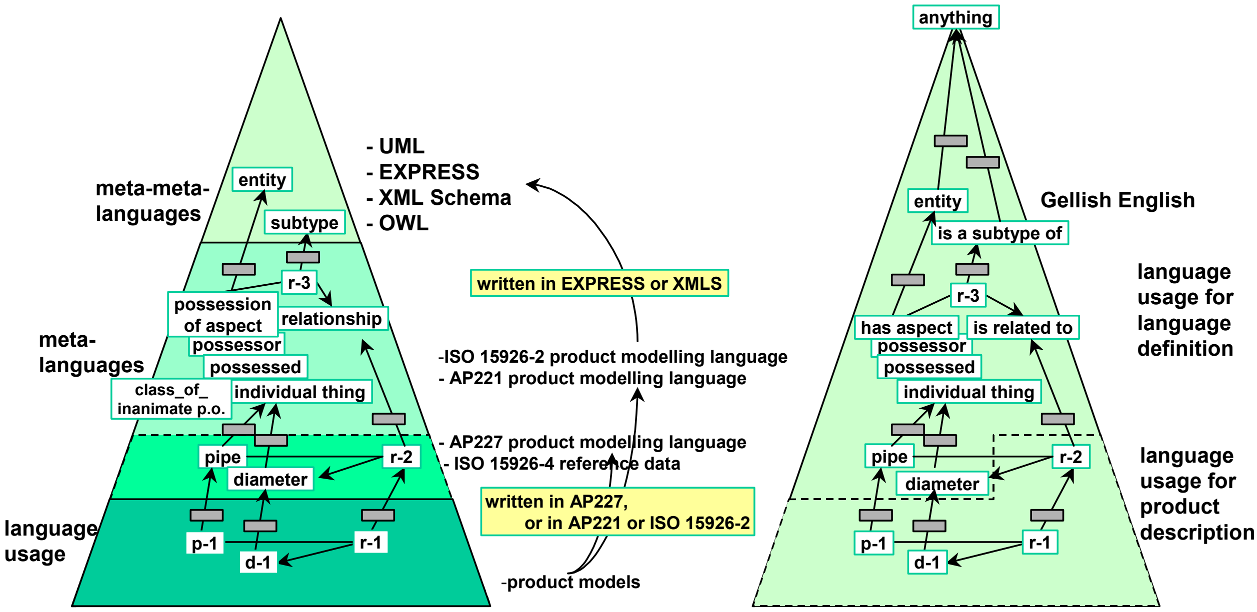


Figure 14, Languages definitions and usage

The left hand part of Figure 14 illustrates the three levels of kinds of languages. Between the lowest and the middle layer there is an intermediate area (with 'pipe', 'diameter' and 'r-2'). In conventional data models such as ISO 10303-227 (AP227) that intermediate area belongs to the middle layer, the language of the software developers. In generic data models such as AP221 and 15926-2 that area belongs to the lower layer, the user language. For example, the concept 'pipe' is a subtype of 'individual thing'. In the ISO product modelling languages AP221 and 15926-2 'pipe' is defined as an instance of the meta-concept (entity type) 'class_of_inanimate_physical_object'. On its turn the concept 'class_of_inanimate_physical_object' is defined using the meta-meta-language concepts 'entity', 'attribute', 'subtype_of', etc. from one or more of the meta-meta-languages.

In more conventional data models, such as ISO 10303-227 (AP227) for piping systems in process plants, the product modelling language mainly consists of concepts from an application domain, such as pipe and diameter, This means that those concepts are defined as part of that product modelling language (data model). These conventional methods imply that the data model (the product modelling language) is constrained by its scope, because only product models can be expressed that use the

limited set of concepts defined in such a data model. For example, the 'AP227 data model' can only be used to describe piping systems[14], whereas AP221 and ISO 15926 enable in principle to be used for any physical object and any aspect and any occurrence, provided that the domain concepts are defined first in or as extensions of their 'class library' (as instances of special entity types).
Anyway, in the ISO product modelling languages the concepts, such as 'has aspect', 'possessor' and 'possessed' are meta-language concepts. They are defined using either of the different meta-meta-languages, such as EXPRESS and XML Schema. But semantically it is irrelevant whether their definition is defined in a meta-meta-language or in a meta-language, or in a domain language! Gellish English does not make that distinction in meta levels. It considers everything to belong to one language domain. Therefore Gellish is one integrated language, just as a natural language, as is illustrated in the right hand triangle of Figure 14.

Thus, although the above mentioned meta languages are called 'languages', they cannot really be compared with the rich semantics of a natural language or even with a subset that is required to describe business or engineering objects. On the other hand, Gellish (with its variants such as Gellish English) is a structured subset of natural language and is not a meta-language, although the domain concepts of Gellish are largely shared with the 'reference data library (RDL) or ontology of the above mentioned ISO standards.

The **application layer** is more directly addressed in the emerging top level ontologies, such as Sowa's top level ontology, Cyc's upper ontology (with some 3000 concepts), implemented in the CycL language and the Standard Upper Ontology (SUO), see Pease and Niles (2002). Those ontologies, address the upper level, generic concepts and therefore do not aim to satisfy the requirements of business applications, in other words they do not define business or engineering concepts. Business level concepts are defined for example in the Suggested Upper Merged Ontology (SUMO), see Pease et al (2002) and in various 'domain ontologies', such as UNSPSC[15], RosettaNet[16], E-class[17], etc. However, the latter 'ontologies' are not integrated with an upper level ontology. They have a fixed number of respectively five, two and four levels in their hierarchy. This means that they are not complete taxonomies, but seem to be a mixture of a subtyping, decomposition, functional (roles) and grouping hierarchy. This limitation means that they have a weak internal structure, measured on the scale of Lassila and McGuinness (2001) as shown in Figure 15. From the more stringent philosophical definition of an ontology they can not really be called ontologies.

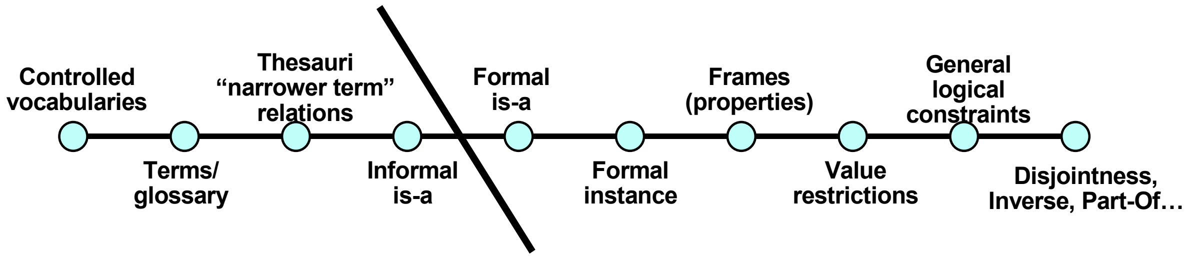


Figure 15, Categorization of ontologies according to Lassila and McGuinness

From left to right the above figure indicates an increasing richness of internal structure of the 'ontology' (assuming that even a vocabulary may be called an ontology). The semantic richness of an 'ontology' is defined more precisely by the concepts and kinds of relations between concepts that are required and used to express the knowledge that is included in the ontology. This is illustrated in

[14] Conventional models often increase the flexibility and scope of their language by creating an attribute type of entity types (often that attribute is called 'type'). This is intended to enable to define subtypes of the entity type. However those subtypes are not really subtypes and therefore do not really extend the data model.

[15] www.unspsc.org

[16] www.rosettanet.org

[17] www.eclass.org

Figure 16, which indicates for some of the categories of ontologies which concepts and kinds of relations should be included.

		1	2	3	4	5	6
Object name	**Gellish synonym**	**Vocabulary**	**Dictionary**	**Taxonomy**	**Knowledge models without product structure**	**Knowledge models with product structure**	**Individual product Models**
language / language community		(x)	(x)	(x)	(x)	(x)	(x)
identifier of concept (or of individual object)		(x)	(x)	x	x	x	x
term (name)		x	x	x	x	x	x
definition			x	x	x	x	
naming relation	is referenced as		(x)	x	x	x	x
alias relation	is a synonym of		x	x	x	x	x
specialization of concept relation	is a specialization of			x			
collection relation	is an element of					x	x
conceptual assembly relation	can be a part of a					x	
scale for characteristic relation	can be mapped on scale				x	x	
conceptual possession of aspect relatio	can have as aspect a				x	x	
quantification relation	is quantified on scale as				x	x	x
requirement for possession of aspect	shall have as aspect a				x	x	
possession of aspect relation	has as aspect						x
assembly relation	is a part of						x
classification relation	is classified as a						x
"other" relations (to be specified)	etc.					x	x

Figure 16, Models with increasing semantic richness

The least stringent requirements apply for a vocabulary or list of terms, in which case the requirements are: an (implicit or explicit) indication of the language used, an identifier for each concept and a name of each concept[18]. The other extreme is a requirement for an extensive grammar of a subset of a natural language, as is required to describe knowledge, requirements and individual phenomena and processes in the imaginary and real world.

> Note that the relation types listed in Figure 16 illustrate an example of a semantic distinction between variations between relation types that are rarely made in ontologies:
> - a conceptual possession of aspect ('can have as aspect a'),
> - a requirement for a possession of aspect ('shall have as aspect a') and
> - an (actual) possession of aspect ('has as aspect').
>
> Similar distinctions are the semantic differences between 'can be a part of a', 'shall be a part of a' and 'is part of', etc.

The semantic richness of an ontology determines which relation types are defined as part of the ontology and which ones are applied in the ontology.
The table in Figure 16 also illustrates that a less rich ontology is in principle a subset of a richer ontology, whereas it should be possible to identify such a subset of a richer ontology by the selection of only those facts in the richer ontology that are expressed through the use of particular kinds of relations that are applicable for the subset ontology.

The Gellish English language version of Gellish that is presented in this document intents to extent the above 'languages' and ontologies in such a way that it intents to provide a complete language at the 'application layer', that integrates an upper level ontology with domain specific ontologies and with the definition of a grammar. This gives the language a sufficient rich semantic expression capability so that it can be used in business practice to express knowledge and to express requirements and constraints, as well as information about individual things and processes. For that purpose, Gellish English includes an English dictionary in the form of a taxonomy that defines a formal subset of a natural language English, including standard phrases that define a formalised English grammar. In

[18] A mark between brackets '(x)' indicates an optional presence of the language indication, explicit identifier or naming relation.

addition to the hierarchy of concepts in the taxonomy it also includes other relations between concepts that specify additional knowledge about the defined concepts.

2 Philosophical perspective

2.1 Ontological models

Since Aristotle various integral ontologies of reality have been developed. In the philosophical tradition such ontologies are usually documented in thick books that are often difficult to access. There is little use of graphical or schematic presentation and thus it is difficult to summarise the resulting structure of reality or to present and enhance such a structure. Furthermore, those structures are usually limited to a top structure that is of interest from the conceptual perspective of philosophers, whereas the structure and relationships of the day to day kind of objects is not really included and integrated in the overall picture. The further detailing is left to the various scientific disciplines, but philosophy has not provided a structured methodology to do that nor to integrate those details in a bigger whole. Because of that situation, such an ontology can only serve the increase of knowledge and understanding, but such an ontology is not practically usable in information technology or engineering. Furthermore it is difficult to formally validate the ontology (by testing its applicability in information systems) or to improve it partially.

Methodologies available in information science for the documentation of 'data models' might be helpful to support the developments in philosophical modelling. This can be seen when it is understood that 'data models' or 'schemes' in information science can be interpreted as ontologies of parts of reality. However, most data models have a limited scope and are significantly more specific than the generic ontologies stemming from philosophy. Nevertheless, *methodologies* from information science are suitable to describe generic ontological models from philosophy. Furthermore, when philosophical ontologies become further specialized, or when data models are further generalized, then those models/ontologies start to overlap and an incentive arises for a fruitful cooperation and synergy. This could improve the scientific ontological basis to data modelling in information science on one hand and might make some results of philosophy more understandable and even practically applicable.

Various philosophers have developed a more or less integral ontology. The common representation of those ontologies in Gellish could have a number of advantages:

- it becomes easier to compare them,
- it becomes easier to systematically improve and extend them,
- it becomes easier to integrate them,
- it becomes easier to make them practically applicable, especially in information technology.

2.2 Language independent ontologies and the unique identification of anything

The philosopher Barry Smith (1999) distinguishes between ontologies that are developed in philosophy and ontologies that are developed and used in information technology ('computational ontologies'). He makes that distinction in order "to address the Babylonian confusion". According to Smith, it would be a characteristic of a philosophical ontology that it is language independent, whereas a computational ontology would be language dependent. He states that ontology in philosophy is a theoretical exercise "which results are the same, irrespective of the language that is used to express the ontology"… "it has as purpose to classify entities". Philosophical ontologies "try, at least in principle, to capture the truth about the things themselves ... within a certain domain". On the other hand, in the realm of information systems, ontology is a "neutral and computer traceable description or theory of a certain domain, which is accepted and can be reused by all information collectors in that domain". It deals with "a single standardised description of *terms*" (*Italic* by me, AvR). "It is a software artefact, designed with the intention of a specific use in a specific computer implementation environment. Because of that, it consists of a specific vocabulary in a programming environment". A computational ontology "deals with languages, descriptions of concepts and with software representations…".
Smith also mentions "compromise positions" between these two kinds of ontologies, but the example

ontologies that he gives are also based on the idea of a common vocabulary. This means that they in fact do not have the particular characteristic which he defined for a philosophical ontology and are thus of the second category.

The distinction made by Smith between philosophical ontologies that are about the essence or nature of things and ontologies in information technology is in my opinion inaccurate. It is true that in information technology the term ‘ontology’ is often used to indicate a collection of terms and definitions (or vocabularies), as Smith says. For example, the concept ‘class’ is used in information technology to indicate a kind of collection of *information about something*, whereas it does not indicate a kind of thing only. In my opinion this use of the term ‘ontology’ in information technology is a reduction of the meaning of the concept, whereas I see no need that could justify such a reduction. I am convinced that it is certainly feasible, and even advisable, to develop and use an ontology for application in information technology that does have the characteristic that it is language independent as is indeed a characteristic of a proper (philosophical and computational) ontology. Such an ontology could deliver the definitions of concepts for information technology to serve the development of information systems.

The realization of this might be hampered by the fact that most information analysts are not educated in philosophy, neither are they experts in the discipline that needs an ontology to support the development of a software system. Therefore, they should borrow the methodology to develop an ontology from philosophers and deliver an implementation of it that is suitable from an information technology perspective. In addition to that, information analysts should guide domain experts in such a way that these experts are able to develop a detailed ontology for their discipline in accordance with a consistent and ontologically sound methodology.

The essence of the proof of the possibility to make a philosophical ontology applicable in information technology is to demonstrate how language independent things can be represented in a computer and how those representations relate to linguistic terms.

First we need a solution to the philosophical problem of what a ‘thing’ is, or whether anything has a unique identity. Here we take the intuitive position that anything has a single identity, whereas we don’t need to choose between the opinions about the nature or duration of such an identity (see Rene van Woudenberg (2000)[19].

The representation of a thing should not necessarily have the form of a term from a (natural) language, opposed to what seems to be assumed by Smith. ‘Something’, for example a concept, or something that can be observed, can be represented in a computer memory by a memory position, which is the starting position of a ‘bit string’. Such a bit string should be an encoding of and should be interpretable as a meaningless but unique identifier (UID), such as a number, without any further information. In information technology terminology we say: an instance or an entity without ‘attributes’. Such a unique identifier then represents the concept or the observable thing. It can represent it without the necessity of the use of a language: there is no term related to the concept yet. The representing UID is language independent. The unique identifier should be unique within the context of a managed ‘world’ of people, organizations and systems that share the UID’s as representers of concepts and other things.

However, only representing something by a unique language independent identifier is not enough. We need associations, among others with ‘terms’ in various languages, which can be used to point to the concept and to provide information about the concept, in order to become human interpretable and to become of practical use. And we have to solve the problem of homonyms, being terms that point to various things, depending on the context in which they are used.

In order to be able to define associations between UID’s and ‘terms’ in a computer we need other computer memory positions that are the starting positions of ‘bit strings’ of which it is defined how

[19] Some philosophers are of the opinion that things are ‘bundles of aspects’, so that the identity would be a plurality. Others are of the opinion that an identity is a spatio-temporal state or part of the universe. In the latter view continuity is seen as a sequence of discrete states. In any case, when we mean ‘something’ we can say that that something is a unity, which is the subject we communicate about.

their pattern is presented on paper or on a visual display screen as characters or symbols. In that way some of those bit strings are presented according to the rules of natural languages or of coding systems, by which they are coded as ‘character strings’ that can be interpreted by human beings who belong to the appropriate language community such that they can understand that language, dialect or coding system as ‘terms’ (including ‘phrases’), that have a role as ‘names’ of ‘individual things’ or of ‘classes of things’.

Once the terms and phrases are available, a person who knows which things are referred to by the UID’s can create associations between those representing UID’s and various terms in the form of a table of pairs of pointers to the UID’s and the terms. These ‘naming relations’ define how something is ‘called’ or ‘described’ in the context of the various languages, although of course the things themselves are language independent. Each of such naming relations is therefore only applicable in the context of a particular language community, which can be a community that speaks a normal natural language or a discipline language or sub-culture dialect.

It should be noted that the bit strings that represent the terms and phrases don’t need UID’s, because they themselves are already unique. It may be that a bit string that represents a spoken word sounds identical in different languages or that a bit string that represents a written character string is identical in different languages. For example, the character string ‘room’ can be interpreted in English as well as in Dutch, with completely different meanings. However, the string (the pattern) is unique and when the string is used in the context of the English language speaking community to refer to a concept, it is unambiguous that it refers to a concept of some kind of space or chamber. Similarly the interpretation is unambiguous when the string is used in Dutch to refer to a concept.

This illustrates that a bit string that refers to a term or phrase actually refers to a class or kind of encoding aspect (a qualitative encoding aspect). That aspect is common to all written or spoken individual terms or phrases that use that term or phrase. It is typically used to classify individual encoding aspects of individual texts. In other words, an individual term refers to a kind of thing, being a common qualitative aspect, that can have a role as a name of a thing.

The representing UID can be further associated with various other UID’s, each of which is a representer of something else. Together those associations (also called ‘relationships’ or ‘relations’) form the context for the interpretation of the meaning of the first ‘thing’ and provides information about the ‘thing’.

An information structure consists of two parts: a human ‘readable’ part and a structure of relations. The bit strings that can be interpreted as ‘names’ provide the human readable part of the semantics for the interpretation and understanding of a data structure. Each relation in the pattern represents an aspect of the thing that is related. A number of those relations together provide a structure that represents the logical part of the definition of the thing or the information about the thing.

Figure 17 illustrates a structure for many language specific references to a language independent concept, as it can be expressed as a structure or coherent collection of binary relations in Gellish.

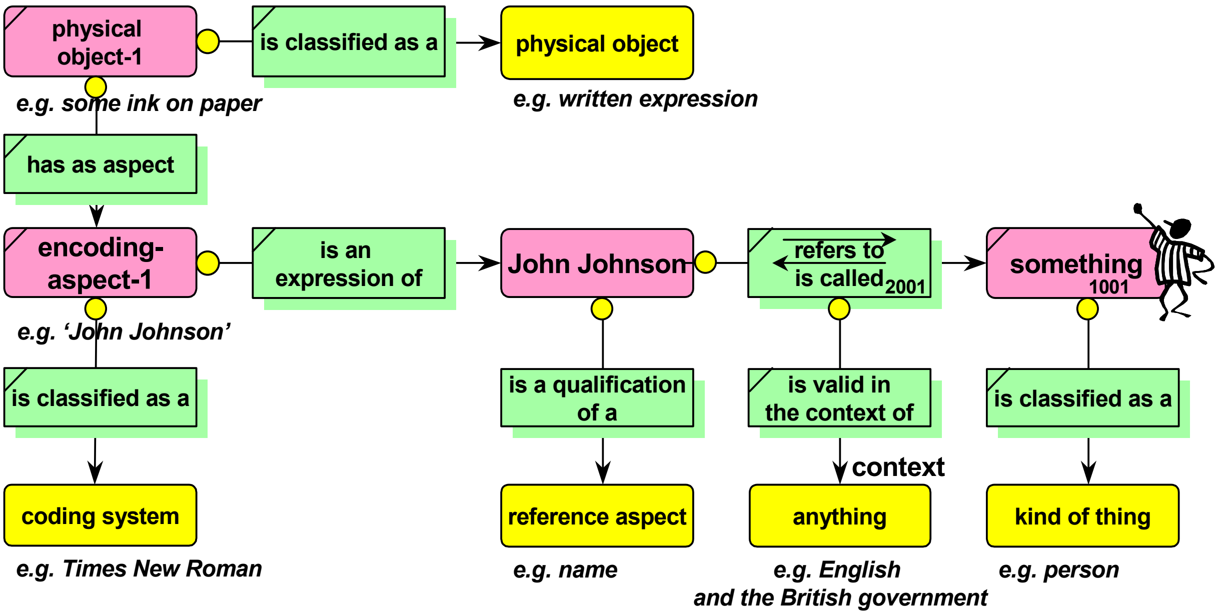


Figure 17, A model for a multi-language reference to a language independent concept

The model of Figure 17 illustrates how the representing UID of something is distinguished from the names of it in various languages and further contexts and is also distinguished from the various individual expressions (being shaped artefacts).

In Figure 17, the box on the right hand side with the label 'something', which has a particular UID (say 1001), represents a particular thing in a language independent way. That thing can be either an individual thing or a kind of thing. For example, the person John Johnson, the pump P-101, the unit of measure kg, or the kind of thing 'person' or 'pump', etc. There is a fact about the 'something', which relates its identifier to a qualified reference aspect, such as John Johnson, which qualified reference aspect can be a qualification of a name, abbreviation, code, etc. That fact (with fact ID say 2001) is represented by a reference relation between UID 1001 and a particular concept 'information' or qualified 'reference aspect', which relation expresses that UID 1001 *is called* 'John Johnson' or 'P-101' or 'person' or 'pump' etc. In inverse wording: the name or code *refers to* the representing UID of the thing. Furthermore, that reference relation is only valid within the context in which the thing is referred to with that name. That validity constraint is defined by a constraining relation that points to that context, for example a governmental registration system and a language context. The language context for some names of things, such as persons, countries and units of measure, is 'multi-lingual', for class names the context will be a particular natural language or language community.

Fact 2001 is represented by the main naming relation, and is associated with two auxiliary relations (the relations to the language and to the validity context). These 3 relations represent a 'molecular fact'. Each of such a molecular fact is expressed on one row in a Gellish Table[20].

[20] The Gellish Table definition will be discussed in more detail later.

The following Gellish Table provides examples of three of such molecular facts.

UID of fact	Language context	UID of thing	UID of relation type	Name of relation type	UID of naming context	Name of naming context	String (name of thing)
2001	multi-lingual	1001	1770	is called	990005	government	John Johnson
2002	English	1002	1770	is called	191152	engineering	pump
2003	Nederlands	1002	1770	wordt genoemd	191152	techniek	pomp

Row 1 illustrates how it is expressed that the person with UID 1001 is called John Johnson in various languages (multi-lingual). On row 2 and 3 an example is given of different names of concept 1002, which is called 'pump' in English (which term originated in the engineering discipline) and is called 'pomp' in Dutch (Nederlands).
Note that the concept 'is called' has a language independent UID of 1770.
Furthermore, the reference aspect (e.g. John Johnson's name) is a qualitative name that is the common concept of all expressions of that name. Therefore it has a qualification relation with name. In other words, it is a qualitative reference aspect that is an abstract concept that can be represented as various encoding aspects that are expressions, encoded in various formats.
For example, the name can be encoded as audible or readable physical phenomena, that can be observed by human beings. Only the expressions comply with a coding system. Thus a qualitative reference aspect is a thing that can be expressed in various ways in various languages as encoding aspects of physical objects. Those physical objects that possess the encoding aspects can also have other aspects, such as a colour or a loudness.
The inverse of the expression relation can be called an interpretation relation, because a reference aspect is an interpretation of an encoding aspect. For a further discussion on the relation between terms and meaning see Martin Stokhof (2000).

2.2.1 Synonyms and Homonyms

A thing, represented by its UID, can be related to more than one reference aspect, each by an explicit relation. Different reference aspects that refer to the same thing are called each other's synonym. Synonym references are valid either in the same or in different contexts.

A 'term' that has more than one meaning has polysemy. The use of such a term to refer to different things is called a homonym reference. In other words, homonym references are relations between different things and one particular term by which multiple things are referred. The differences between the things can only be derived from their context. This means that the term has different reference relations with different things with different unique identifiers. Each of those reference relations apparently becomes meaningful only in its own context.
The relations in Figure 18 illustrate this.

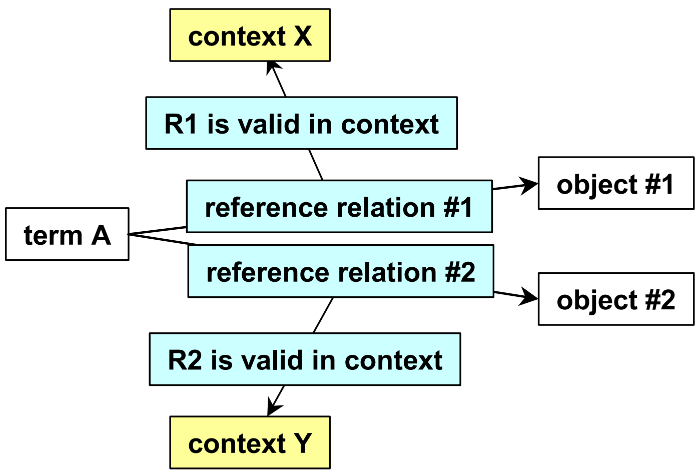


Figure 18, A term as reference to different objects in different contexts

An example of a homonym relation is the use of the term ‘vessel’. It can be used to indicate a subtype of ship or the term can be used to indicate a subtype of container.

This example is depicted in Figure 19 using the general structure for homonyms as presented in Figure 18.

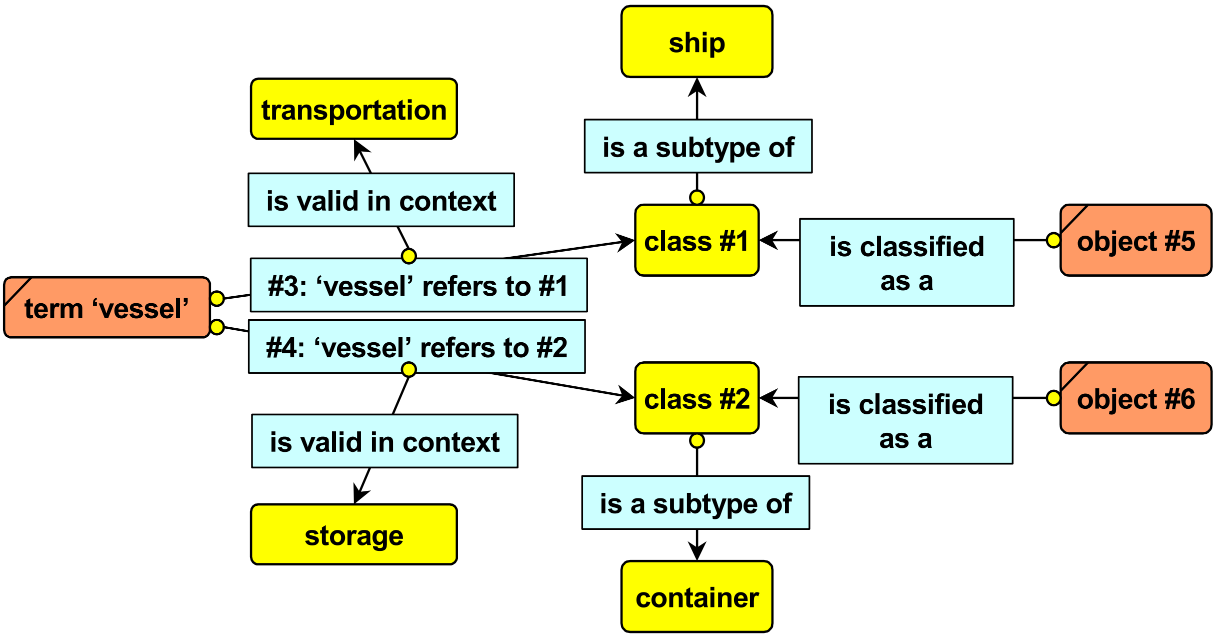


Figure 19, Homonym references to different objects depending on their context

In Figure 19 the term ‘vessel’ in English is used to refer to a class (1) that is a specialization of the class ‘ship’, which reference is valid in the context (X) of transportation. The same term ‘vessel’ is used to refer to another class (2) that is a specialization of the class ‘container’, which reference is valid in the context (Y) of storage.
In Gellish English these two facts are expressed as follows:

Relation 3: ‘vessel’ refers to (1). Relation 3 ‘is valid in the context of’ transportation
Relation 4: ‘vessel’ refers to (2). Relation 4 ‘is valid in the context of‘ storage

If we have two individual objects, a particular ship #5 and a particular container #6, both classified as vessel (but with different meanings of the term ‘vessel’), then in Gellish English they will be classified as follows:

(5) is classified as a (1). This class (1) is called ‘vessel’ in the context of transportation.
(6) is classified as a (2). This class (2) is also called ‘vessel’, but in the context of storage.

So, the structure that is described above serves multiple naming of objects, including abbreviations and codes. It also solves the issue of homonyms and synonyms. As a language or language community, such as a dialect or discipline language is also a context, it is also suitable to document names of things in various languages, even for cross-language homonyms, such as the term ‘wet’ which has the same pronunciation, but a different meaning in English and Dutch.

This demonstrates how things that are identified in a philosophical ontology can be represented in a computer in a language independent way, and how the representing UID’s can be **associated** with names in various languages. This makes a philosophical ontology usable in information systems.

2.3 Integration of ontologies

The main structures in ontologies are generally represented as subtype/supertype hierarchies, also called specialization/generalization hierarchies. In addition to that, ontologies often identify cross-relations between the concepts in a hierarchy. For example, composition relations, also called part-whole relations, possession of aspect relations, etc.

In the philosophical literature and in knowledge modeling various specialization/generalization hierarchies of phenomena are documented, but an integrated complete ontology, which covers the whole reality, requires that also a specialization/generalization hierarchy of kinds of relations, occurrences and correlations is integrated in the whole ontology.
There are several philosophers who have developed a more or less integral ontology. Those ontologies can in principle be presented in a similar way as the Gellish ontology, whereby common components can be made identical. Such a common presentation would have the following advantages:

- Insight in the structure and cohesion of the ontologies will improve.
- The ontologies become better comparable and thus easier to improve.
- They are better accessible for systematic extension and usage.
- It becomes easier to harmonize and merge their content into one integrated ontology.
- By an improved quality and addition of further subtypes they obtain an increased practical applicability.

The process of comparison and merging ontologies can be executed stage wise, because software can assist to visualize ‘branches’ of the ontologies that are formed by subtype hierarchies of the same supertype kind of thing. Then commonalities and differences can be identified and improvements can be made relation by relation. In our practical experience it appeared to be easier to merge bottom up (from common subtypes to agreement on the definition of the common supertype) than top down. This is probably caused by the fact that it appeared to be easier to identify the commonality between concepts that are defined by a larger number of constraints (by identification of the discriminating kind of aspect) than to get agreement on the definition of the generic concepts that have less constraints. Maybe this is caused by the fact that abstract thinking is more difficult than concrete thinking and because people tend to define abstract concepts with some more concrete concepts in mind.

Ontologies can be made practically applicable for the structuring of knowledge, for example, in knowledge bases and search engines on the internet or for the structuring of product catalogues in e-commerce or as a generic data model in database technology. To make one ontology applicable in various applications it is required that such an ontology forms a widely accepted structure and is based on an unambiguous methodology for extensions. This might be achievable if we could agree on a methodology that allows for systematic enhancement on a common work-ontology. This could possibly lead to more focus in philosophical discussions, as it would capture improvements and extensions in an ever growing and improving common model. Such a model or ontology could provide great benefits to the information society.
This research led to a proposal for such a methodology and presents an initial content for such an ontology, by presenting a base ontology and by inviting people to integrate their ontologies and discipline specific structures with it in order to stepwise develop a common systematically structured ontology. This document provides examples from various partial ontologies that are compared with and integrated in the Gellish ontology.

However, in literature about ontologies in information science, especially in the world of artificial intelligence, it seems to be a widely accepted idea that every application area should develop its own ontology, without integrating them. This is illustrated by the article of Smith which was already cited above. He states: Each scientific discipline will *naturally* have its own preferred ontology, which is determined by the discipline language and the generally accepted formulations that are used in the discipline theories (Italic by me AvR). In information science it is a common practice to limit analysis for the design of systems to a particular application area. Even when such an area is called a “Universe of Discourse” (see van Griethuyzen, 1982), the practice is far from an analysis of the universe. This has as consequence that the basic question is whether it is necessary and possible to work towards integration of various discipline ontologies, in order to really address the Babylonian languages problem.

It is clear that Smith in this context points to the various specific discipline languages or jargon that is used in each scientific or technology discipline. It is true that every ‘language community’ has its own

specific concepts and uses its own terminology. Nevertheless, those language communities share a large number of common concepts and within one natural language the disciplines and sub-cultures use a large number of common terms for those concepts. A main challenge will be to develop an ontology of common concepts and to determine how these concepts are named in the various language communities. Such an ontology will form a collection of related *semantic concepts* and the names of those concepts in various languages form the basic commonality of automatically translatable terms between those languages.
This document presents an initial collection of *basic semantic concepts* integrated in a large ontology of semantic concepts that is built up as subtypes of the basic ones.

In linguistics there is a school of development, based on the work of Anna Wierzbicka (see Anna Wierzbicka, 1996) that defines basic 'semantic primitives' as concepts that are used in all natural languages and that cannot be defined by simpler concepts. Those fundamental concepts result from the application of the rule that less fundamental concepts shall be defined by only using more fundamental ('simpler') concepts in order to avoid circular definitions. For the most fundamental concepts there are no concepts left to be used for their definition. This process resulted up to now in a list of some 60 basic or fundamental concepts that cannot have a definition and therefore they shall be intuitively understood by their users. So an ontology based on this approach would have a hierarchy of some 60 top concepts instead of the single top concept ('anything') presented in the Gellish ontology as presented in this document. Therefore the question arises how these 'semantic primitives' relate to the Gellish hierarchy of concepts.
The first observation is that the linguistic school of Wierzbicka uses a pragmatic approach by comparing the concepts present in natural languages and by the discovery that in all those languages the same semantic primitives are shared as basic concepts and as a set of universals (because they are used in all languages). This seems to demonstrate a basis for common understanding in mankind. Anna Wierzbicka even talks about 'universal semantic primitives' and of 'innate fundamental human concepts capable of generating all other concepts' (p. 13 and 16). If the Gellish ontology is complete, then each of these pragmatically identified semantic primitives has (or shall have) a place in the ontology. This document presents an ontology with one top concept, whereas Wierzbicka talks about some 60 top concepts. That difference seems to be caused by the fact that in Gellish *logically* more fundamental concepts are recognized that are not considered to be 'more fundamental' by Wierzbicka, because she uses other criteria to determine whether a concept is basic. For Wierzbicka a concept is not basic if it does not appear in all languages and cannot be defined by other concepts in a definition hierarchy.
However, I don't see why it could not be possible that some languages miss some of the *logically* more conceptual concepts and that those languages only apply the more concrete derivatives. Furthermore, there are concepts that are defined as each other's opposite and thus to some extent they define each other. This means that they depend on each other without one preceding the other in a definition hierarchy. Finally, the concepts in the Gellish subtype/supertype hierarchy are not required to be defined only by more generalized concepts in the hierarchy, but may be defined by concepts (qualifications) from other branches in the hierarchy.
An example of such a pair of concepts is 'big' and 'small'. For Wierzbicka, these two concepts are fundamental, because in her view they cannot be defined by more fundamental concepts. However, in Gellish they are not basic, because they both are qualifications of one more general concept 'size', which on its turn is generalised by (is a subtype of) the concept 'quality', which is further generalized into 'aspect', etc. So, in Gellish size is semantically more fundamental than big and small, whereas big and small are each other's opposite with respect to their qualification of size. This illustrates that 'more fundamental' in Gellish means semantically 'more generalised', irrespective of the question whether there are concepts available in all cultures to give a proper definition.
Another example of a concept that is fundamental according to Wierzbicka is the concepts 'part of'. In Gellish there are two different concepts that are about equivalent to that 'part of' concept: (1) 'is a part of', which relates two individual things and (2) 'can be a part of a', which relates two kinds of things. The first one expresses a fact about two individual things and the second one expresses a possible fact between members of those kinds of things. Both are defined in Gellish as subtypes of a more fundamental concept, called 'relation' (also called 'is related to').
Also the concept 'kind of' is a fundamental concept according to Wierzbicka. This concept is

apparently equivalent to the concept ‘is a subtype of’ (or ‘is a specialization of’) in Gellish. Also this concept is generalized in Gellish by the concept ‘is related to’ and its further generalizations in Gellish.
The above discussion illustrates that there seems to be no reason to adopt the large number of basic concepts according Wierzbicka as the basic concepts of Gellish. However, further research might result in mutual fertilization, because the more generalized Gellish concepts might provide candidate fundamental primitives for Wierzbicka, whereas Wierzbicka’s ‘fundamental concepts’ and her rigorous methodology to create high quality definitions might initiate improvements in the Gellish concept hierarchy.

2.3.1 The solvability of the classification and ontology integration problem

The above conceptual and terminological problem is not only a problem for a scientific discipline or language community, but it is also a problem that causes a lot of misunderstanding for every individual person. Every human being has his or her unique history and experience. That determines his or her memory, language and concepts. The ‘network’ of relations between concepts in his or her memory forms the ‘context’ for every concept. Each new piece of knowledge, acquired through observation or through communication and logic, is added to that network and will be interpreted in the context of that existing network. This means that every human being has his individual context and therefore also his individual definitions of concepts. By communicating we align our contexts and by studying the commonality of our concrete as well as abstract concepts we are apparently able to discover common concepts that are shared in mankind. Therefore I take the traditional position that human beings share common concepts, irrespective the language in which they are referred to. Otherwise communication about factual information would be de facto impossible. However, I think that we should more precisely say that we share concepts more or less, dependent on the extent to which our context networks are aligned. This means that human communication and understanding is a matter of continuous mutual alignment of contexts. Common understanding only emerges when sufficient similarity is achieved in the structures of patterns of relations between the concepts in their contexts. However this alignment is always imperfect. Therefore, it is an incomprehensible miracle that communication and mutual human understanding is nevertheless possible, although it remains a human struggle. This possibility of understanding is also the basis under the belief that a common ontology is possible.

The struggle with terminology in the book ‘Part and Moments, Studies in Logic and Formal Ontology’ (Barry Smith et al., 1986) is a typical example of the difficulties in philosophy with terms, understanding and definition of concepts. That book discusses ‘how one thing can also be many’. With that example I will illustrate how various closely related concepts can be represented in a model and how an integrated model can clarify the discussions.
In his essay about the philosophy of numbers, called ‘Number and Manifolds’, Peter Simons gives account for his choice of the term ‘manifold’ for a concept that is essential to him (see Barry Smith, 1986, page 195 note 2), but about which he says that very few philosophers have recognised the plural things that refer to manifolds. He points to the fact that a plural term, such as ‘my friends’ or ‘Jack and Jill’, refers to more than one thing at once. To clarify the meaning of the concept named ‘manifold’, Simons puts the term ‘manifold’ in the context of related terms, such as ‘set’, ‘class’, ‘aggregate’, ‘collection’, ‘multiplicity’ and ‘plurality’, of which terms, according to Simons, the first four were seen in the 19th century as “roughly synonymous”. Furthermore, he says, that he himself avoids the terms ‘set’ and ‘class’, because Peano and Frege have given a specific meaning to the concepts that are indicated by those terms. However, to really understand the issue it is essential not only to get an understanding about what Peter Simons means with the concept which he indicates with the term ‘manifold’, but it is also necessary to understand how ‘his’ concept relates to the other concepts about which other authors write (as he understands them). This is necessary, because only through this insight in the relationships between these concepts it is understood which concepts are meant. On the other hand Simons does not avoid the terms ‘class’ and ‘set’ completely, because in a next essay he discusses them extensively. It would have been clarified a lot if Simons would have presented the relationships between the concepts in a scheme. Figure 20 presents a proposal for such a scheme, which reflects my understanding of the concepts and relationships according to Simons essay. If Simons would have presented such a scheme, it would have led to a better understanding about the

nature of the relations between the concepts, in a similar way as it can be verified by means of the proposed scheme whether I have understood Simons correctly.

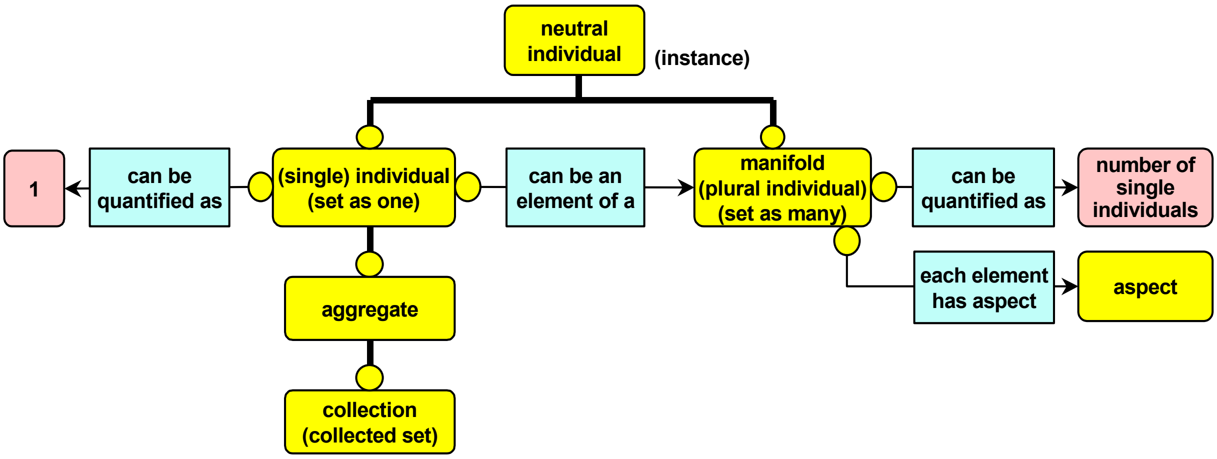


Figure 20, Summary of the philosophy of number according to Peter Simons

The Gellish ontology follows Simons by acknowledging that there is a distinction between single concepts and plural concepts. For example, next to the concept 'child' there is the concept 'children' and next to 'stamp' we have 'stamp collection'. However, Simons only talks about the differences between the concepts and pays insufficient attention to the individual things. Therefore he does not recognize that individual things can have multiple classifications: one for its plurality aspect, the other for the nature of its elements. A particular collection is a single collection with a plurality aspect that is qualified as plural. Therefore the individual thing has two distinct relations:

1. A relation with the plural concept 'collection', which concept classifies the manifold as one plural thing.
2. Another relation with a singular concept, which specifies that 'each element of the collection is classified as a' single thing.

For example, the object that is called 'my stamp collection' is classified as a single collection (one collection of stamps), and it also has another relation with the concept 'stamp' that specifies that each of its elements is classified as a stamp.

In a similar way as the figure above it would have been helpful if Simons would have clarified his definition and use of term 'class' by presenting the relations between the concepts in a scheme. Different philosophers give different definitions to concepts referred to by the same term. Therefore actually they define different concepts. By referring to them by the same name, they actually create homonyms. To clarify the differences between those concepts, it is required that a more detailed scheme which includes those differences would refer to the different concepts by using different identifiers or names to distinguish between the concepts as used by different persons and thus in different contexts. For example, Simons is of the opinion that there are a number of different concepts that are referred to sometimes by the same term and sometimes by different terms. For example, he distinguishes the terms class(Frege), set(Frege), class(Simons), set(Simons), collection(Simons), plurality(Simons), multiplicity(Husserl), etc. all as different, but it is difficult to retrieve from his essay what the relationships between those different concepts are in his opinion. Furthermore, for apparent artificial terms, such as 'class as single' and 'class as plural', a scheme would be helpful to unambiguously distinguish these concepts by clarifying the kind of the relations between them. Without that it remains unclear how many different *concepts* are exactly distinguished by Simons and by others and which *concepts* are indicated by which *terms*. On the other hand it is more or less clear how many *terms* are involved, although many other terms (words) are apparently required (which seem not to require a definition) to indicate whether or not the terms are used as synonyms.
In the ontology of Gellish each concept (which is possibly referred to by various synonymous terms and terms in various languages) has at least one relationship with a more generalised concept and possibly various other relationships with other concepts. Those relationships can be presented in schemes similar to the above figure. It is in principle possible that a computer program can

automatically present such schematic structures, because of the formal definition of the relationships in Gellish.

The above illustrates that an understanding of a context is essential for the interpretation of a term and that the nature of the relationships between the concepts determine whether concepts are really identical or not. Because of this, in Gellish, it is always indicated in which context a term is defined to be a name of a concept.

An analysis as described above enables one to determine which concepts are shared and thus to determine a common collection of concepts and how to merge specialist terminology and terminology from various cultures and sub-cultures as partial additions to a common terminology.

Similar considerations apply to the reasoning why partial ontologies that are developed from a different perspective can be merged. For example, one can distinguish between *object-oriented ontologies,* such as ontologies of (physical) objects*, ontologies of facts* and *ontologies of occurrences or processes*, including discrete as well as continuous processes. A merger of such ontologies integrates facts about objects with facts about the participation of those objects in occurrences and processes. This is also intuitively correct as long as one agrees that information and knowledge from various perspectives describe aspects of the same reality.

Such an integrated ontology consists of definitions of concepts in various contexts and a large varied structure of relationships of various kinds, together with names of those concepts and relationships. Together they form a common 'language' that is necessary to enable a dialogue across disciplines and to communicate about and integrate the results of those disciples.

The large overlap in concepts that are used in various professional disciples is an additional reason to integrate ontologies and to aim for one common ontology. It is self evident that it remains possible to distinguish parts of such a single ontology that are only useful and known in limited communities.

As long as every community keeps developing its own ontology the 'Babylonian confusion of tongues' will only increase. Unfortunately the latter is the case in information technology. In fact every system developer develops for each system a special dedicated language and, when it becomes a requirement to exchange information with other systems, he first learns the language of that other system and subsequently he writes translation software to convert data from one language to the other. It is no surprise that a large part of current software development is devoted to the creation of 'interfaces', which is nothing more than creating such translation programs. This should be superfluous.

The above illustrates the importance of the effort to develop a single common ontology and to define the contexts in which the terms that are used refer to the concepts in that ontology.

A good quality *integral ontology* should not only include definitions of concepts, facts and occurrences, but also collections of relationships that define aspects of objects. Furthermore, such an ontology should be extensible and multi-lingual. Such an integral ontology should be described as an extensible hierarchical network of relationships. The primary hierarchy in that network is a subtype/supertype hierarchy of definitions of concepts. That specialization/generalization hierarchy has a network structure, because a number of the concepts is a subtype of more than one supertype. In addition to this basic hierarchy, the network contains a large number of 'cross-relationships' between the concepts. Those cross-relationships are expressions of the knowledge about facts on the concepts, expressed as '*relations between concepts*'.

2.3.2 The position of individual things in the ontology

In most sciences individual things are only used to discover laws for kinds of things, after which the individual things do not appear anymore in the scientific theories. For the same reason individual things generally do not appear in ontologies. However, the Gellish ontology is intended to be used as a common language. This means that not only concepts (kinds of things), but also generally known individual phenomena belong to the shared awareness and knowledge of a language community. Therefore the Gellish ontology includes a number of generally known individual things and knowledge about those things. They are related to the kinds of things mainly by classification relations. Examples of such individual things are: the earth and other geographical things, such as continents, countries, oceans, cities, locations, etc. and also organizations, such as governments, standardization organizations, companies, etc.

2.4 Summary of basic concepts

2.4.1 Presentation and scope

This document presents the Gellish ontology as a formal language definition in a model that is presented philosophically in words as well as information technologically in a computer interpretable form. The information technological implementation makes use of a formal computer interpretable Gellish Table form. The Gellish Table definition is presented in a later chapter.

The presented model includes a generic upper-ontology that covers the totality of reality and imagination. It includes fundamental philosophical concepts, for example concepts that are already discussed by Aristotle, such as the concepts individual thing, kind (class), concept, part, whole, aspect, single, plural, static, dynamic, etc. The ontology is expanded in more detail mainly for technical artefacts and their relationships and behaviour, as they are in use in practical applications in the current business practice. At the moment, the model contains most details in the broad application area of engineering and in particular in facility design, construction and operation in the process industries and goes to such details as aspects of various kinds of nuts and bolts.

The Gellish language is defined primarily by the definition of a hierarchy of standardized relation types (by defining which roles they require and which things may play those roles) and by the definition of a dictionary and taxonomy of concepts. The following paragraphs discuss the basic concepts in outline.

2.4.2 Basic semantic concepts: Standardized types of relationships

This section discusses the minimum set of basic semantic concepts that are arranged in a basic structure that forms the fundament of the Gellish language and ontology.

The presented ontology reflects my conclusion that information and knowledge is not a property of isolated ‘objects’, but that information and knowledge can only be derived from and expressed by a network of relationships. This is opposed to a way of thinking in a main stream information technology where the information is encapsulated in the ‘objects’, as applied in ‘object-oriented’ modelling. My conclusion is that *the information is in the relationships*. Nevertheless object-oriented modelling has proven to be a useful technique for software development, but relationship oriented modelling has the potential to improve the reflection of reality and to add flexibility to the models, because in relationship oriented modelling the objects are defined independent of their ‘attributes’ as will be discussed in section 4.2.4.[21]

This conclusion is in line with statement 2 of the ‘Tractatus’ of Wittgenstein, where he states that ‘what is the case, the fact, is the existence of atomic facts’, which I interpret as stating that everything that is the case can be called a fact and can be composed of ‘atomic facts’.
This conclusion is expressed in the Gellish ontology by modelling reality and imagination as a network of ‘atomic relationships’, in which each atomic fact is expressed as a relationship between (two or more) things, which relation is classified by one of the standardised relation types that are included in the ontology or that will be added to the ontology.

In that network, anything has a role as a ‘node’ and all facts are expressed as relationships between those nodes. This includes for example the facts that ‘objects’ have ‘attributes’ or ‘properties’, or that objects are involved in occurrences. Those facts are also expressed by relationships between those objects and their aspects or between those objects and those occurrences. A large number of types of

[21] In this study the term ‘object’ is not used as a synonym of ‘physical object’, as something you can kick, but is used as a synonym of phenomenon and can refer to a totality (a complete composite) as well as to an aspect of a totality. In addition to that it can refer to an imaginary as well as to a real phenomenon.

relationships are included in the Gellish ontology as standard relation types as is presented in Appendix B, together with an extensive Gellish English dictionary/taxonomy. This makes Gellish English a semantically rich language. The methodology results in a rich '*semantic network*' that forms a description of (an idea about) the structure of reality and imagination (see also Sowa 1984, p. 76). This goes much further than the Web Ontology Language (OWL) as defined by the World Wide Web Consortium (W3C) and than ORM as is discussed in section 1.2.

A coherent collection of facts will be expressed in Gellish as a coherent network of relationships between things. In such a network there can be two kinds of atomic relationships:

- Relationships that represent information about situations that remain the case without change (these relationships express **ordinary facts** that can be qualified as *'static facts'*) and
- Relationships that express (dynamic) **occurrences**. Therefore, in some sense we can refer to occurrences as *'dynamic facts'*.

Both, static and dynamic facts, can be expressed as relationships between things, each of which related things plays a particular role in the relationship, as is illustrated below.

The following five relations are examples of a coherent network of expressions of ordinary (static) atomic facts:

O1	has aspect	A1
O1	is classified as a	wheel
A1	is classified as a	diameter
O1	is a part of	O2
O2	is classified as a	car

In other words, or better said: in the same words with another syntax:

- wheel O1 is a part of car O2 and wheel O1 has diameter A1.

The other fact type was an occurrence, expressed as a dynamic relationship between a number of involved things. This makes an occurrence in principle an n-ary relationship in which the number of recorded involved things may vary over time.

The following table presents an example of an occurrence (T2) where the involvement of two things in the occurrence is expressed as a collection of two binary atomic involvement relationships and three classification relationships:

T1	is performer of	T2
T1	is classified as a	transformer
T2	is classified as a	transformation
E1	is subject in	T2
E1	is classified as an	electric current

In other words:

- transformer T1 transforms electric current E1 in process T2.

Note that the roles played by the objects in the above relationships are note made explicit in these expressions of atomic facts.

Thus occurrences, processes and activities are modelled as nodes in a network, which enables to vary the number of involved things over time. The Gellish ontology standardises the types of involvement, being the types of roles played by the involved things. It also standardises the types of occurrences, including also processes and activities, being represented by verbs and types of acts that appear in the Gellish Dictionary/Taxonomy.

A definition of a concept consist of a structure of relations with other (logically) earlier defined concepts and may not refer to the concept that is being defined, otherwise circularity of definitions would occur. If such a definition is expressed in natural language, this structure will have the form of one or more sentences, whereas the structure is determined by the syntax rules of the used natural language. However, if it is expressed in the Gellish language then the structure consists of a (relative small) network of concepts and their relationships. For the expression of any definition, including the first one, there seem to exist a *basic semantic structure* that is typical for the definition of concepts as will be illustrated below. Secondly, the concepts used in those structures necessarily form a hierarchy of concepts, because each concept can be defined as a more constrained subtype of a less constrained supertype concept. This hierarchy shall include a basic collection of axiomatic concepts that are required to express the first definition. In other words, to define any concept we need to express one or more facts that together form the definition, whereas those facts are expressed as relationships. The fact that even the first definition requires a structure implies that it is required that the collection of basic concepts also includes a collection of basic relationship concepts.

The *basic semantic structure* of the Gellish ontology is presented in Figure 21.

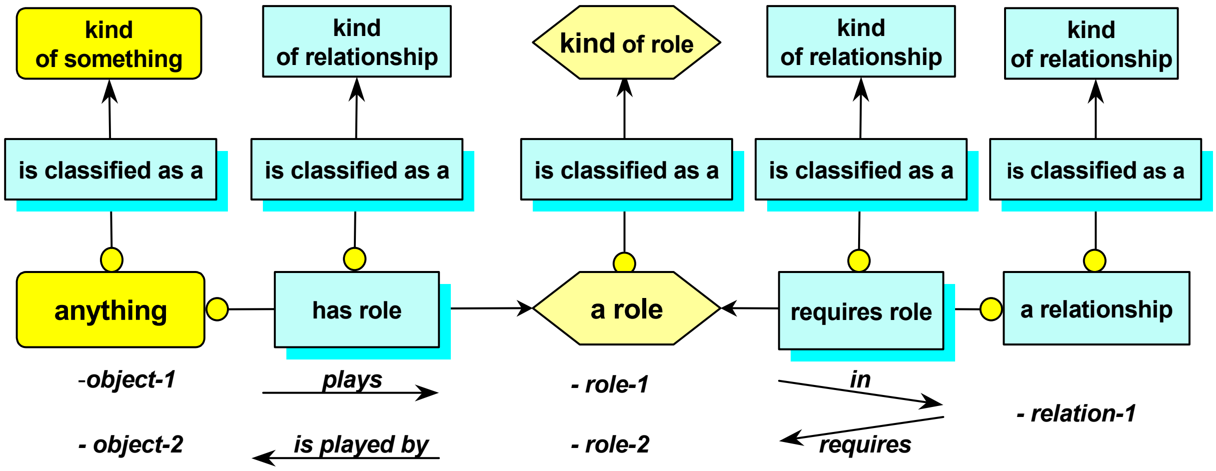


Figure 21, The basic semantic structure of the Gellish ontology

Note 1: The boxes with concept names that begin with 'kind of' represent respectively the concepts 'concept' or one of its subtypes, 'relation' (or 'is related to') or one of its subtypes and 'role' or one of its subtypes. This illustrates that 'kind of' usually stands for 'subtype of'.

Note 2: Remember that the arrows start with a circle and ends with an arrow point to indicate that the name of the relationship implies a direction for reading (in English the circle refers to the left hand side, the arrow point refers to right hand side of the natural language expression that expresses the whole fact). The arrow has no semantic meaning as every relationship can be inverted into an inverse expression that reads in the reverse direction and expresses the same fact.

Note 3: Remember that a box that is shaded contains a name that indicates the classification of the relation that is represented by the box.

The structure of Figure 21 comprises *basic concepts* including *basic relationships* of Gellish. The structure describes in generic terms *the expression of the definition of a fact*. That expression includes also the *interpretation context* for the expressed fact, because it includes the relationships to the concepts that indicate the kind or nature of the elements that form the expression of the fact. This means that a number of secondary facts are required to form the structure that expresses the prime fact.

The structure of Figure 21 forms the pattern that is necessary to define the other concepts and semantics of Gellish. Therefore, Figure 21 is some kind of 'bootstrap' model. It can be used as a 'template' for the expression of new facts.
The facts that can be expressed using this structure include individual facts about real or imaginary individual things as well as knowledge facts that form a structure of relationships between kinds of things. In other words it can be used to define 'anything', including individual things as well as classes and relationships.

Ternary relations or higher order **n-ary relations**, such as occurrences and correlations, can be expressed as a collection of binary relations. Each of those binary relations expresses that a related thing is involved in a particular role in the n-ary relation. This is explained in further detail for occurrences in section 3.3.6: Occurrences as dynamic facts. The representation of n-ary relations as a collection of binary relations enables to implement the Gellish language in a single table structure, as is explained in section 5.

Unary relations, also called single attribute relations, which are relations in which only one object is involved, can be represented in Gellish, although actually they are incomplete expressions of a fact. For example, it may be argued that the expression 'John walks' expresses a unary relation with only one object (John) involved. However, the relation is an abbreviated expression of a binary elementary relation, because the word 'walks' indicates that John is the performer of the 'walk'. If we make that role explicit, then, more completely expressed, we should say that 'John is performer of a particular walk', which shows that the expression actually expresses a fact that should be represented by a binary relation. Furthermore, the expression is an incomplete description of an n-ary act or dynamic relation, because 'walks' is an act or occurrence in which several things are involved. John is only one of them. Others are for example, the place of departure, the place of destination, the path along which he is walking, etc. This illustrates that it is questionable whether unary relations exist.

When the above structure is applied to **binary relations** between two individual things, this results in the picture of Figure 22, in which the structure of Figure 21 is applied twice to express both sides of the relationship.

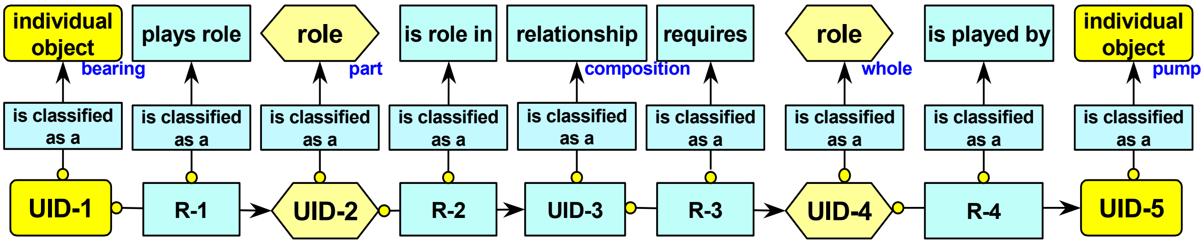


Figure 22, Basic semantic structure applied to express a (binary) individual fact

In Gellish, the concept relationship (of which a subtype classifies UID-3) is also called 'relation' or 'is related to'. The latter enables to express the basic fact presented in Figure 22 in a natural language as: UID-1 is related to UID-5. In this expression the roles UID-2 and UID-4, as well as the elementary relations R-1, R-2, R-3 and R-4 are left implicit, as is usually the case in natural languages and therefore also in Gellish.

The structure of Figure 22 expresses an individual fact between two individual objects, indicated as UID-1 and UID-5, which are related by the relationship UID-3. In other words it can be used to express that UID-1 is related to UID-5, or in full detail, explicitly mentioning the roles:

- UID-1 plays role UID-2 in UID-3, which requires UID-4 which is played by UID-5
  whereas each of the things that is identified by a UID is classified by a kind of thing as follows:

  - UID-1 is classified as an individual object
  - UID-2 is classified as a role
  - UID-3 is classified as a relationship (is related to)
  - UID-4 is classified as a role
  - UID-5 is classified as an individual object

Each of the things that is identified by a UID may also be classified by one of the subtypes of those kinds of things. Those subtypes are used to express more specialised facts. For example Figure 22 includes the example of a 'composition' relation (which is called in Gellish an 'is a part of' relation) that is a subtype of 'relationship'. This relation expresses the fact that a particular bearing is part of a particular pump. Such an expression uses the concept bearing which is a subtype of individual object, whereas part is a subtype of role, 'is part of' is a subtype of 'is related to', etc. The use of these subtypes enables to express a more specialised example as follows:

- UID-1 plays role UID-2 in UID-3, which requires UID-4 which is played by UID-5,
  whereas

  - UID-1 is classified as a bearing
  - UID-2 is classified as a part
  - UID-3 is classified as a composition (is part of)
  - UID-4 is classified as a whole
  - UID-5 is classified as a pump

All things identified by UID's or by R-1 through R-4 (which also stand for UID's) optionally have a name. For example, if UID-1 is called 'the front end bearing' and UID-5 is called 'P-4601', then a Gellish expression could be: The front end bearing of P-4601 (UID-1) is part of P-4601 (UID-5), which expression leaves the roles implicit.

The semantic definition of a kind of relationship determines the constraints for the facts that can be expressed by relations of that kind. For example, consider the above kind of relationship called 'composition'. The semantic definition of the generic concept of a composition relation states that it requires two roles: one classified as 'part' and the other classified as 'whole'. Later it will be explained how such a knowledge fact is expressed in Gellish. Then it will also be explained how the knowledge fact is expressed which states that such roles can be played by instances of 'individual object' or one of its subtypes. Here it is only noted that the example in Figure 22 mentions those roles and allowed role players, without defining those constraints explicitly.

So, Figure 22 illustrates a relation between individual things. Scientific knowledge is generally expressed as relationships between kinds of things. This can be done in Gellish by the application of the structure of Figure 21 to a relationship between two *kinds* of things, which illustrates that the structure can also be used to express knowledge.

This is illustrated in Figure 23.

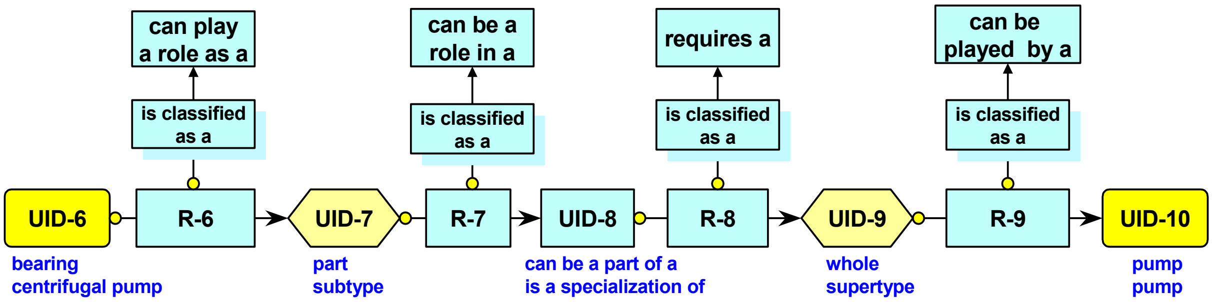


Figure 23, Basic semantic structure applied to express a conceptual fact (knowledge)

The general structure of Figure 23 can be used to express explicit knowledge by replacing the kinds of things by a consistent set of their subtypes. Two more specialised examples are given in the figure:

1. The knowledge fact: a bearing can be a part of a pump.
2. The knowledge fact: centrifugal pump is a specialization of pump.

It is on purpose that UID-6 through UID-10 are not explicitly classified as being a kind of thing. This is left out as being superfluous, because they inherit from their supertypes that they are defined as being kinds of things. Because of the inheritance mechanism, only the top class in the hierarchy of kinds of (individual) things (being the concept 'individual thing') need to be classified as a 'kind of thing' or 'class').

The concepts in Figure 21 are called basic concepts or axioms, because the structure as a whole is required for the definition of further concepts and because less concepts provide insufficient semantics for the expression of definitions of more fundamental concepts. Once the structure is available and the meaning of these basic concepts is known, then further concepts can be defined using the structure.

The basic concepts are:

- anything (something)
- role
- fact
- relationship
    - role of something ('something plays role' or the inverse: 'role is played by something')
    - role in relationship ('relation requires role' or the inverse: 'role is required by relation')
    - is classified as a (kind of something)
        - is classified as a kind of role
        - is classified as a kind of relationship
- kind (or concept)
    - kind of something
    - kind of role
    - kind of relationship

To understand also the meaning of the structure it is required to understand that the structure includes *expressions of elementary facts*. Each elementary fact is expressed by an elementary relationship.

The elementary relationships are defined as follows:

- An *elementary relation* is either a 'requires role' relation between a role and its requirer (a relation) or a 'plays role' relation between a role and its player.
- A *plays role* relation is a relationship between *anything* and *role* or between a pair of their subtypes, which indicates that *something* plays or has the *role*.

- A *requires role* relation is a relationship between *relation* and *role*, which indicates that the *relation* implies the *role*.
- An *is classified as a* relation is a relationship between *anything* and *concept*, or between a pair of their subtypes, which indicates that *anything* is classified by *concept or by one of its subtypes*. The latter relationship is usually called the 'classification' relation.

The latest requirement indicates that the concept '*subtype*' is also something that should be known as one of the basic axioms. On one hand this seems unnecessary, because it is possible to define the specialization/generalization relationship (called '*is a specialization of*' or '*is a subtype of*') using the above structure. The definition of that specialization relation includes that it requires two roles: a *subtype* and a *supertype* role. Each of those roles is defined to be played by a different *kind*. On the other hand it appears that this definition itself makes use of the idea of specialization, because the specialization relation itself is already a specialization of the concept '*relation*'.

This suggests that the specialization/generalization relation is not a derived relation, but a basic one. If we thus include the specialization relation in the structure of basic concepts, then we might eliminate the subtypes of '*kind*'. This results in a basic structure as presented in Figure 24, where all three classification relationships point to the same 'concept' or 'kind', whereas it is meant that each classification relation actually points to a different subtype of concept.

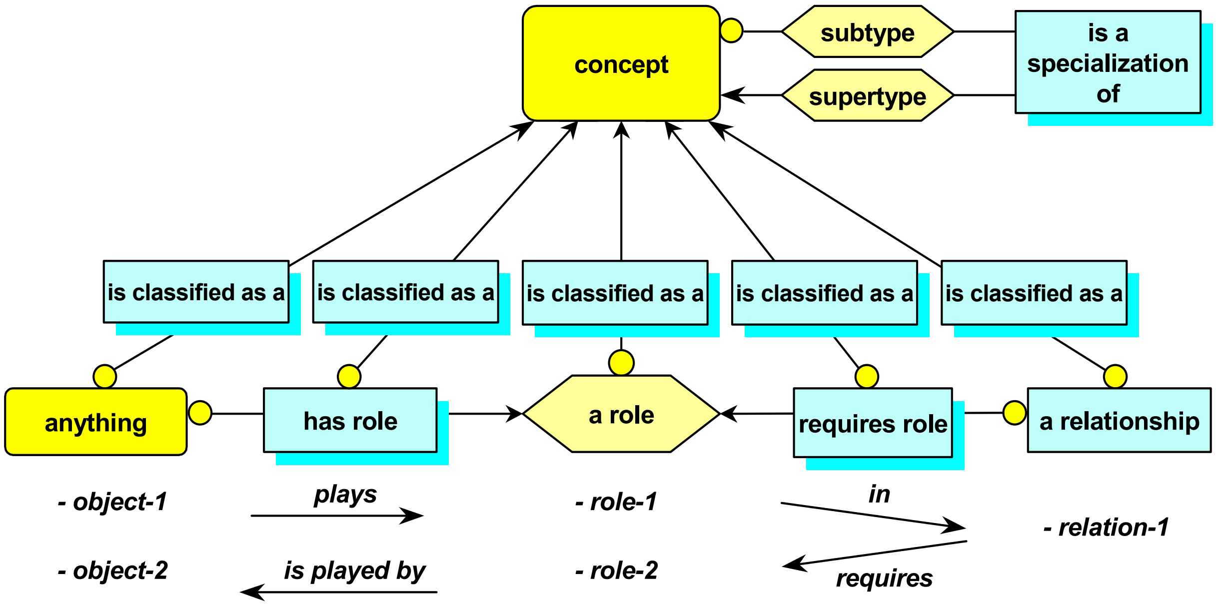


Figure 24, Alternative basic semantic structure of the Gellish ontology

To interpret the above basic semantic structure correctly, it is required that subtypes of 'concept', such as '*role*' and '*relation*' are known, because we need to have those subtypes available in order to make the classification of the roles and the relationship explicit. This alternative basic structure therefore implies that the specialization relation is a basic concept.

This means that this alternative requires that we have to extend the structure of basic semantic concepts with the specialization/generalization relationship and with the subtypes of role as follows:

- *is a specialization of* (also called *is a subtype of* or their inverses) is a relation between two different concepts.

This relation requires two roles:

- *subtype*, which is a role of a concept of being more constrained than the supertype concept.
- *supertype*, which is a role of a concept of being less constrained than the subtype concept.

This addition implies that in this alternative basic structure the concepts *role* and *relation* become derived concepts and are not true basic concepts.
The structure of Figure 21 and Figure 24 are practically equivalent, because we can apply Figure 21 to

define the specialization relation of Figure 24, and on the other hand, we can apply Figure 24 to define the subtypes of 'concept' of Figure 21. In both cases the specialization relation of Figure 24 needs to be applied anyway to define the subtypes of role and relation that are necessary to define other semantics of the Gellish language.

The above story clarifies why facts, expressed as a network with a fixed pattern of relationships between things ('anything'), are the founding structure of the Gellish language.
These patterns and networks are natural language independent, because the things in such a network are independent of the names for those things, although English names of the things are often filled-in in the boxes in the figures. Furthermore, the fixed pattern defines a grammatical structure which makes the structure a 'language' with as rule for the expression of facts that a fact shall be expressed using the pattern of Figure 21.

In other words, each atomic fact shall be represented as an 'instance' of the basic pattern or structure to express a fact in a way as given in Figure 21. Furthermore complex facts can be expressed as a coherent collection of atomic facts. This is the background why it is possible to define a single 'Gellish fact' entity that defines a Gellish Table for a single table database implementation of Gellish as is described in section 1.

An example of a collection of elementary facts that together express an atomic fact about an individual thing is the following: The atomic fact that "my engine is part of my car" can be expressed in Gellish as a structure composed of four expressions that express four *elementary facts*, arranged in two pairs, each of which pair is an instance of the lowest line in Figure 21:

- object-1	has role	role-1
- role-1	is a role in	relationship-1

and

- object-2	has role	role-2
- role-2	is a role in	relationship-1

These four elementary expressions can be interpreted correctly when it is clear which things are meant with the objects, roles and relationships. There are two methods to provide those interpretation rules, being the rules for the understanding of the expression of the atomic fact. One method is to point to the objects and their roles towards each other in reality, which sign-language is sufficient for an understanding of the expression. Another method is to add a number of other expressions of atomic facts. In other words, by adding verbal explication. This can be done in Gellish by provision of the following collection of classification relations, which in themselves express atomic facts:

- object-1	is classified as a	engine
- role-1	is classified as a	part
- relationship-1	is classified as a	'is a part of' relation (a composition relationship)
- object-2	is classified as a	whole
- role-2	is classified as a	car

If we combine all the atomic expressions graphically in one structure, then we can draw a picture such as in Figure 25.

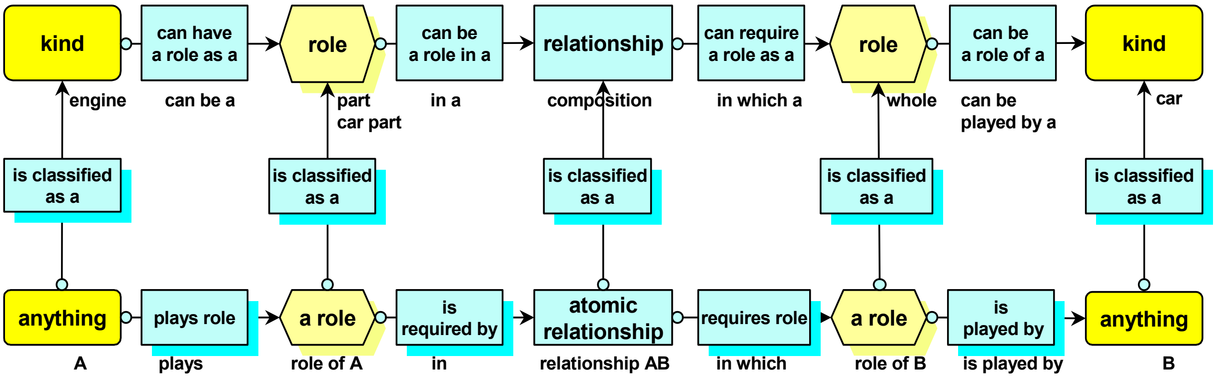


Figure 25, Expression and interpretation of the composition of a physical object

The lower line in Figure 25 expresses a sentence that can be read from the figure as follows:

- A plays 'role of A' in relationship AB in which 'role of B' is played by B.

The classification relations with the concepts at the upper line provides the interpretation rules, so that we can understand that:

- engine A plays a role as part in a composition relation in which the role of whole is played by car B.

In this way it is possible to describe reality and imagination in Gellish as things, represented as 'instances' of 'anything' and facts about things, represented as instances of atomic relationships between things.

As an illustration of the expression of knowledge, the relationships between kinds of things are added on the upper line in Figure 25. Those relationships express which kinds of roles can be played by things of a kind and which kinds of roles are required by relationships of a kind. An example of an instance of this structure is the sentence that can also be read from the figure as follows:

- an engine can be a part in a composition in which (the role of) whole can be played by a car.[22]

In summary: *Figure 21, The basic semantic structure of the Gellish ontology, presents an expression of the structure of an atomic fact and its interpretation context (its definitions). This expression includes representers of things and relations between things. Every atomic fact can be expressed by an atomic relationship which associates two things via two instances of 'role of something in relationship' and two instances of anything which participate in that single relationship in those roles.*

Therefore, the expression:

- my engine E   is part of   my car C

is an expression of an atomic fact and is an abbreviated summary of the above collection of elementary facts. But also the expression:

- an engine   can be a part of a   car

is an expression of an atomic fact that is an abbreviation of a similar collection of elementary facts. The latter expression expresses knowledge about what normally can be the case.

[22] Usually, the subtypes of role and relationship are not specialised as detailed as the subtypes of other things. This seems to be caused by the linguistic phenomenon that a high level kind of role and relation is sufficient for an unambiguous interpretation of expressions. For example, engine and car are deeper subtypes than part and composition. However, in some contexts such deeper subtypes do appear. For example, the concept 'car component' could be used instead of just 'part', but for interpretation that would be superfluous.

2.4.3 Definition of additional semantic concepts

Once the basic semantic concepts and their structure are known, it is possible to use that structure and those concepts for the definition of new concepts. For example, we can generalise the latter expression up to the level of a definition that an

- individual object can be a part of an (other) individual object.

This generalization implicitly defines the knowledge of what a 'composition of individual object' means and thus defines the concept 'can be a part of a'.
This illustrates that the structure of Figure 21 can also be used to define new relation types, which are semantic concepts that on their turn can be used to express other kinds of facts. In this way new semantic concepts can be added to the Gellish language by providing the semantic structure of the definition of such concepts. This makes Gellish extensible.

The process to define a new relation type is as follows:

1. Define a subtype of relation ('is related to') by the specification of a specialization relation between the subtype and 'relation' or an *existing* (logically earlier defined) subtype of 'relation'.

   For example:
   - 'is involved in' is a specialization of 'is related to'.
2. Define which kind of left hand role and right hand role are required by the subtype of relation.
   For example:
   - 'is involved in' requires a first role as an 'involved'
   which role is played by the object on the left hand side in a language that reads from left to right, and
   - 'is involved in' requires a second role as an 'involver'
   which role is played by the object on the left hand side in a language that reads from left to right.
3. Define those two kinds of roles via specialization relationships with the existing supertypes of those kinds of roles.
   For example:
   - 'involved' is a specialization of 'related'
   and
   - 'involver' is a specialization of 'relator'.
4. Define which kind of thing can have instances that can play a role of the kind of left hand role and kind of right hand role respectively.
   For example:
   - 'physical object' can play a role as a 'related'
   and
   - 'occurrence' can play a role as a 'relator'.

The definition of such a new semantic concept is illustrated in the following Gellish Table, using the above example.

UID of Fact	**Language context**	**UID of thing**	**Name of something at left hand side**	**UID of relation type**	**Relation type name**	**UID of thing**	**Name of something at right hand side**
1	English	4767	is involved in	1146	is a specialization of	4658	is related to
2a	English	4767	is involved in	4731	requires a first role as a	640118	involved
2b	English	4767	is involved in	4733	requires a second role as a	4773	involver
3a	English	640118	involved	1146	is a specialization of	4824	related
3b	English	4773	involver	1146	is a specialization of	4729	relator
4a	English	730044	physical object	4714	has a member with a role as a	640118	involved
4b	English	192806	occurrence	4714	has a member with a role as a	4773	involver

Note: In the above table the numbers of the fact UID's refer to the steps above. The other UID's are selected from the Gellish dictionary.

In addition to the above it needs to be defined per relation type how many relations of the same kind are allowed with one thing of a kind at the same time. These 'cardinality constraints' are indicated in Gellish by numbers that indicate the minimum and maximum left hand and right hand simultaneous occurrences that are allowed. Each of those numbers indicates a minimum or maximum number of different roles of the kind that is specific for the relation type that may simultaneously exist in relationships of the kind with one thing that is of the 'relator' kind. Therefore, the numbers are related to the roles as being the 'number of simultaneous allowed roles of the kind'. The cardinality concept is discussed in more detail in section 3.3.9.3.

Finally, for the definition of a new relation type concept it is required to define whether a thing can or may have a relation of a certain kind with itself. Relation types for which this is not allowed are defined to be a subtype of a kind of relationship between things that has the constraint that the related (left hand) thing can or may not be the same thing as the other (right hand) thing. Via the subtype/supertype hierarchy for kinds of relationships this constraint is inherited by all more specialized kinds relationships that have that constraint. Relationships between concepts that express that members of concepts (classes) can have a certain kind of relationship often don't have such a constraint, because it is often possible that different instances of the same kind have a relation of that kind. For example, the recursive expression that 'an activity can be a successor of an activity' is true and is allowed in Gellish, because it expresses that (different) members of the concept 'activity' can be related in such a way. However, the expression that 'activity A is a successor of activity A' is incorrect, because this expresses a relationship between individual things, while an individual activity cannot be a successor of itself. Therefore the definition of the succession relation specifies (or inherits) that the two related members shall be different.
This constraint is also used to express the rule in Gellish that a whole is by definition more than each of its parts and thus that a whole cannot coincide with one of its parts. This rule prevents the paradox of Russell.[23]

[23] The Paradox of Russell is a criticism to set theory that states that the set theory concepts allow a contradiction, because of the definition of set. That definition allows for a set S with the following definition: Assume that a set S is defined as the set of all elements that are not members of themselves. The answer to the question whether S is a member of itself leads to a contradiction as follows. If S is not an element of S, then it can be argued that S should be an element of S, (continued on next page)

In a similar way we can define a conceptual relationship that expresses that members of a kind can be composed to members of another (or the same) kind. The following is an example that illustrates how a 'conceptual composition relationship between individual objects' is defined in Gellish, in a way compliant with the structure of Figure 21.

1	can be a component of a	is a specialization of	can be related to a
2	can be a component of a	requires as first role a	conceptual component
3	can be a component of a	requires as second role a	conceptual whole
4	conceptual component	is a specialization of	conceptually related
5	conceptual whole	is a specialization of	conceptual relator
6	individual object	can have a role as a	conceptual component
7	individual object	can have a role as a	conceptual whole

It is self evident that it is required that the concepts such as 'individual object' and the relation between classes called 'can be related to a' have to be (and are) defined before these expressions can be interpreted. In that sense there is a consistent hierarchy of definitions that ends in one top: 'anything'.

As mentioned earlier each concept and each relationship in Gellish has its own unique identifier (UID). In the above example the concepts are indicated by names, assuming that the definition of the concepts is sufficiently clear from their names.

Further definitions of all the semantic concepts of kinds of relationships in Gellish are defined in the subset 'upper ontological facts' of the Gellish language definition, in the upper ontology part (TOPini) of the Gellish Dictionary / Taxonomy. The current collection of kinds of relationships is presented in Appendix B, Upper ontology of relations with their roles and role players. An further example of the

---

(continued from previous page) but it can be defended that the opposite is also true, which is a paradox (see "Knowledge Representation" by John F. Sowa, page 102 and 103). This paradox is avoided in mereological theory (a theory of "parts and wholes") developed by *Lesniewski*, which states that *by definition everything is part of itself*. Then the set S defined by Russell does not exist, because there are no sets that are not members of themselves. This solves the paradox.
I have not adopted this solution in Gellish, because of the following reason.
In the theory of Lesniewski, a part is by definition a part of itself. If such a 'fact' would be expressed by an explicit relation in Gellish, then it would be *a composition relation between an object and itself*. It is questionable what the meaning would be of such an expression. In my opinion such a proposition would not express any knowledge. Declaring it 'by definition' does not make it 'the case'. Furthermore, it could be argued that the part object has again a composition relation with itself. This would result in an endless list of composition relations between an object and itself. This problem is caused by the fact that the part-whole relation is defined from a mathematical perspective, whereas in mathematics the parts and wholes are classes (numbers appear to be classes: the number 1 exists only once according to mathematical theory). In a physical world, where a part-whole relation is primarily a relation between individual objects, it is more feasible to define a part-whole relation as a relation between two objects, *where the whole object is by definition more than each of its parts*. This adds a constraint to the composition relation that states that the part object may not be the same as the whole object. So, *an object cannot be part of itself*.
This leads to the issue that the collection of all collections does not exist, because if such a collection would exist, there is at least one collection missing, which is the collection itself. However, this seems reasonable, because it can be argued that such a collection would have an infinite number of elements and thus it shares its problem of existence with the mathematical concept of infinity. This questionable existence of 'the collection of all collections' is not a problem for the definition of a collection in Gellish, because in Gellish the class 'collection' is not defined using set theory, but it is defined as the criterion that the plurality is not by definition 1.
So, in *the Gellish ontology parts (and collections) are by definition not part of themselves*. This avoids the Paradox of Russell.

definition of a kind of relationship is presented in Appendix C, Upper ontology concept definition example.

2.4.4 Individual things and Kinds of things (concepts)

This paragraph discusses the differences between relationships between individual things and relationships between kinds of things. This distinction forms the basis for the statement that a generic language such as Gellish shall have different semantics (different kinds of relationships) for the description of individual things and for the description of knowledge as a description of the commonality between instances of kinds of things.
This distinction also forms the background behind the statement that Gellish as a language is suitable to describe individual things for which not yet a model 'on class level' is created. This is opposed to conventional data models in information technology, which are models of relations between classes and where every information about an individual object can only be expressed after a data model for such a kind of thing is created on class level. In other words: where the knowledge about such a kind of thing is expressed as relations between classes.

It is generally accepted that there is a fundamental distinction between an 'individual thing' and a 'kind' ('kind of thing') or 'class' (note that the term 'class', 'as single' is used here as a synonym of 'kind'). We experience in the real world only individual things, but we recognise 'commonalities' between individual things. Based on that we categorise individual things in 'kinds' or we talk about the common nature of things or about concepts. These kinds or concepts only exist in our minds.
Therefore we can define a '*kind*' as '*a criterion for the determination of commonality between things or for the creation of things that comply with the criterion*'.

Often those criteria are not made explicit and when that is tried, it appears that it can be very difficult. It is even defended that for a number of concepts it is fundamentally impossible to give a proper definition. In other words, for some concepts it seems to be impossible to make the criteria explicit without becoming circular by referring to the concept itself. In those cases the criterion is 'native' and the concept seems to be an 'innate concept' (see Wierzbicka 1996, p 9, 12, 16).

The distinction between individual things and kinds or concepts is illustrated in Figure 26.

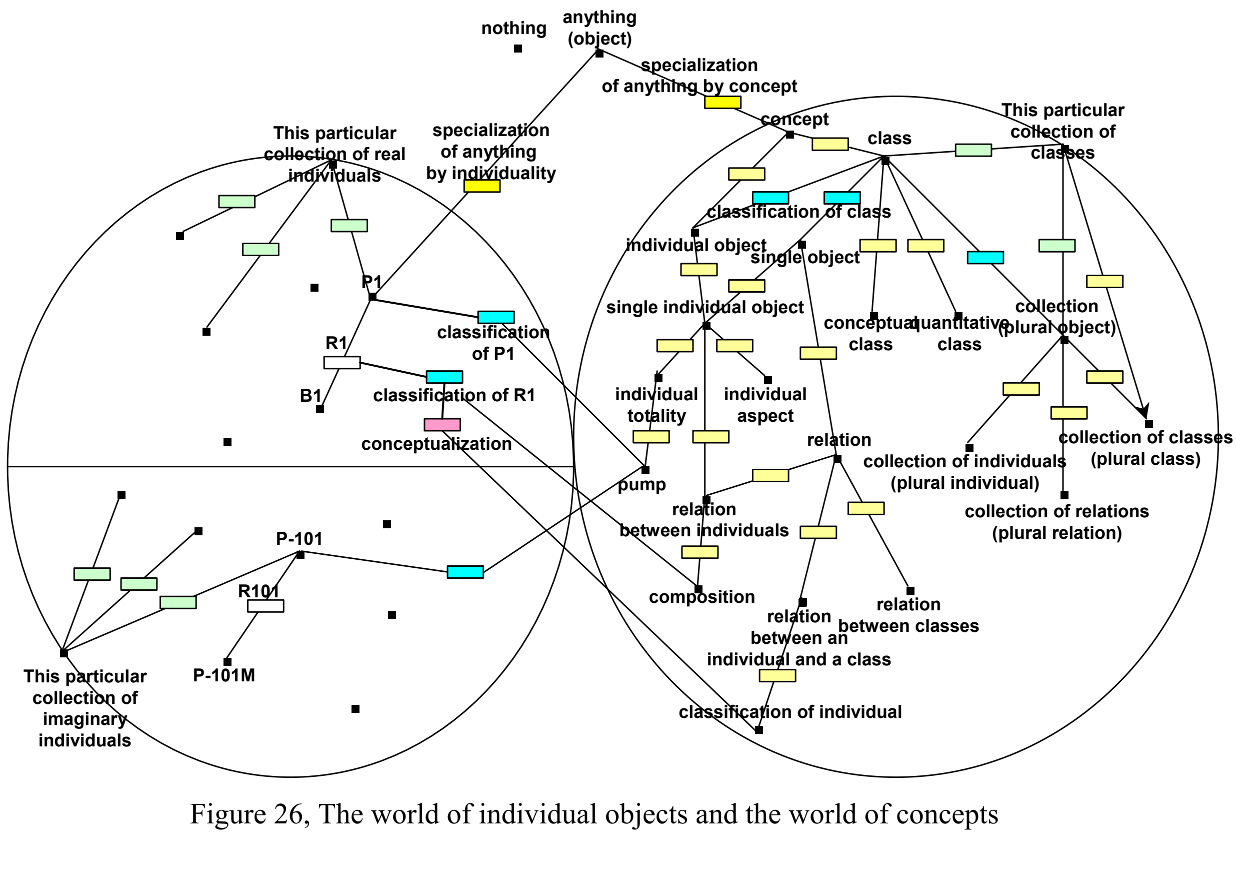


Figure 26, The world of individual objects and the world of concepts

In Figure 26 things are subdivided in two ‘worlds’: the world of the individual things (left hand) and the world of concepts (right hand), whereas the individual things are further subdivided in real individual things (left hand upper part) and imaginary individual things (left hand lower part).
In most scientific models, as well as in data models of information technology, the individual things are missing. Exceptions are descriptive sciences, such as history and geology and similar sciences that deal with a description of individual things. Also in the application of technology and in business practice the descriptions of individual things often occurs. For example, a design can be seen as an imaginary individual thing of which many (imaginary or real) copies can be derived[24]. Also individual maintenance activities are planned and executed as actions on real individual things.
In natural sciences one tries to identify commonality of aspects and behavior among individual things. A definition of the commonality among individual things forms the definition of the kind. As a next step in modeling in natural science one formulates laws as correlations between kinds of aspects. Those laws describe the commonality of relations between aspects, which aspects are members of the correlated kinds. Individual aspects of individual things behave according to individual laws, but if it appears that there is consistency between the classification of things and the classifications of their aspects, then there is also commonality of the behavioral laws. The results of this scientific practice is documented as kinds and relationships between kinds of things. These kinds of relationships express knowledge about instances of those kinds[25].
So, general knowledge is documented as relationships between kinds of things, but that knowledge is indirectly knowledge about aspects of individual things that are members of those kinds.

A complete ontology should in my opinion cover both worlds: the world of individual things and their aspects as well as the world of concepts (kinds) and generally valid aspects that are valid for individual things that have a commonality regarding those aspects and are therefore members of those kinds.
Such a complete ontology therefore includes the structure of individual things as well as the structure of generally applicable knowledge about commonality between individual objects.

These two worlds require different relation types, because the semantics of a relation that expresses an individual fact is different from the semantics that expresses knowledge about possible relations or requirements for relations between members of classes.

---

[24] For a detailed discussion about designs see section 3.5.1.

[25] The phrases ‘kind of’ or ‘class of’ or ‘type of’ are in most contexts synonyms of ‘subtype of’. For example, the phrase ‘the class of physical object C’ (e.g. ‘pump’) expresses that a thing C is a concept (or class or kind) that is a subtype of ‘physical object’ that can be used to classify an individual thing P as being a physical object or a subtype of it. The collection of classes of physical objects is therefore identical to the collection of subtypes of the class ‘physical object’. This implies that it would be superfluous to define all kinds of collections of classes, as such collections can easily be determined by collecting all the subtypes of the appropriate concept.

This is illustrated in Figure 27.

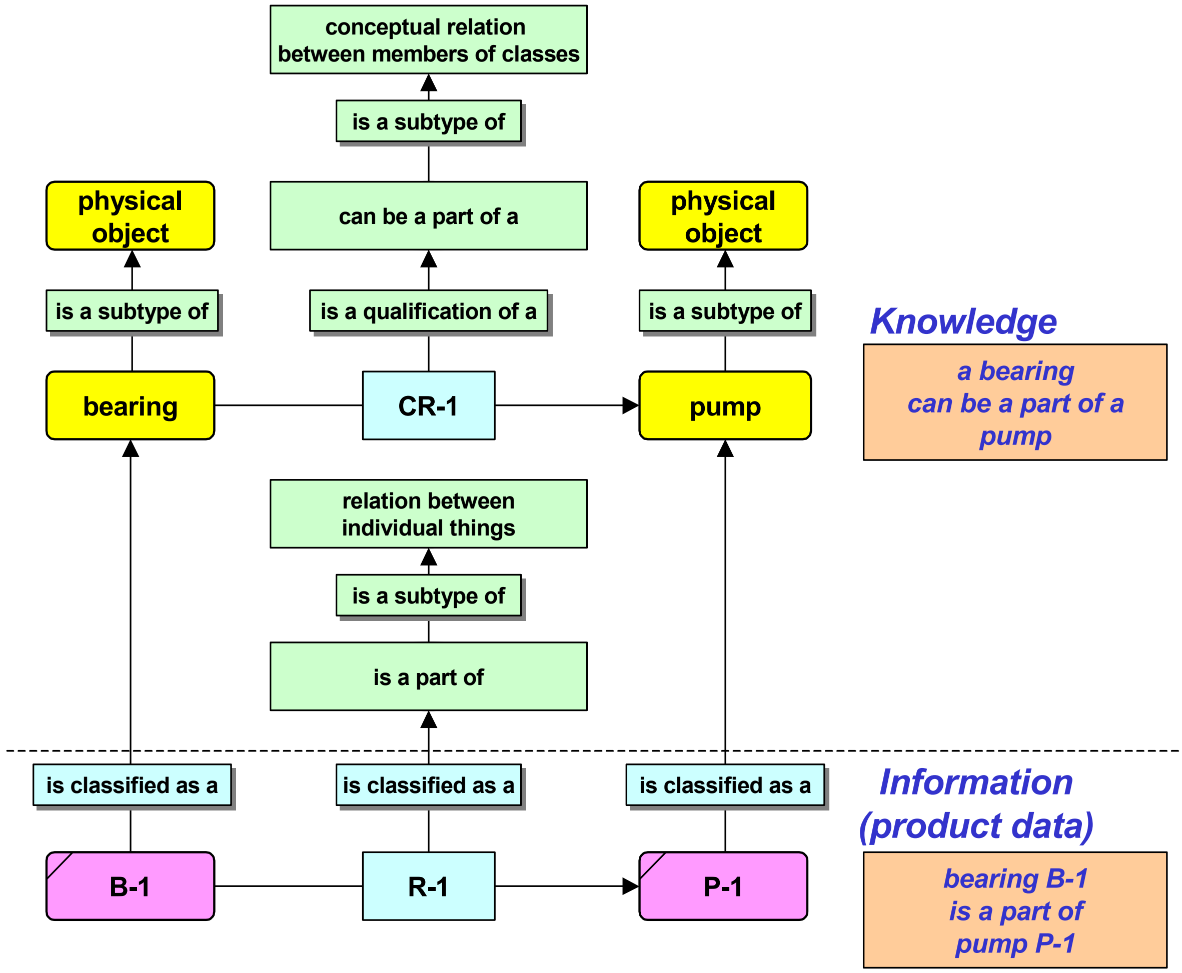


Figure 27, Example of expression of an individual fact and a knowledge fact

Figure 27 illustrates first that every individual thing, such as B-1, R-1 and P-1, can be classified by a kind of thing or class. A fact is of a certain kind, a role is of a certain kind, a relationship is of a certain kind and each other object is of a certain kind. Often they can even be classified by various kinds. However, that classification can be on various levels of abstraction. We can classify a particular individual thing R as a composition relationship as follows:

- R is classified as a 'is a part of'.

But we can also classify the same thing R as a realization of a knowledge relationship CR, where the latter expresses the fact that a wheel can be part of a car. This can be done as follows:

- R is classified as a CR

in which CR is a relationship between classes, whereas that kind of relation is described by the string:

- a wheel can be part of a car.

The second classification relation indicates that relationship R (an 'is a part of' relation) is in accordance with the knowledge CR. However the fact R is also a valid expression of a fact if the knowledge CR is not known or not modeled yet.

Figure 27 further illustrates that facts such as R-1 can be expressed and interpreted correctly without the need to have fact CR-1 being defined. It is a ‘knowledge acquisition’ act to generalize R-1 (and similar R-2, R-3, etc.) to a generic conclusion that CR-1 is true[26].

The relationships between the kinds are of a different nature than the relationships between the individual things. The nature of such relationships is that they are themselves kinds of relations that can primarily be used to classify individual relationships when they exist. Secondly, these kinds of relationships can be used to verify whether a statement about a relationship is a valid one in view of the knowledge expressed by the kind of relationship. Thirdly, during a design it is possible to use these kinds of relationships as a template or pick list of known options (knowledge) from which design alternatives can be selected. This will be discussed in more detail in section 6.3.

For each kind of relationship (relationship between classes) it is indicated in Gellish which kinds of roles are required and for each of those kinds of roles it is indicated which kinds of things can play roles of those kinds. Those kinds of relationships describe the recognized fact types as expressions of the possibilities and impossibilities or constraints on the individual relationships and roles of those kinds.

The knowledge about *kinds* of facts that classify facts that occur in reality, can be expressed as relationships between kinds of things. This is illustrated on the top line of Figure 25. For example, the fact that ‘a steering wheel can be a part of a car’ can be represented by fact S1 (a conceptual composition). Such a fact can be expressed in Gellish as a collection of relations between kinds of things. For the example fact, this results in the following kinds of relations:

- a	steering wheel	can have a role as a	part (a car part)
- which	part	can be a first role in a	‘can be a part of a’ relation

whereas

- a	car	can have a role as a	whole (a car assembly)
- which	whole	can be a second role in that	‘can be a part of a’ relation

The relationships between kinds of things are themselves also facts. They can express generally valid knowledge or they can express requirements that are valid only in a particular context. These kinds of relationships are to be distinguished from relationships that relate to individual things. Thus there are two groups of facts:

1. Facts about individual things
2. Facts about kinds of things.

The latter are facts that are valid (in general or within a particular context) for individual things that are members of the particular related kinds of things. Both, the individual facts as well as the kinds of facts, are subtypes of the generic more abstract concepts, as will be discussed later in more detail.

Example of a definition of a new kind of fact

As stated above, the definition of new kinds of facts can be expressed as a collection of expressions conform the pattern of Figure 21. This can be illustrated with the following example: Assume that we want to define a new concept, being a kind of relationship called ‘composition of physical object’.

---

[26] In fact the relationships with a name that start with ‘can be’ are not primarily intended to indicate a possibility, but are a generalized conclusion about what normally is the case. It indicates the knowledge that it is conceptually the case within the cardinality constraints. These cardinality constraints also indicate whether a relationship is an optional or obligatory relationship. This optionality also covers the difference between definitional relationships and optional relationships. A definitional relationship is obligatory because it forms part of the definition of the concept. An optional relationship may exist or not, without becoming in conflict with the definition of the concept. For example, a pump might be defined to require (‘shall have’) at least one bearing as an obligatory relationship, but it may or may not have (‘can have’) an impeller.

This can be expressed as a generalization of the example from Figure 25 as follows:

A fact can be classified as a ‘composition of physical object’ if and only if:

- object-1	is classified as a	physical object (or one of its subtypes),
- role-1	is classified as a	part (or one of its subtypes),
- role-2	is classified as a	whole (or one of its subtypes),
- object-2	is classified as a	physical object (or one of its subtypes)

whereas

- composition of physical object   is a specialization of   relation between individuals.

Via the specialization relation the new relation inherits the condition that:

- object-1   is not the same as   object-2.

A more detailed description of how facts are represented and expressed in Gellish is discussed in section 3.3 Facts and their dynamics – Ordinary facts and Occurrences.

2.4.5 Specialization hierarchy

A large proportion of an ontology can be represented as a taxonomy or subtype/supertype hierarchy[27]. This paragraph describes how such a partial ontology is represented as a specialization hierarchy in Gellish. It also defines the specialization relation between kinds of things and the related constraints.

A specialization hierarchy is defined as a coherent collection of relationships between kinds of things that can be called specialization relationships or generalization relationships depending on the direction from which one starts. When starting from the subtype we read: ‘is a specialization of’, when starting from the supertype we read: ‘is a generalization of’. Both names refer to the same relationship type, but read in an inverse direction. In the specialization hierarchy of Gellish each kind of thing has at least one specialization relationship with its direct supertype, except the top concept, called ‘anything’. The result of this requirement is that the whole hierarchy of Gellish concepts forms one coherent hierarchy of which ‘anything’ is the top object. This includes not only physical totalities, but also occurrences, aspects, relations, roles, etc.

A specialization relationship is defined in Gellish using again the basic semantic structure of Figure 21 as follows:

- specialization relationship   is a specialization of   relation between kinds of things

or with the Gellish name:

- ‘is a specialization of’   is a specialization of   relation between kinds of things

provided that the following constraints are satisfied:

- object-1	is a qualification of	concept (kind of thing)
- role-1	is a qualification of	subtype
- role-2	is a qualification of	supertype
- object-2	is a qualification of	concept (kind of thing)

whereas it is again inherited that object-1 is not identical to object-2.

[27] There is a lot of confusion about the homonyms of the English expression ‘is a’, caused by impreciseness of the use of natural languages. These homonyms are about different subtypes of ‘is related to’. The phrase ‘is a’ is used as a name for at least two concepts:
1) A classification relation, for example in the expression ‘P1 is a pump’ or in the expression ‘pump is a class’.
2) A specialization (subtype) relation, for example in the expression ‘a line shaft pump is a centrifugal pump’. Gellish makes a further distinction between classification of individual and classification of class relations.

The definition of the concepts ‘subtype’ and ‘supertype’ form a pair of roles that are defined in relation to each other, because the following rule holds:

- the kind of thing that has a role as subtype in a specialization relation is defined by criteria for its members to be a member of the subtype concept that are a superset of the criteria that define the supertype concept in such a specialization relation.

In other words, the subtype is distinguished from the supertype because the subtype members have to satisfy one or more additional constraints on the degrees of freedom for its aspects. Therefore we call those additional constraints the qualitative aspects (values) of the *discriminating aspects*.

So, a discriminating aspect is an aspect of a supertype concept which values are the discriminators between the subtype concepts (see also section 2.4.7). This is illustrated in Figure 28.

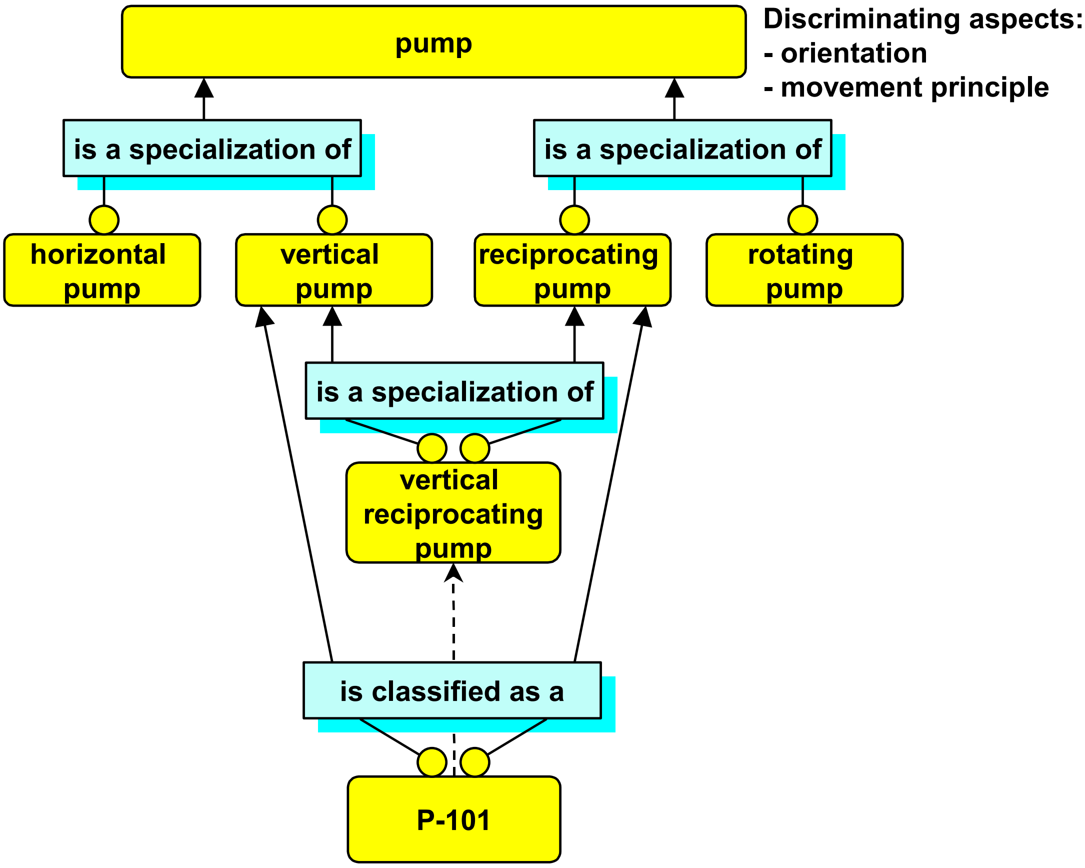


Figure 28, Non-mutually exclusive subtypes and multiple supertypes or multiple classifications

In the example of Figure 28 the subtypes of the kind of thing called ‘pump’ can have as discriminating aspect their orientation (horizontal and vertical shaft) or their movement principle (rotating or reciprocating). This results in four subtype kinds of things: horizontal pump, vertical pump, rotating pump and reciprocating pump. The first two are mutually exclusive and the second two are also mutually exclusive, but a kind from the first pair and a kind of the second pair are not mutually exclusive. However, a particular individual pump, say P-101, can be for example a vertical pump and also a reciprocating pump. This makes that the individual pump is double classified.
Furthermore, there can be further sub-subtypes of pump, such as horizontal rotating pump, vertical rotating pump, horizontal reciprocating pump and vertical reciprocating pump. Each of these subtypes has two supertypes (a ‘child’ with two ‘parents’). A Gellish specialization hierarchy allows for such multiple supertypes for a subtype, as well as for multiple classification of an individual object. When only one discriminating aspect would be allowed it is forced that one has to define a preference for the discriminating aspect that should come first as criterion in building the specialization hierarchy. The facts that Gellish allows for multiple supertypes of a subtype solves a lot of debate about such a usually arbitrary preference, but about which people often have strong opinions. This can illustrated by the fact that for example the American Petroleum Institute (API), which is an authority on pump

type specialization, has ‘standardized’ a specialization hierarchy that has a built-in preference of discriminating aspects, so that certain types of pump are not allowed anymore, although they are widely used in practice.

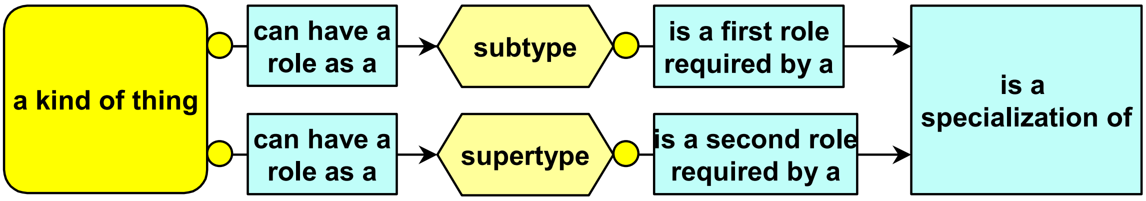
The example of Figure 28 also illustrates the general case that one subtype distinguishes itself from the other subtypes of the same supertype by differences between the values of the discriminating aspects. Often the discriminating aspects of two or more subtypes are different qualifications of the same conceptual aspect. In that case, these differences normally are mutually exclusive, which makes that the subtype kinds of things belong to a collection of kinds of things that have mutually exclusive definitions. In such a case a member of a subtype is not also a member of another kind of thing in the same collection of subtypes. However, sometimes a supertype is also specialized on the basis of several qualifications of another conceptual aspect. This results in another collection of subtypes that are mutually exclusive. Then a member of a subtype of the first collection can at the same time be a member of a subtype of the second collection. But it is also possible that a further sub-subtype exists that is a subtype of a kind of thing in the first collection as well as a subtype of kind of thing in the second collection.

So, a subtype is distinguished from other subtypes of the same supertype by differences in the constraints on the degrees of freedom for their aspects. This means that a subtype has the same constraints as the union of the constraints of its supertypes, apart from the additional constraints defined for the subtype. In other words, a subtype has all aspects that a supertype also has, but the subtype has more constraints on its aspects. In information technology this principle is called ‘inheritance’ (see also the 'genus/species' relations of Aristotle). It expresses that the subtype kind inherits all aspects from its supertype(s).
Two remarks should be made here:

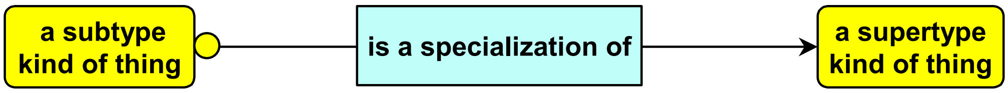
1. The term ‘inheritance’ is inaccurate, as the subtype kind does not inherit anything from the supertype kind, but the constraints on the supertype apply also to the subtype.
2. The ‘inheritance’ follows from a relation between concepts, not between members of the concepts. This means that members of the kind (or ‘instances’) have aspects that shall *satisfy the constraints of* the kind, but those aspects of the members are not identical to the (conceptual) aspects of the kind. The individual things do not inherit from the classifying concepts, but shall satisfy the constraints of the concepts (classes, kinds).

The above description demonstrates that a complete specialization/generalization hierarchy can be expressed as sets of instances of the entities in Figure 21. It also illustrates that this hierarchy can be instantiated together with all other expressions of facts in the same database with only the data model of Figure 21.

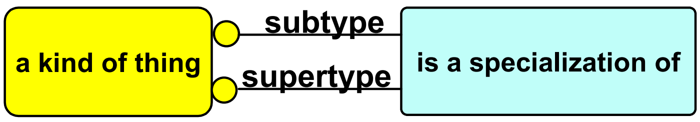
The graphical representation of a specialization/generalization hierarchy can be done on various levels of detail. This is illustrated in Figure 29.



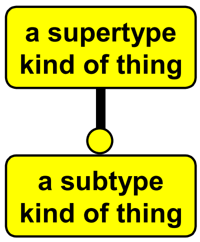
a) Representation used for the definition of the relationship concept in Gellish



b) Representation used to represent a specialization relation in Gellish



c) Formal EXPRESS-G representation according to ISO 10303-11



d) Alternative EXPRESS-G representation used to represent a specialization relation

Figure 29, Equivalent graphical representations of a specialization hierarchy

A detailed representation (the top one) shows the subtype and supertype roles as separate objects (hexagons), the less detailed version only shows the relationship as an object and the least detailed version shows a thick black line between the two kinds of objects, whereas in the latter representation a dot indicates the subtype. Depending on the purpose of the figure either of the representations is chosen.

2.4.6 The Upper ontology

The top of the Gellish specialization hierarchy or upper ontology is presented in Figure 30, using the least detailed graphical representation d) of Figure 29.

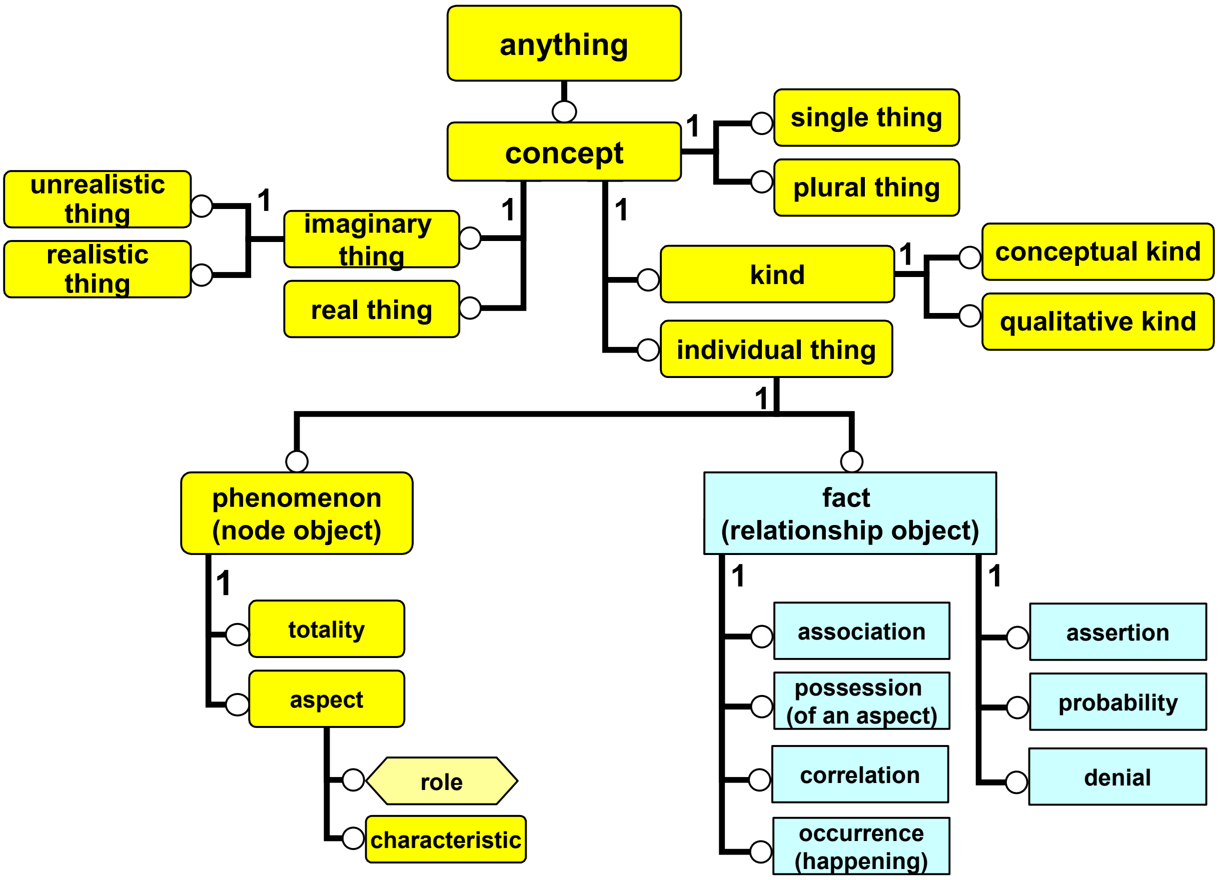


Figure 30, Top of the Gellish specialization hierarchy (Upper ontology)

In this hierarchy the digit 1 indicates that the subtypes are mutually exclusive, which means that something can be a member of only one of the subtypes at the same time. Each of these concepts is discussed in one the following sections.
Note, that the specialization hierarchy is a hierarchy of kinds of things, also called concepts or idea's. Therefore each subtype and supertype role can (only) be played by a kind of thing (and not by an individual thing). This is also apparent from the definition of specialization, as given in Figure 29a. This means that this part of the ontology, with as backbone a hierarchy of kinds of objects, can only represent the right hand 'world' of Figure 26.

2.4.7 Discriminating aspect for specialization

In each specialization relation the supertype conceptually has a discriminating aspect, whereas each subtype has a qualitative aspect that is a qualification of the discriminating aspect. One supertype can have a group of subtypes based on one discriminating aspect and other groups of subtypes based on other discriminating aspects. Different subtypes of the same supertype which specializations are based on the same discriminating aspect have different qualifications (values or grades) of that aspect, whereas those qualifications are usually mutually exclusive.

For example, the specialization of ‘concept’ in Figure 30 is done on the basis of its plurality aspect. This is illustrated in Figure 31.

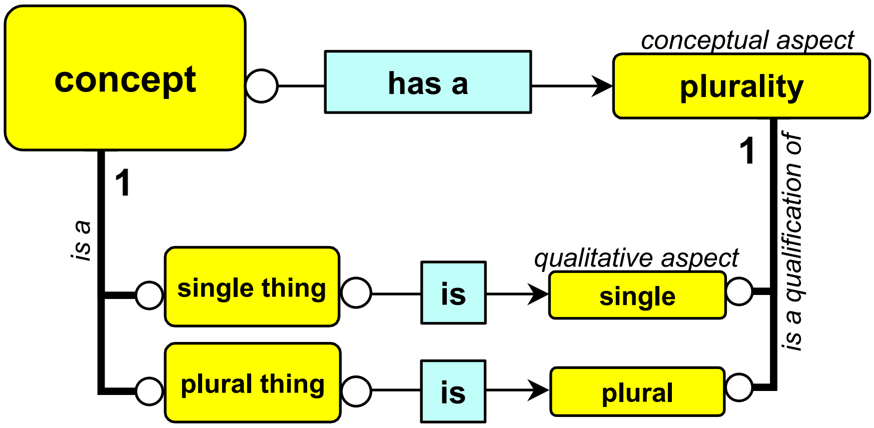


Figure 31, Specialization based on the qualification of a discriminating aspect

Conceptually any ‘concept’ has a plurality aspect. This ‘plurality’ has two qualifications or grades: single and plural. This means that the plurality aspect can be used as a discriminating aspect on which basis the idea of a ‘concept’ can be specialized into two subtypes, called ‘single thing’ and ‘plural thing’, so that a ‘single thing’ has by definition a plurality ‘single’ and a ‘plural thing’ has by definition a plurality ‘plural’.
A similar structure holds for other discriminating aspects, although the number of qualifications can be more than two. The qualifications of the aspects of a concept can be used to determine its subtypes. However, whether those subtypes are actually explicitly defined is a matter of choice. There seem to be no scientific criterion for those choices. It is a matter of good business practice. In general manufacturers of products will recognize more subtypes than incidental users.

The facts that subtypes imply differences in qualitative values of the discriminating aspects can be applied in two ways. First they are applicable for automatic classification. In other words, to determine to which kind of thing something belongs can be done by evaluating the value of its discriminating aspects: if something possesses a plurality aspect that is classified as ‘single’, then it is a ‘single thing’. Secondly, if something is classified as belonging to a certain kind, then it can be concluded automatically that it apparently has the qualitative aspect that defines that kind, plus it inherits the aspects that define the supertypes of the kind. So, being a member of a kind and having particular qualitative aspects are both true and shall be mutually consistent. This means that it is not a dilemma of either specifying subtypes or qualitative aspects, but a consistent combination of both, which correlation enables an automated derivation of the one from the other.

It seems that there exists a certain logic in the sequence of kinds of aspects that are used as discriminating aspects to determine the specialization hierarchy, but as far as I know there is not yet a known criterion that determines whether one sequence is better than the other. In the course of history several philosophers have used different sequences, resulting in different specialization hierarchies. Often there is no reason to select one discriminating aspect above the other, so that various discriminating aspects need to be used in parallel. This is also the case in Figure 30, where ‘concept’ is specialized according to three discriminating aspects in parallel: multiplicity, individuality and reality. The consequence is that further subtypes can have multiple supertypes whereas an individual thing or a further subtype will have multiple qualitative aspects. For example, something is single *and* individual *and* real.
This also occurs on lower levels in the hierarchy. For example, the concept ‘bearing’ can be specialized according to the direction of force of bearing as an axial bearing or radial bearing, but also according to its kind of rolling elements as a ball bearing or a roller bearing. Specialization hierarchies often choose a preference of one above the other in order to avoid multiple supertypes. However, this implies that the other pair of subtypes cannot be defined and thus doesn’t occur in the hierarchy. Therefore it is better in such cases to apply the rule to specialize them in parallel, in which case further subtypes are defined by combination of the two aspects. In case of the example of a bearing this

results in the definition of the concepts axial ball bearing, radial ball bearing, axial roller bearing and radial roller bearing, each of which has two supertypes.

2.4.8 Requirements for high quality definitions

Every concept has a definition. This implies that the concept is related to other concepts. A good definition is not circular. This means that the other concepts need to be defined 'earlier' or 'higher' in the specialization/generalization hierarchy of concepts. A good definition, therefore, requires that a concept has a specialization relation with one or more generalized concepts of which the defined concept is a subtype. This means that in a good ontology, all concepts shall have a position in a hierarchical network of concepts.

The definition of a concept is thus determined by (the definition of) its supertype(s) in the hierarchy, in combination with the constraints on its discriminating aspects. In other words: the definition of a concept is to a large extent determined by its supertypes in the hierarchy of which it inherits all the definitions and thus their constraining qualitative aspects. The remainder of the definition consists of the additional constraints on the discriminating aspects, to which the members of the kind comply and which distinguishes the concept from its 'neighbour' subtypes. By this inheritance of aspects, the lower level concepts in the hierarchy automatically get richer and richer definitions. In other words, the network of relations that define a subtype is always bigger than the network that defines its supertype. The additional aspects that define a subtype have also the role that they specify the additional constraints on the degrees of freedom for the aspects of the further specialized subtype kinds of things.

Everything, including each thinkable concept or observable thing or aspect of something, is a specialization of 'anything' which is most generic. Therefore 'anything' is located at the top of the generalization/specialization hierarchy, as is shown in Figure 26 and Figure 30. Logically, this top of the hierarchy consists of a two elements 'anything' and 'nothing', both of which cannot be defined in terms of other concepts higher in the hierarchy, because such higher concepts don't exist[28].

From top to bottom in the hierarchy every subtype adds a constraint, being a qualitative aspect. Seen from bottom to top every supertype eliminates a constraint, resulting in a more generalized concept.

Once all concepts are arranged in a specialization hierarchy, then a complete ontology consists of additional facts, expressed as additional relations between the concepts, but those relations will not add additional concepts anymore (assuming that also the relation types are included as concepts in the hierarchy).

For the time being the capabilities of computer interpretable artificial languages do not yet include complete explicit expression of all constraints of all concepts. This also holds for the Gellish language. Therefore, it is necessary to allow for definitions of things by a textual string that expresses the constraints in a natural language. However, the richer the language becomes, the more concepts do not require such a textual definition and sometimes not even require a name. This can be clarified as follows:

In theory a textual definition is unnecessary, because each thing is semantically defined by relations with other things that form its context. This can be concluded from the fact that a textual definition actually expresses one or more relations with other concepts that are represented by some of the words in the definition. So a definition of a thing is in principle a collection of facts about the thing, expressed as relations with other things. The most important kind of relation for the definition of individual things is the kind of relation that is used to classify an individual thing (the 'is classified as a' relation). The most important kind of relation for the definition of kinds of things is the kind of relation that is used to define that a kind of thing is a subtype of another kind of thing (the 'is a specialization of' relation). These two kinds of relations provide the major part of the definitions.

[28] Potentially a definition of 'anything' could be: 'anything' is the generalization of everything. A definition of 'nothing' could be: 'nothing' is 'the opposite of anything'. However, this builds on the definition of 'everything' which seems to be a concept lower in the hierarchy as it relies on the idea of 'plural'.

They express either:

- individual thing-1 is classified as a kind of thing-1 that …

or:

- kind of thing-1 is a specialization of kind of thing-2 that …

In the first definition the part indicated by the phrase 'that …' only requires additional criteria that specify in which aspects the individual thing is further constrained than the classifying the kind of thing. In the second definition that part only requires the additional criteria that specify in which aspects the subtype is further constrained than the supertype kind of thing in distinction to its co-ordinate subtypes. If the knowledge base includes 'sufficient' other facts about the thing that express those additional constraints, then the thing is explicitly defined and does not need a textual definition and possibly not even a name. The collection of relations then forms an expression that is equivalent to a textual expression in a natural language.

The definitions of the basic most important concepts from the Gellish ontology are discussed in the following paragraphs. The complete hierarchy, with all definitions of the concepts in the ontology is provided in the Gellish dictionary / taxonomy.

2.4.9 Discriminating aspects used for prime specialization

Figure 30 presented the collection of prime choices of discriminating aspects for the first level of specialization in the Gellish ontology. This paragraph briefly describes the discriminating *aspects* that are used for prime specialization and why those choices were made. The following paragraphs provide further subtypes of those aspects and those choices.

'Anything' is a concept that is unconstrained and therefore 'nothing' is excluded from what it stands for. To determine a prime specialization we need to determine prime discriminating aspects. These top discriminating aspects were determined by defining a generalization hierarchy of concepts. Generalization is a method of elimination of constraints. The most generic constraints that appeared to be left consists of *four concepts with the role of discriminating aspects, each with two qualifications:*

- The phenomenological aspect.
- The reality aspect.
- The plurality aspect.
- The individuality aspect.

Thus for anything, we can distinguish the following four prime qualifications of the discriminating aspects:

1. The **phenomenological aspect,** with two qualifications: **phenomenon** and **fact**.
   This means that a nature of things is determined by the answer on the following question:
   - is something a phenomenon or is it a fact?
   These two qualifications are structural concepts that determine the structure of the ontology, because their roles in the structure are respectively 'node' and 'relation'. Therefore, they determine the identification of phenomena (nodes) and the cohesion between the phenomena (relations), which includes the facts about them, plus the representations of those facts: what exists and what is the case and how that is represented.
   a. A **phenomenon** is further qualified as a **totality** or as an **aspect**
      These concepts are further discussed in section 3.2.
   b. A **fact** can be further qualified on the basis of its *dynamics*, which means that it is either:
      - An ordinary **static fact**
      - A (dynamic) **occurrence** that happens in time.

      If it is a static fact, then it is one of the following:
      - An **association** (or external relation) between equivalent totalities.

- A **possession** (or internal relation) in which the possessed aspect is dependent on the existence of the possessor, and is it thus subordinate to the possessor.
- A **correlation** between aspects[29].

An ordinary static fact is not subject to change, but usually it does have a limited duration of validity, whereas the related objects may vary in quality, grade or intensity. For example the (static) fact that I have a temperature (expressed by a possession of property relation) is a true fact, independent of time during my existence, although that temperature will vary in height. Similarly, an association, such as a connection, can stay in existence, although the connection intensity can vary over time and become loosely connected or strongly connected. A static fact neglects 'irrelevant' changes and only records whether something is the case or is not the case.
If a fact is a dynamic occurrence or an interaction, then the factuality is subject to change over time. An example of a dynamic fact or occurrence is the fact that a process or activity takes place. Continuous processes have a special position. They have a static aspect and a dynamic aspect. For example, a river or stream may have a constant flow rate, but will have a changing transported quantity.
In addition to these distinctions about the nature of a fact (association, possession, correlation or occurrence) we can distinguish its **factuality** or **truth** according to the grades: *true*, *untrue* and *probable*. Furthermore, the knowledge about the fact can vary in degree of certainty.
These concepts are further discussed in section 3.3.

2. The **reality aspect**, with two qualifications that indicate whether something is **imaginary** or **real**.
   a. **An imaginary thing** is a thing that is a product of the mind.
   It can be further distinguished according to its **realism** or realizability, by evaluating whether it is *realistic* or *unrealistic*. In other words whether something is in accordance with the observable or spiritual reality and the laws that apply for that.
   b. **A real thing** is anything that can be observed or can be concluded to be the case from observations.

   These concepts are further discussed in section 3.5.
3. The **plurality aspect**, with two qualifications that indicates whether something is **single** or **plural**.
   a. **A single thing**, has a quantity (number) that is 1.
   b. **A plural thing**, has a quantity that can vary over time. It is usually greater than 1.
4. The **individuality aspect**, with two qualifications that indicates whether something is **individual** or **conceptual**.
   a. **An individual thing** is a thing that has an identity of its own. It can still be real and observable or imaginary, such as fictive persons or designs.
   b. **A concept** or **kind of thing** is a commonality of real or imaginary things. It derives its identity from the things that have the commonality in common. A kind of thing can be further distinguished in a *conceptual* kind and a *qualitative* (or quantitative) kind.
   Examples of conceptual kinds are the concepts 'colour' and 'temperature'. Examples of qualitative kinds are: red, green, blue, 37 degree C, 60 degree F, etc.

The above aspects of 'anything' are discussed in more detail in the following chapter.

In a graphical representation of an ontological model there are essentially three types of things: nodes, relations and networks.

---

[29] A correlation is typically a relation between aspects of the same possessor, for example the properties that are consistent and occur at the same time in a state. In some cases aspects of different totalities are correlated, for example an action force and a reaction force.

1. **Nodes.**
   We represent a phenomenon (a totality or an aspect) by a symbol (such as a rectangle with rounded corners). A phenomenon has a role as a *node* in an information network.

2. **Relations.**
   A fact is represented in such a model by a relationship that is also represented by a symbol (such as a rectangle with sharp corners). A fact therefore has primarily a role as a *connector* between totalities and/or aspects (nodes) and is therefore connected by other nodes through lines. As a secondary role, a fact can also act as a node in a network (which Nijssen called an 'objectified fact'[30]).

3. **Networks.**
   Information is represented by a symbol (a node!), but one that represents a *network* (or *structure*) of facts, which unites totalities and aspects in a coherent 'expression' of complex facts or collection of facts.

The above distinctions are illustrated in Figure 32.

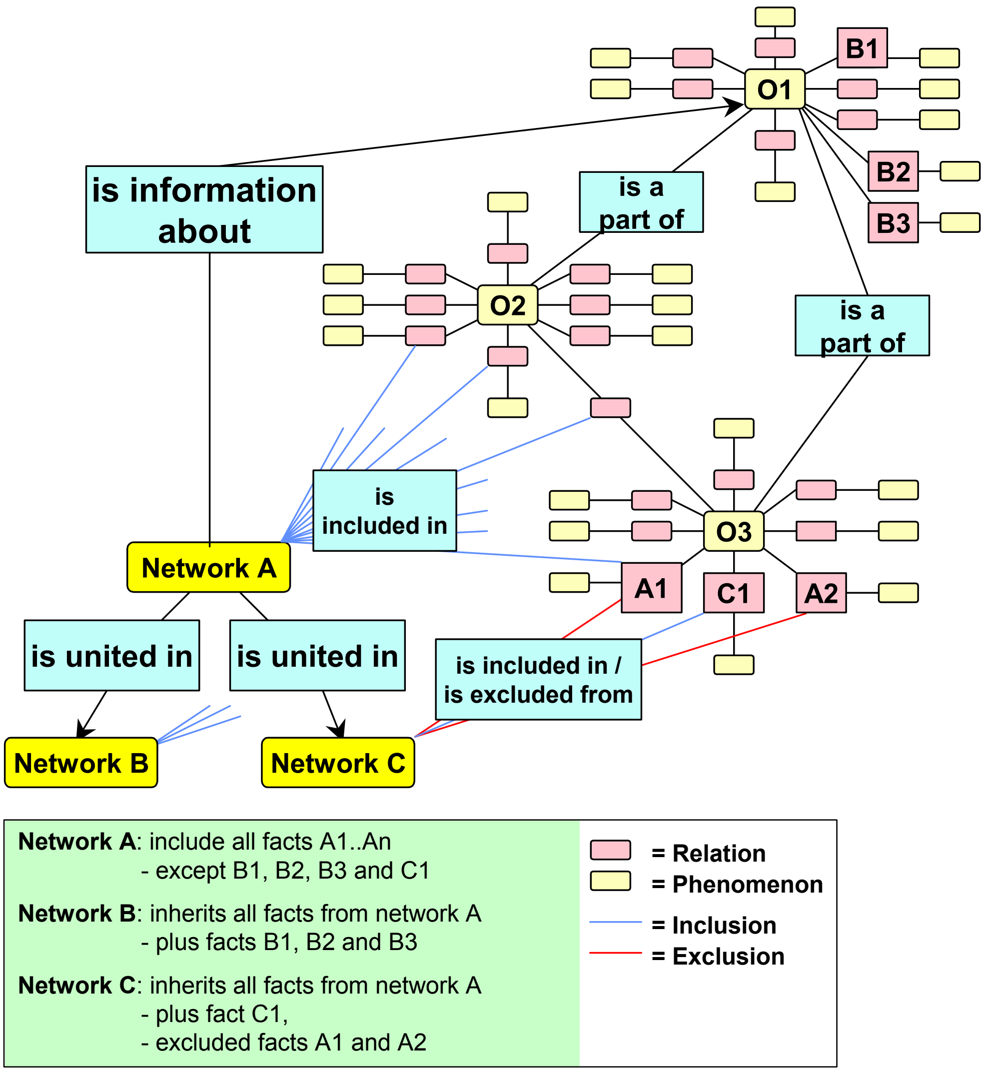


Figure 32, The relation between phenomena, relations (facts) and networks

Figure 32 presents various networks. Some information about an object ('01') is represented by a node (called 'network A'). That node A is clearly distinguished from the other node that represents the object about which the information is given (node '01'). Note that network A is represented by a node

[30] See the History of FCO-IM on http://www.fco-im.com/Literature_Article_History.html

itself, whereas the inclusion relations define the collection of facts that are included in the network, and whereas the expression relations that are included form (the details of) the network itself.

Figure 32 also illustrates relations between networks. For example the ‘is united in’ relations between network A and networks B and C express that all facts that are included in network A are also included in networks B and C (one union relation between collections A and B and another one between A and C). The inclusion relations indicate which facts are additionally included in B and C and the exclusion relations indicate which facts that were included in A are excluded from inclusion in C. That exclusion is superposed on the inclusion via the union relation.

Together these three: phenomenon, fact and information (meaning), represented by node, relationship and network, are the mechanisms used to describe a practical and applicable *structure* of reality (and of realistic imagination) and its behavior and the roles that things have in such a structure.

Note, that a ‘node’ symbol can represent a ‘cloud’ of information (a network) which can be expanded (by ‘zooming in’) to a more detailed level, so that it appears to be a network of nodes and relationships that expresses complex facts[31].

[31] Such a node in a network has a similarity with the role of a circle on a roadmap, where the circle represents a town. At a more detailed level, that town itself consists of a network of streets, which is represented on a more detailed map of that town. In a similar way it is possible that a line on the map that seems to represent a connecting road appears to consist of several parallel lanes with nodes that connect smaller sideways that were not represented on the main map.

3 Details of the Gellish language

3.1 Language definition versus language usage

When we talk about a language, we have to distinguish between the language definition and the language usage. In the following sections the Gellish language is defined through the definition of standard relation types and a dictionary / taxonomy of concepts plus the relations between the standard relation types and the roles they require and the things that can play those roles. Many examples in those sections already illustrate the usage of the language.
This section focuses on a the distinction between Gellish language definition and usage and introduces a few categories of usage.
Figure 33 illustrates the relationship between the Gellish language definition and its usage.

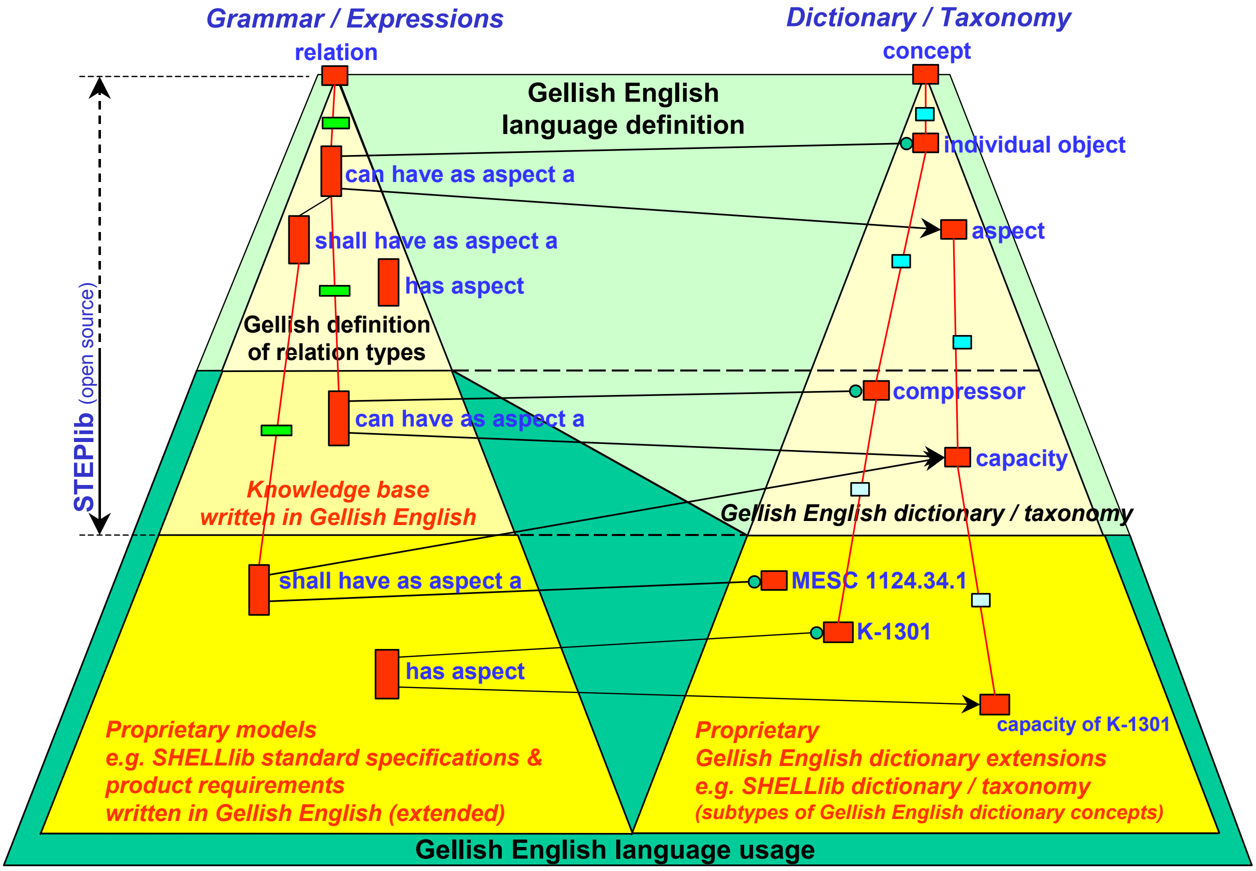


Figure 33, Relationship between Gellish language definition and usage

The combination of both triangles represent facts expressed as relations between objects, written in Gellish.

- The upper part of the left hand triangle represents the standard Gellish relation types (kinds of relations = subtype of 'relation'). They are related to kinds of roles (= subtypes of 'role') which they require, and those kinds of roles can be played by kinds of things. For example, a 'has aspect' relation is related to two subtypes of role, being possessor and possessed, and those roles can be played by respectively an individual thing and an aspect.
- The upper part of the right hand triangle represents the kinds of roles and kinds of role players that are needed to define the semantics of the standard relation types in the upper part of the left hand triangle. Those kinds of roles and kinds of role players form the upper part of the Gellish dictionary / taxonomy. This includes a hierarchy of specialization relations between those kinds of things.

- The middle part of the right hand triangle represents the rest of the Gellish dictionary / taxonomy. The concepts in that part are subtypes of the concepts in the upper part. Those subtypes are required to express further knowledge and information through relations in the middle and lowest parts of the left hand triangle.
- The middle part of the left hand triangle represents the 'open source' Gellish knowledge base that does not add concepts to the Gellish dictionary / taxonomy, but only defines relations between concepts that are already defined in that dictionary / taxonomy. Those relations are classified by the relations in the Gellish language definition in the top of the left hand triangle.
- The lower part of the left hand triangle represents other (mainly proprietary) facts that represent knowledge, specifications, requirements and product models or other information, for example about transactions. Those facts are expressed as relations between concepts in the Gellish dictionary / taxonomy or concepts in proprietary extensions of that dictionary / taxonomy.
- The lower part of the right hand triangle represents the proprietary extensions of the Gellish dictionary / taxonomy. It includes concepts that are considered necessary, but are missing in the Gellish dictionary / taxonomy and it includes identifiers and names of individual things. For example, a proprietary buying specification, coded as MESC 12.34.51 is a kind of thing that is defined as a subtype of an existing concept in the Gellish dictionary / taxonomy and K-1301 is classified as a compressor, which is also an existing concept.

Expressions may only have the status of being 'written in Gellish' when the standard Gellish relation types are used in the expressions and when the concepts from the Gellish dictionary / taxonomy are used, or when proper subtypes of them are used, provided that those subtypes are defined according to the rules of the definition of new concepts. In addition to that individual things may be used, provided that they are explicitly classified as one or more of the concepts in the Gellish dictionary / taxonomy or as a properly defined subtype of them.

Once the Gellish language is defined, the language can be used to express various kinds of facts such as:

1. *Knowledge*, resulting in *knowledge models*
2. *Standard specifications*, resulting in *standard specification models*
3. *Individual products*, resulting in (individual) real *product models* or (imaginary individual) *requirements models*

The expression of these three kinds of information is illustrated in Figure 34.

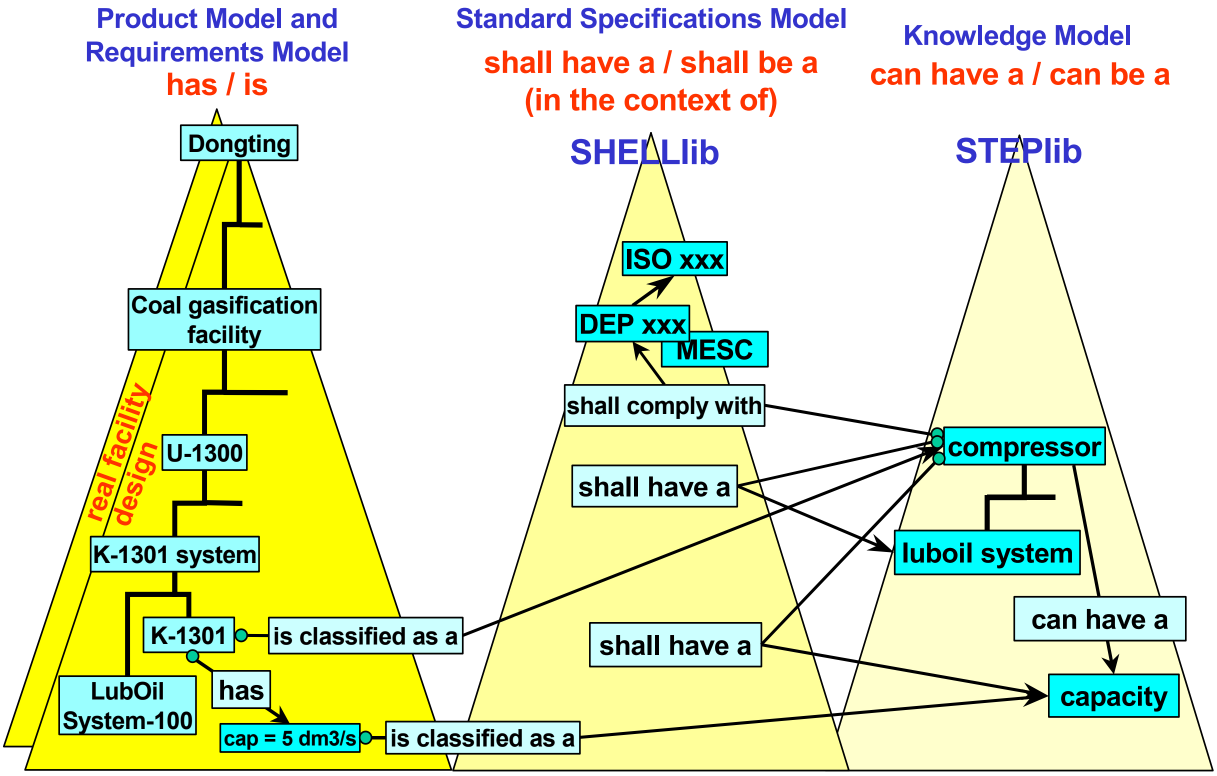


Figure 34, Product, specifications and knowledge models

The three triangles represent expressions that are all 'written in Gellish'.
The right hand triangle represents relations between kinds of things, which kinds of things are defined in the Gellish dictionary / taxonomy. Those relations express generally valid knowledge. Examples of such knowledge are provided as open source knowledge in the Gellish knowledge base.
The middle triangle represents relations between kinds of things that can either be defined in the Gellish dictionary / taxonomy or in a proprietary extension of that. Those relations express proprietary standard specifications (requirements for kinds of things), which specifications are only valid or applicable in a particular context.
The left hand triangles represent relations between individual things or relations between an individual thing and a kind of thing (such as classification relations). Those relations express either facts about real individual objects or facts about imaginary individual objects, such as designs. These two worlds are indicated by the two triangles behind each other at the left hand side.
There can also be a 'realization relation' between a real thing and its corresponding imaginary thing to indicate that the real thing is a realization of the idea.

3.2 Phenomena – Totalities and aspects

3.2.1 A specialization hierarchy of phenomena

Usually an ontology of subtypes of concepts arranges kinds of phenomena, being kinds of the things that are observed in the reality and their equivalent imaginary counterparts, in a specialization hierarchy, according to the inherent common nature of phenomena that are classified by the kinds. The hierarchy is then based on a sequence of intrinsic (internal) discriminating aspects, because extrinsic (external) aspects normally vary over time and thus do not classify what by nature 'is the case'. Such a hierarchy basically includes also kinds of things from metaphysics. In this document this is only partially the case, because this research focuses on technical artifacts, their design, fabrication, usage and classification. Therefore, only specializations of concepts from the general metaphysics are included, but specializations of concepts from the special metaphysics, such as the more specialized concepts in ethics and religion are not included in the ontology. However, there is reasonable evidence that the methodology is also suitable to describe things and 'facts' or beliefs about such things and kinds of things. Differences of opinion about the nature of such things can be accommodated in the ontology by their qualification with aspects such as their reality, certainty and the recognition that an expression of a 'fact' in essence represents either a private or a common opinion or conviction. For example, angels can be included in the ontology because they are assumed to have an individuality and can be classified, although depending on the opinion or conviction whether they are real or imaginary they may be classified differently. This means that for atheists and theists the position of the kind (the concept 'angel') in the ontology is identical, although the reality aspect of individual instances will be qualified differently.

Depending on their insight, different philosophers have created different ontologies. This results in different hierarchies of specializations of concepts. Sometimes this is caused by different preferences and choices on the sequence of the kind of discriminating aspects that are used as a basis for specialization. In the Gellish ontology it is possible that different choices are made simultaneously and that thus multiple specialization hierarchies are integrated in one total hierarchy. This means that different subtypes, based on different discriminating aspects, are integrated in one hierarchical network. This is possible, because one choice for a discriminating aspect usually does not exclude that other choices are also allowed. The multi-parent capability of the ontology then enables to define further subtypes that combine the multiple aspects and those subtypes therefore have multiple supertypes. The result is that the ontology has the form of a hierarchical network and does not have a pure tree structure.

The Gellish ontology is primarily intended for practical application, although it can also be used for scientific research and aims to be scientifically sound. For that reason, it does not contain all possible hierarchies. During its development certain choices were made on the basis of 'best practice' in business and technology, but these choices are always subject to further discussion. Those choices are usually made explicit, which makes them available for improvement, so that it remains possible to strive for synthesis of various views from different perspectives, as long as they do not exclude each other. The result therefore does not claim for 'the truth', but only claims to provide a practical common artificial 'language' that applies essential characteristics of natural languages and that is extensible with new concepts and with which new facts and new kinds of facts and knowledge can be expressed. Furthermore, it is possible to define new concepts that are only valid or applicable within a limited context and that have a name and a definition that is only applicable or valid within such a context. This implies that different concepts can be included that are indicated by the same name, although in different contexts, which makes them homonyms. Because homonyms are allowed in this ontology it is unnecessary to debate about names or to force that different concepts have to have unique names. Agreement about a common language is primarily agreement about the existence of different *concepts* and their relationships towards each other, after which we can agree to indicate them either by the same name or (in different contexts) by different (synonym) names, provided that those contexts are made explicit[32].

A **phenomenon** may present itself to us as a more or less independent **totality** ('entirety'[33]), which we experience as an individuality with its own identity. Other phenomena present themselves to us as an **aspect** of such a totality or as an aspect that is at the same time an aspect of another totality, because of the interaction that we observe between totalities. In the following paragraphs we discuss these concepts of totality and aspect in further detail.

---

[32] A more detailed discussion about the distinction between common concepts and common names, synonyms and homonyms was given in section 2.2.1 of this study.

[33] I prefer the term 'totality' above the term 'whole' or 'part', to indicate the concept that is meant, because the latter two refer to roles that are only defined in relation to each other, whereas what is meant is not the role 'whole' or the role 'part', but the players of those roles. So 'whole' and 'part' are context dependent and not intrinsic. De terms 'entirety' and 'individuality' are also candidates for being a name of the intended concept. Disadvantages of the terms totality and entirety are that they may suggest that small particles would not be totalities, whereas every 'individuality', big or small, is intended to be included in the concept. The term 'individuality' has as disadvantage that it seems to include every individual thing including individual aspects. However, it is intended that all individual aspects are excluded.

3.2.2 Totalities

This section discusses the nature of a totality in relation to its aspects and parts. A totality has a certain degree of independence in its existence in time[34]. An aspect is for its existence dependent on at least one totality that 'possesses' the aspect. An aspect of a totality cannot be 'disconnected' from its possessor. Furthermore, a totality has a nearly infinity number of aspects. Totalities are not only macroscopic entities, also microscopic things, such as atoms or subatomic particles or waves are totalities that can exist more or less on their own or can be seen as parts of bigger totalities of which they are a component.

I prefer the term totality as a name of this concept above for example the term 'whole', because the term whole is better suitable to indicate a *role* that is played by the totality in a relation in which another totality plays a role as a part. Those part-whole relations, or composition relations, are discussed below. Here we can conclude that a totality is not a totality because it is a whole, but because it is to a certain extent an independent individuality that *has* aspects and that *is not* an aspect (the role 'part' and the role 'whole' are examples of aspects that are played by totalities in composition relations). For example, a building is a totality. It has a certain independence for its existence. Nevertheless that independence is not absolute. A building cannot exist without the earth on which it stands, nor is it independent of the molecules from which it is composed, nor is it independent of the laws of nature that cause it to remain in existence, etc. On the other hand it has a different nature when compared with its aspects. Such a building has a nearly infinite number of aspects. For example, it has a shape, it has roles or functions in various situations and usages and we can distinguish many relationships between the building and many people, it has many relations that vary over time and the building has many relations with all its parts, etc.

"Parts and Moments", or totalities and aspects.

What do we mean with a composition of a totality? Does it include the relations between a totality and its aspects? And does it include relations between aspects? As is said above, the term **'part'** (or 'component') as well as the term **'whole'** is in most cases a name for a kind of **role** that is played by a totality in a composition relation with another totality that plays the opposite role (we ignore here for a moment that those concepts are also used by analogy for compositions of other kinds of individual objects, such as occurrences). Every role, part or whole, is an aspect of the totality that fulfills the role. Smith and Simons however use the term 'part', in line with Husserl, not as a role, but as an indication of a role player, being a totality itself. Unfortunately they do not use a separate term for a totality in distinction from the roles whole and part. This probably caused that they do not make sufficiently clear what the nature is of the totalities in the various kinds of part-whole relations. Therefore, in my opinion they use the terms 'part' and 'whole' in a too wide sense as an indication of a generalization of various kinds of roles in various kinds of relations, without sufficient arguments why these are all correctly called (subtypes of) part-whole relations. This suggests that those kinds of relations would be subtypes of a generic composition relation, whereas the nature of that generic composition relation is unclear. Certainly there is a generic composition relation, which for example is the supertype of an assembly relation that indicates that a totality is part in a whole totality, as is the case with a physical assembly. It is already questionable whether this generic supertype has as subtype a collection relation which relates an element to a collection, because if the supertype relates two single things, then a more constrained subtype cannot relate to a plural thing. Furthermore, in complex phenomena of correlated aspects, in which a number of aspects can be distinguished that have some correlation with each other I question whether it is correct to say that those aspects have a composition relation with the 'total' phenomenon. My impression is that in such cases the various aspects appear to be correlated by a physical law, whereas each aspect is involved in, or subject to, the law, but is not 'part of' some bigger thing. An example of such a questionable part-whole relation ('composition relation') is a vibration, which might be thought to be 'composed' of a frequency and an amplitude. However the mechanism of such a 'composition' is difficult to define. It seems better to consider the vibration not as an

[34] It is debatable whether a real totality can be observed directly, or that its existence can only be concluded on the basis of the observation of its aspects. Therefore, it is debatable whether the name 'phenomenon' is an appropriate name for a concept that is a supertype of totality as well as aspect.

composition of aspects, but as a movement (occurrence) described by a law in which both aspects are correlated. Another example is a lower limit and upper limit of a range. The limits are not parts of the range, but a totality possesses a range and also possesses limits, whereas the range and the limits are correlated. I see no reason to define this correlation as a subtype of a composition relation.

In my opinion several kinds of relations are sometimes called composition relations because of an *analogy* with true composition relations in which an assembly or collection activity or a natural creation process preceded the formation of the whole totality. This analogy is an insufficient reason to assume that there is a common supertype relation that should be called 'the' composition relation and thus should be the supertype of all kinds of relations that only have some analogy with 'true' composition.

This is a first reason why in Gellish for example the 'possession of aspect' relation is not defined as a subtype of the composition relation.

Another question is: what is the nature of a totality that is a plurality. Is it one or is it many? Peter Simons (1986) defends the idea to distinguish between two kinds of relations: one relation that indicates that a totality is an element in a 'collection as aggregate' and another kind of relation to allocate a totality as being an element of a plural totality or plurality, which he calls 'manifold'. I think that this distinction is clarifying, because it makes clear that some aspects are aspects of a collection as a totality, whereas other aspects are aspects of each of the elements from the plurality (manifold). For example, the statement that a collection has a mass and a price has a different meaning compared to the statement that in a collection each element has a mass and a price. Semantically the second 'possession of aspect' relation is a plural fact, although those facts may be expressed by one relation, especially if all elements have the same mass (within a tolerance) and the same price. However, I think that this is an insufficient reason to distinguish two different totalities, a 'collection as aggregate' and a 'manifold'. In my opinion the issue is not about the nature of the totalities, but about the allocation of the aspects which is to be reflected in the relation types. Therefore, one kind of composition relation is sufficient to indicate that a (single) totality is element of a plural totality (manifold). But we need two kinds of relations for those possessions of aspects: one to express that a plural totality possesses an aspect (which can be called a 'bulk aspect'), the other to express the plural fact that each element of a plural totality possesses the same (or an equivalent) aspect. These two kinds of relations imply two kinds of roles for the plural totality: a possessor and 'plural possessor' role.

Another question is whether a collection or composition of (all) aspects (sometimes called 'moments'), which Husserl called a whole, coincides with the concept of totality.
Although such a collection of aspects has a role as a whole (and is therefore often called a whole), this does not make it a totality. The basic question is whether there is a distinction between the aspects and the possessor ('bearer') of those aspects. In other words: is a totality 'composed of' its aspects or is the relation between the totality and its aspects of a different nature, because the collection of aspects and the totality are of a different nature? What are the constraints that apply for the things that are related to each other through a part-whole relation? I think that an essential criterion is that things that play the roles of part and whole need to be of the same kind (at least to be of kinds where both are subtypes of a common supertype for which the part-whole relation is semantically defined). Therefore, the question whether a whole that is a collection of all aspects (of something) is more than the sum of its parts (the element-aspects) is correlated with the question whether a 'possession of aspect' relation is a subtype of a part-whole relation. If that is the case, then there is no fundamental difference between a totality and an aspect, and then the term totality is a synonym of 'collection of all aspects of something'. In other words: 'something' and 'the aspects of something' coincide. However, this means that aspects would possess themselves and there would be no mechanism which determines which aspects should belong to the collection. As these consequences are illogical and counter intuitive I reject the idea that a totality would be the sum of its aspects. Therefore, I also reject the idea that the possession of aspect relation would be a subtype of a composition relation.

This means that in my opinion a coherent collection of aspects is not identical or does not coincide with a totality, because a totality is something else than the sum of its aspects. This implies that a totality cannot be derived from its aspects and this justifies that the concept 'totality' is an independent concept next to the concept 'aspect'. If we would have adopted the other option, then the concept phenomenon would become a (single or plural) aspect, the concept of totality would become a totality of aspects (or 'whole of moments') and the concept aspect would become single aspect or partial aspect (or 'part of moment').

Figure 35 illustrates a part of the top of the Gellish hierarchy of totalities.

phenomenon

- totality
  - physical object
    - wave
    - matter
      - material
        - solid item
          - artefact
            - single component item
            - assembly
            - equipment item
              - mechanical equipment item
        - batch
          - batch of fluid
            - batch of liquid
            - batch of vapour
          - batch of solid
    - social entity
      - organization
    - life form
      - plant
      - animal
      - person

Figure 35, Illustration of a part of the specialization hierarchy of single totalities

Figure 35 indicates a subtype/supertype hierarchy of various kinds of physical objects that are defined as subtypes of totality. This includes waves, solids and fluids and including social and biological objects. In the above hierarchy social entities, such as organizations, and biological entities or life forms, such as persons, families, animals and plants are defined to be subtypes of physical object, according to their physical aspect. The fact that there are good reasons to state that such individual things, and especially persons, are more than just physical phenomena can be expressed by defining that these concepts are subtypes of 'physical object' that have an additional constraint on the discriminating aspects of being alive or dead and social or not. Expressions about for example a person who might exist without being a physical phenomenon (when we talk about life after death) requires that those concepts (.e.g. the concept person) is defined independent of being physical or not. This means that those concepts should be direct subtypes of 'totality' and are thus arranged next to 'physical object' (which differs from the hierarchy of Figure 35). This would mean that intermediate concepts are required that are subtypes of physical object as well as of social entity or life form. For example a 'living person' is a subtype of 'physical object' as well as a subtype of 'person'. If we would recognize no other persons than living persons, then we limit our language to physical creatures only. This would also mean that historic persons should be classified as living persons, although that classification was only valid in the past.

In addition to discrete items the ontology also includes bulk items or batches. Subtypes of that are, for example, batch of fluid and batch of solid. It can be argued that these are not intrinsic kinds, because the phase (such as liquid or solid) can vary dependent on the temperature and pressure of the material. However, this holds for nearly all kinds of physical objects, because nearly all physical objects only exists under nearly atmospheric conditions. Nearly all things disappear e.g. under solar conditions. Therefore kinds of things have 'intrinsic' aspects and are therefore classified according to the 'normal' phase under atmospheric conditions. Therefore also kinds that depend on physical phase are included in the ontology.

All totalities that are (partially) made (or shaped) by man are of kinds that are specializations of 'artifact'. Natural items such as mountains, sees and oil reservoirs are solid items or bulk items.

3.2.3 Phenomena as nodes in a semantic network

The phenomena and the relations between them are presented in a graphical representation of the Gellish model by a **network** that describes the structure of the real as well as the imaginary world. The phenomena are represented in that network as nodes in which connections come together. The connections between the nodes represent the relations between the phenomena. We distinguish two subtypes of the concept phenomenon: totality and aspect. Therefore, a node represents either a totality or an aspect. A node only 'stands for' the thing that it represents and does not represent any information about the thing that is represented, because information *about a thing* is something else as the thing itself about which information is given. So, a node in the network represents a 'thing' or phenomenon itself, whereas the part of the network surrounding that node represents a 'cloud' of information that describes the phenomenon and at the same time that cloud forms the context for the interpretation of the elements in the cloud as was illustrated in Figure 32.

This way to describe the structure of reality differs from many other descriptions, as many descriptions do not distinguish between a thing and the information about the thing in such an explicit way. The information about a thing not only includes descriptive information and relationship with other things, but also its names in various languages and language communities, as well as aspects that define the thing. Especially in information technology it is common practice to interpret a limited collection of information about a thing (an 'entity' with its 'attributes') as a representation of the thing itself. For example an 'object' in the object-oriented methodology is in fact not a representative of an 'object' in the reality, but it only represents a limited amount of information about an object, whereas often the object in the reality is not represented explicitly. The consequence is that in the object-oriented methodology it is possible that multiple (information) objects exist which all provide different or overlapping information about the same (implicit) object. In the Gellish language every object is explicitly represented by one and only one node in the total network and that node has an explicit unique identifier. In addition to that, there are in principle an unlimited number of 'clouds' of relations around that node, that represent collections of facts or information fragments about the thing that is represented by the node.

Because a node does not contain any information about the thing which it represents, it seems that the nodes which represent totalities have a similarity with what Kant called a 'noumenon' or a 'Ding an sich', although it is questionable whether Kant considered a 'noumenon' to be a subtype of 'phenomenon'. If Kant would have presented the concepts which he used in a model with defined relations between those concepts, then that might have provided the required clarity (see figure A, B in Plantinga (2000), page 9 and following).

3.2.4 The existence of totalities

In philosophy there is a view (a 'traditional' interpretation of Kant) that states that totalities do not exist next to aspects, because not more than only aspects can be observed. In that view, the synthesis in which a conclusion is drawn about the 'existence' of a totality is considered to be a pure human abstraction. On that basis the idea is that a totality only consists of a collection or sum of aspects, called the 'aggregated appearances' which only 'exists' in the mind of people. In this view, a totality is therefore only a 'plural aspect'. This reductionistic principle is opposed by those who are of the opinion that the reductionistic principle has no solution for the question what is the criterion to decide which aspects form a totality. They are of the opinion that a totality is more than the sum of its aspects. In this holistic view a totality is experienced through the observation of some of its aspects, whereas the totality is the carrier of the aspects. In the Gellish ontology a totality is defined according to the

holistic principle, so that there is a fundamental difference between a totality and an aspect. This implies that a collection of aspects remains distinct from a totality, whatever the size of the collection may be. So, the Gellish ontology accepts the view that a totality is more that the sum of its aspects. Therefore, the normal relations between a totality and its aspects are possession relations.
For example:

- T1 has as aspect A1

However this does not exclude that the reductionistic view can also be used in Gellish. In the reductionistic view a totality is reduced to a collection of aspects, also called a plural aspect. The Gellish ontology keeps the option open to define a collection of aspects, in which case the relation between a plural aspect (a reductionistic 'totality') and the composing aspects is a collection relation. An example of such a relation is:

- T1 is the plural whole for A1

The philosophical debate about the existence of totalities indicates the requirement for the ability to record (the opinion about) the existence of phenomena. In the Gellish model the occurrence of a unique identifier only means that something 'exists' in the sense that there is 'something' about which is and can be communicated. It is still open whether it is imaginary or real. Even the real existence or non-existence of 'it' is information about 'it' and that information is not implied by the occurrence of the identifier. The information about the real or imaginary nature or existence of a phenomenon is recorded as a relation that expresses the fact about the begin or end of existence of the thing and by recording the relation with its reality aspect.

Similar information is provided about the 'existence of a fact' by relations that record the begin or end of validity of a proposition (see the definition of begin or end of validity of fact).

3.2.5 Aspects

Aspects can be distinguished in 'intrinsic' aspects and 'extrinsic' aspects. Intrinsic aspects belong to and are possessed by a totality, more or less independent of the environment of the totality. Extrinsic aspects are roles that are played by totalities in the relations (static facts) or in the occurrences (dynamic facts) in which the totalities interact with other totalities. Examples of such roles are: the role of part, of whole, of being connected, of controller, performer, tool, being subject to, etc.

The philosophical school called "The Philosophy of the Cosmonomic Idea" (Wijsbegeerte der Wetsidee) that followed from the work of Herman Dooyeweerd, performed a systematic analysis of the aspects of totalities. The resulting ontology provides an interesting collection of kinds of aspects that are claimed to be mutually exclusive and are arranged in a hierarchy. They are called 'modal aspects of cosmic reality' (see Herman Dooyeweerd (1953) vol. I, page 3 and 4).
These kinds of aspects are included in the Gellish specialization/generalization hierarchy as subtypes of the concept 'aspect'. In sequence from low to high level they are:

1. numerical aspect (of quantitative aspect)
2. spatial aspect
3. kinematical or movement aspect
4. physical aspect
5. biotic aspect
6. psychical aspect
7. logic aspect
8. historic or cultural aspect
9. linguistic aspect
10. social aspect
11. economic aspect
12. harmonic or esthetic aspect
13. jural or legal aspect
14. ethic or moral aspect
15. pistic or faith aspect.

Dengerink has argued that also time is an aspect of reality which should be added to the list and that should precede the other kinds of aspects. So, the time aspect can be added as follows:

0. time aspect.

It is questionable whether time is an aspect of reality because a totality does not 'possess' that aspect in the way as it possesses other aspects. Time 'exists' independent of the totalities and all totalities are subject to the same, single, time. Because time seems to be a pre-condition for existence of things it might need another position in the ontology. Because of the foundational nature of time it could be argued that it is a direct subtype of phenomenon. In that case the relation between a totality and its period of existence is a kind of relation that is not a subtype of possession of aspect, but a direct subtype of relation. On this basis I define existence of a thing as an 'existence in time', expressed as a relation between the thing and a period in time or between a thing and a start time, possibly followed by a relation with an end time. For the expression of this fact it is irrelevant whether time is defined as a subtype of aspect or not. Therefore, as a compromise between various views, time is included in the Gellish model as a kind of aspect, without defining which thing is the possessor of the aspect.

The above mentioned philosophy claims that the above hierarchy of kinds of aspects is complete and that there are no aspects that do not belong to one of the categories. The Gellish ontology includes all above-mentioned kinds of aspects, so that in principle, if the claim is justified and when the appropriate subtypes are included in the Gellish dictionary/taxonomy, it is possible to express any kind of aspect of reality in Gellish and to relate it to its possessor. Nevertheless, for the time being, this research focuses on technical artifacts and their use by human beings in an economic business environment. Therefore the Gellish ontology is specialized in most detail in area's related to the scope of technical artifacts.

Figure 36 presents a summary of Gellish upper ontology of aspects as a specialization hierarchy. Especially in the area of physical characteristics the full ontology contains much more subtypes. A definition of each of the concepts (kinds of aspects) is given in the Gellish dictionary / taxonomy (as published on the Gellish.sourceforge.net website).

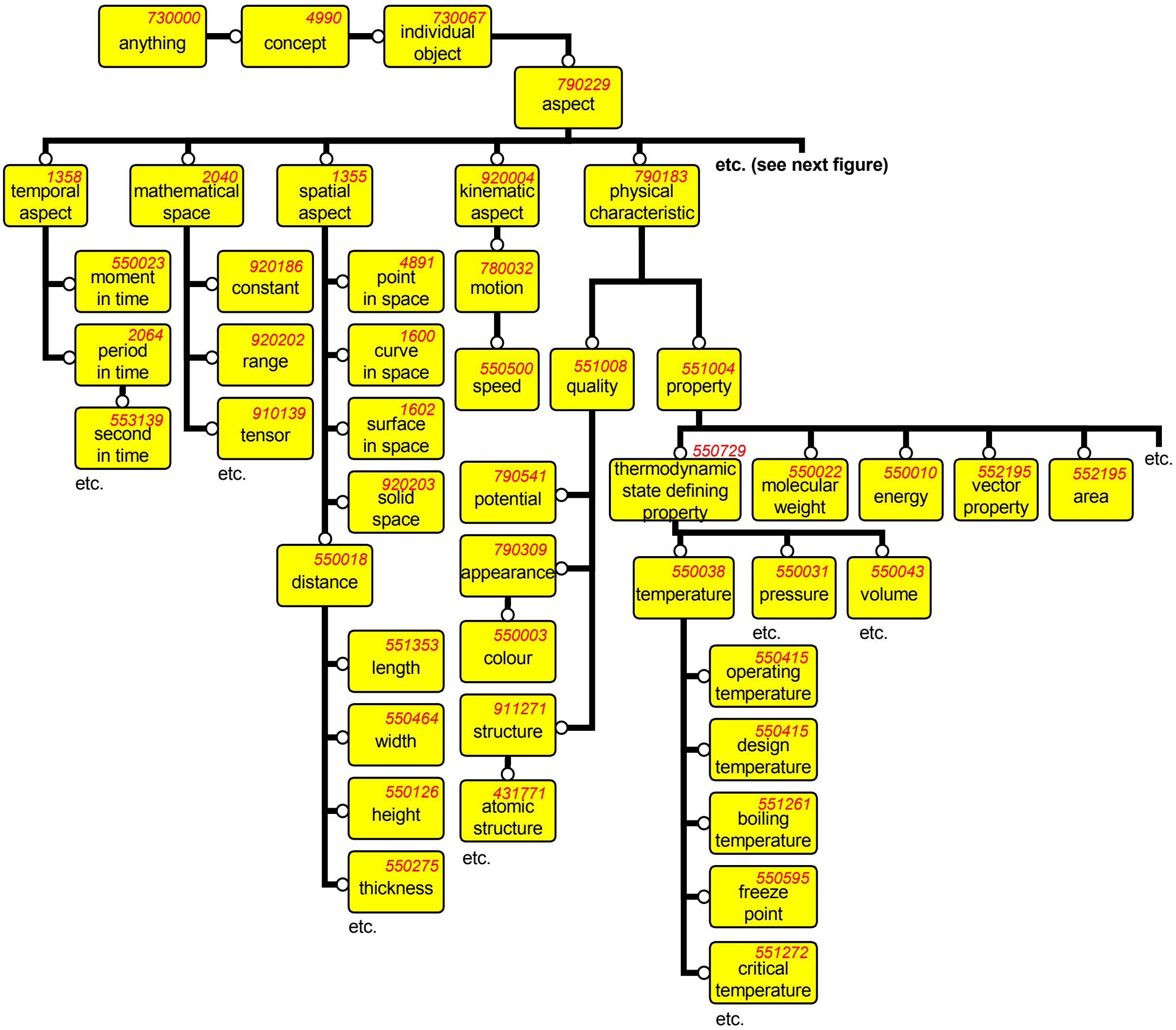


Figure 36, Upper ontology of aspects (part 1)

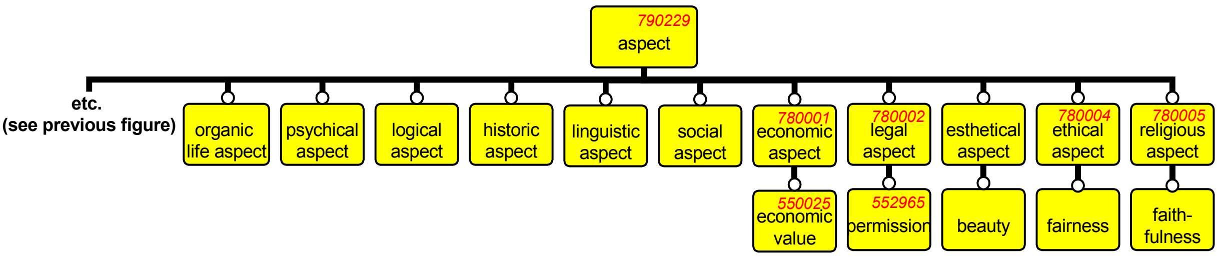


Figure 37, Upper ontology of aspects (part 2)

Next to the above-mentioned kinds of aspects we distinguish kinds of totalities ('objects' in restricted sense) that are primarily qualified by one of the kinds of aspects.

This means that we distinguish:

- Temporal objects (or aspects; see above), such as the year 2000, 1-1-2000, or the concepts year and second.
- Numerical objects, such as numbers, parameters and mathematical correlations, that are primarily qualified by the quantitative aspect,

- Spatial objects, such as empty spaces between atoms, or the space that is occupied by an object, or that was occupied after the object was removed.
- Kinematical objects, such as vibrations or a steady, straight motions,
- Physical objects, that are primarily qualified by the energy that they possess and exchange with their environment.
- Biotic objects or life forms (organic life objects) that are primarily qualified by the aspect that they are alive.
- Etc.

The above philosophy claims that a totality higher in the hierarchy has also aspects in all lower categories, but not the other way around. For example, a tree is primarily qualified as a life form, but has also physical aspects, it can move, it takes space, it is a single thing and exists in time.

The ontology is extended by defining subtypes of these kinds of aspects and kinds of totalities, whereas correlations or laws relate the subtype aspects to each other.

It is remarkable that in the above list of kinds of aspects the physical aspect (number 4) is only one of the aspects, next to other kinds of aspects. This view allows that totalities don't need to be primarily qualified by their physical aspect but can also be characterized as being primarily of another nature. For example, a painting can be primarily qualified as an esthetic object, although it is also a physical object. The concept totality is in this view independent of its prime qualification by one of these categories. A totality can be a (primarily) physical object, but for example, it can also be primarily qualified as a legal object. Therefore the concept totality is not a synonym of physical totality. The choice to keep the option open that a totality is not physical enables that the ontology includes totalities that are qualified as numerical object or as a (pure) spiritual totalities. This means that such an object, for example a 'ghost' can have a position in the general ontology without the requirement that there is agreement on beforehand about the question whether such a thing is an imaginary (psychically) qualified thing, a physically qualified thing (a thought) or a real, but spiritual thing. Constraining the Gellish ontology and language to the physical world would be a reduction of the scope of the ontology that is unnecessary and therefore that idea is not adopted.

Stafleu (1989) has given an extension of the ontology of the Philosophy of the Cosmonomic Idea by providing a partial specialization hierarchy of the first four kinds of aspects, up to and including the physical aspect. The essence of his ontology of number and of spaces is presented as a specialization hierarchy below and is included in the Gellish ontology.

3.2.5.1 An ontology of number

The numeric aspect or quantitative aspect is an aspect of a totality that indicates its number of things and therefore indicates its quantity. Each totality has an (individual) numeric aspect. For single totalities this aspect is always 1, whereas for plural totalities, also called pluralities, collections or 'manifolds' (Peter Simons), it is always greater than 1[35]. Also the property '1 mole' or a 'molal quantity of 1' is a quantitative aspect, because one mole is defined as a number of molecules that is equal to the constant of Avogadro.

In the Philosophy of the Cosmonomic Idea numbers are seen as individual things that are defined as independent things that are derived from the numeric aspects of things. This means that they are defined as totalities which themselves have numeric aspects. They obey their own laws as they are subject to mathematical laws. However it is unclear to me how such numeric totalities relate to their numeric aspects. In my view numbers represent the commonality of the numeric aspects of things.

[35] It can be argued that the number of elements in a collection can be variable and can be either 0, 1 or more. Mathematicians generally recognize only one 'empty collection' with zero elements. However, in Gellish it is allowed to define various collections, whereas the number of elements of each collection can vary over time. For example, various collections of stock items can be defined. This means that there can be various collections that have zero elements at a certain moment in time. Mathematical objects however appear to be kinds of things (and not individual things). So, indeed there is only one qualitative kind of number, being a kind with zero elements.

Therefore, I think that numbers are kinds of numeric aspects. For example, the concept ‘number’ is a kind of aspect (= subtype of aspect) that can be used to classify the conceptual nature of the quantitative aspects of totalities.

Various kinds of numbers, their definitions and coherence can be arranged in a specialization hierarchy. The top of that hierarchy has the following structure:

number
    whole number
        natural number
        negative whole number
    real number
        rational number
etc.

The numbers themselves, such as 1, 3, -25, 0.31, pi, etc. are common numeric or quantitative aspects that can be used to classify the individual quantitative aspects (the distinction between ‘conceptual aspects’ and ‘qualitative aspects’ is discussed in more detail in section 3.7.2.3). For example, the concept ‘three’ is a qualified kind of numeric aspect that can be used to qualify the quantitative aspect of collections with one more than two elements.

In the Gellish ontology, the upper ontology of numbers of Stafleu is combined with the definitions of mathematical concepts as defined in ISO 10303-50 into an integrated hierarchy of specializations of the concept ‘mathematical space’[36].

3.2.5.2 Mathematical operations and functions

Apart from numbers, we distinguish ‘laws’ to which numbers obey, as Stafleu also does.
Mathematical laws can be distinguished in (dynamic) mathematical *operations* or assignments and (static) mathematical *correlations*. Operations progress in time and generate output from input. They appear to be subtypes of calculation, which is a subtype of activity. Mathematical correlations indicate a time independent relation. The latter can be used to describe facts about numeric aspects, which facts can be true or untrue. Both operations and correlations relate numeric objects or kinds of numeric objects.

For example, a part of a hierarchy of kinds of numeric *operations* is the following:

numeric operation
    addition
    subtraction
    multiplication
    division
    root determination
etc.

[36] The mathematical concept ‘space’ has to be distinguished carefully from the three dimensional concept ‘space’ that is an aspect of the physical world, because the term ‘space’ as defined in mathematics only indicates a collection (set) of numbers or a collection (set) of groups of numbers. The mathematical term ‘space’ is therefore a homonym that only has some analogy with the name of the concept space as a three dimensional part of the universe.

An example of a hierarchy of mathematical *correlations* is given in the following hierarchy:

mathematical function
    equality function (a = b)
        a = b + c
        etc.
    inequality function (a ≠ b)
    greater than function (a > b)
etc.

In the Gellish ontology we combined the ontology of kinds of mathematical correlations as described by Stafleu with the mathematical functions that are defined in the above mentioned ISO standard into an integrated specialization hierarchy of mathematical functions. The kinds of mathematical operations are combined with a specialization hierarchy of activities or occurrences.

3.2.5.3 Ontology of space

The specialization hierarchy of spatial aspects as described by Stafleu can be combined with geometrical concepts (for example those of ISO 10303-42) and 1, 2, or 3-dimensional spatial concepts as defined in ISO 31. An outline of the combined result will be about as follows:

```
spatial aspect
        shape
            triangular shape
                        isosceles triangular shape
                                equilateral triangular shape
                quadrangular shape
                        rectangular shape
                                square shape
                elliptic shape
        distance
                length
                   diameter
                        internal diameter
                        external diameter
                   thickness
            width
            height
            depth
etc.
```

By turning these aspects into independent (imaginary) totalities one can think of things that exist on their own as parts of the three dimensional space. For example, a triangle can be seen as a spatial totality with a triangular shape as its characterizing aspect. In a similar way the concepts such as a quadrangle, ellipse, pyramid, ellipsoid, and a geometric model can be seen as spatial totalities because they are qualified primarily by their spatial aspect. They are imaginary because only totalities that have also temporal and physical aspects belong to the real totalities. These spatial totalities not only have a shape, but they also have a size and quantitative or numerical aspects. In other words, they are not aspects, but have aspects. For example, a rectangle has a rectangular shape, a length, a width, a circumference, an area and is singular. It can also have relations to other things, such as a position and orientation in space.

3.3 Facts and their dynamics – Ordinary facts and Occurrences

3.3.1 Atomic facts

A *fact* can be defined as *'that what is the case'*. This definition includes denials, as it may be the case that something is not the case. *In Gellish each fact is expressed as one or more relations between* *things.* The expression of facts includes also opinions (propositions), because expressions can always be qualified by others as an opinion. This seems to be in line with what Wittgenstein (1961) states in his 'Tractatus': "the world is the collection of facts, not of the things" and a fact "is the existence of a connection between things"[37]. These statements stress that things do not exist in isolation, but always exist in relationship to other things. Only through those relations we get knowledge about the things. In the above statements the term thing includes everything which is apparent in the world. This includes also imaginary things, being products of the mind. It also includes aspects of real or imaginary things, which aspects are to be distinguished from the things of which they are aspects.

In positivism, it is considered that only facts (and thus things) 'exist' that are derivable through logic from observations that are done according to the methods of physics. This also seem to be the position of Wittgenstein when he states (in statement 6.42 and 6.421) that nothing can be said with certainty about esthetic and ethical aspects. Nevertheless, it is difficult to deny that things have esthetical aspects and that especially human acts have ethical aspects. Even chemical plants have esthetical aspects and doing business is subject to ethical considerations. So, we consider those aspects also part of the 'world' that need to be covered in the ontology and thus need to be described. Even supernatural things are subjects about which statements are made and can be made. That is the case, independent of that fact that it can be debated whether Gods, gods or angels and godly design or objectives and acts are imaginary things or real things, being either part of 'this world' or part of 'another world'. When agnostic people express their thoughts that we cannot say anything worthwhile about those things, they nevertheless make denying statements about a fact about those things, which statements are considered worthwhile by them. It is very difficult to prove such statements, as the possibility to prove something in this respect is already excluded on beforehand. Nevertheless, in human communication, even in the communication of those who deny the 'existence' of such things, those things do occur (see for example the Tractatus, statement 6.522), although those people are usually of the opinion that those concepts have to be excluded from (scientifically) justified statements. It is true that we cannot record the facts themselves, but only our knowledge or opinion about the facts. Therefore, an ontology and the associated semantics, is an expression of our knowledge and opinion about real and imaginary facts, with the aim that it is in accordance with those facts. Therefore, we will not constrain the Gellish ontology by excluding things that can be said, as Wittgenstein seems to do when he talks about statements that are not worthwhile. It shall be possible to express all 'facts' that we can derive or can think of. This means that the semantic expression capabilities of Gellish allow the expression of all kind of 'facts', even if others are of the opinion that they are not 'facts'. Therefore, expressions of 'facts' can have various levels of *truth* or *certainty* and thus can be qualified as untrue, probable, improbable or unsure. Of some facts it is generally agreed that they are uncertain, such as the uncertainty of location and time of an elementary particle, as is described by the uncertainty relation of Heisenberg. In other words, Gellish also allows the expression of opinions and probability, which are considered as 'facts' that have certain level of certainty and probability, or have a validity (only) in some context. Furthermore, there are facts of which the degree of *reality* varies and that can be qualified as imaginary, realistic, unrealistic, not realizable, etc. These include, for example, design decisions about imaginary objects, even if those objects are unrealistic or unrealizable. They include also misunderstood 'facts' about reality or imaginary facts about an imagined reality, irrespective whether it is a realistic imagination or not. Sometimes it might even be on purpose that the imaginary 'world' is not conform reality or does not conform to the laws of reality.

So, in the Gellish definition of fact there is no constraint that something is a fact only if there is (absolute) certainty about its being the case. It may be thought of being the case only. This allows for expressions about reality even if reality appears to be different than the interpretation of the expression suggests, as is often the case, even in scientific expressions. Our ontology can only reflect our *image* of what is the case and thus is only a *model* of that. Our image of reality is something else as the reality itself and we cannot prevent that an ontology can contain parts that are not conform the reality of which it is a model, simply because of the fact that our knowledge and understanding is limited.[38]

---

[37] Ludwig Wittgenstein (1961), Tractatus Logico-Phylosophicus, statement 1.1, 2 and 2.01.

[38] In this context it is remarkable that in Jewish and Christian religion it is acknowledged that there is a reality of which it is forbidden to make a model, because such a model is impossible. This is based on two elements: one

Thus, it appears that relations can not only be used to express true facts, but they can also express possible or potential facts or uncertain opinions about ‘facts’, expressed as propositions, which include expressions of opinions about what is thought to be the case. Therefore, the Gellish language and ontology uses relations (relationships) as the basis for the expression of facts and opinions, whereas a fact is expressed as a relation that is qualified as an expression of a fact (using the rule that an unqualified relation is intended to be interpreted as a fact), and where an opinion is expressed as a relation that is qualified as an opinion. When we hereafter talk about facts we mean facts and/or opinions.

3.3.2 Expression of facts by relations between things

Networks of atomic facts

Every atomic fact is represented in Gellish by a thing that is an atomic relationship between two or more things. Each atomic relationship is expressed as a collection of elementary relations. Elementary relations are binary relations each of which describes a relation between the atomic relationship and an involved thing and the role that the involved thing plays. The number of elementary relationships is equal to the number of things that are involved in the relationship. In other words, an atomic fact is expressed as two or more involvement relations, whereas each involvement expresses the role that the thing plays in the relationship.

Most facts appear to be expressed by two elementary (binary) relations and the roles played by the two involved objects can often remain implicit. Therefore Gellish allows for an expression of those facts by an abbreviated notation as one atomic binary relation, in which the two elementary relations are implied.

The linguistic expression of the abbreviated notation of such an atomic fact then becomes the same as the notation of elementary facts. Both are presented in Gellish English as instances of the conceptual relation:

- something ‘is related to’ something

whereas the relation can be either an (abbreviated) atomic relation or a (detailed) elementary relation.

Graphically, this can be presented as a formal graphical representation of a data model or schema according to the conventions of a particular method. For example, NIAM schema’s or EXPRESS-G (as defined in ISO 10303-11). This document only uses figures with informal illustrations, because it is considered arbitrary and confusing to try to model all the semantics of Gellish in graphical symbols. For details about the notation in the figures see the chapter on Nomenclature.

A graphical representation of an instance of the above expression is given in Figure 38. The whole figure (indicated by the ellipse) represents a binary fact. The chain of three related things presents an expression of the fact, whereas the two roles played by the involved two things are implicit.

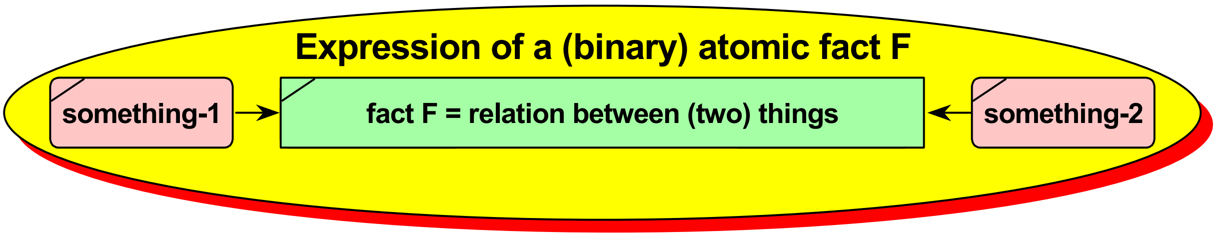


Figure 38 , Binary atomic fact, expressed as a relation between two things

The two arrows indicate elementary facts. The fact that we use an arrow indicates that in natural language, as well as in Gellish, an expression by a relation has a direction for reading. Semantically, a

---

is the story about the told observation that God said and has carved in stone that people are not allowed to make a model of him. The other is the story about the revelations recorded in the bible that express that his personality is above imagination. Nevertheless, it is also accepted that God revealed ‘facts’ about himself, see the second commandment in Exodus 20: 3 en 4.

fact is direction independent, because it is one fact, but syntactically the expression is direction dependent: there are two equivalent expressions of the same fact and those two expressions are each other's inverse. The inverse expression is linguistically another form, but with the same semantic content. The two expressions are each other synonym. Gellish allows both expressions and recognizes an inverse expression for every expression, whereas it recognizes also the unique identity of the direction independent single fact. So each standard relation type in Gellish has one identity and (in each natural language) at least two expressions that are the names of the relation type that are each other's inverse.

For example, from one side one can read:

- component-1 is a part of composite-1

whereas read from the other side the expression of the same fact would be:

- composite-1 is a whole for component -1

In this example, the two expressions represent a single fact, being a composition relation. The single relation type that classifies the relation has two inverse names: 'is a part of' and 'is a whole for'. Gellish has defined a unique identifier for this relation type, being the arbitrary number 1190.

Facts are connected to each other because their expressions share the use of the involved things. Those involved things therefore can be regarded as forming nodes in a multi-dimensional *network of relationships* and such a network expresses knowledge or opinion about the modeled world.

Figure 39 introduces a few more specialized kinds of relations.

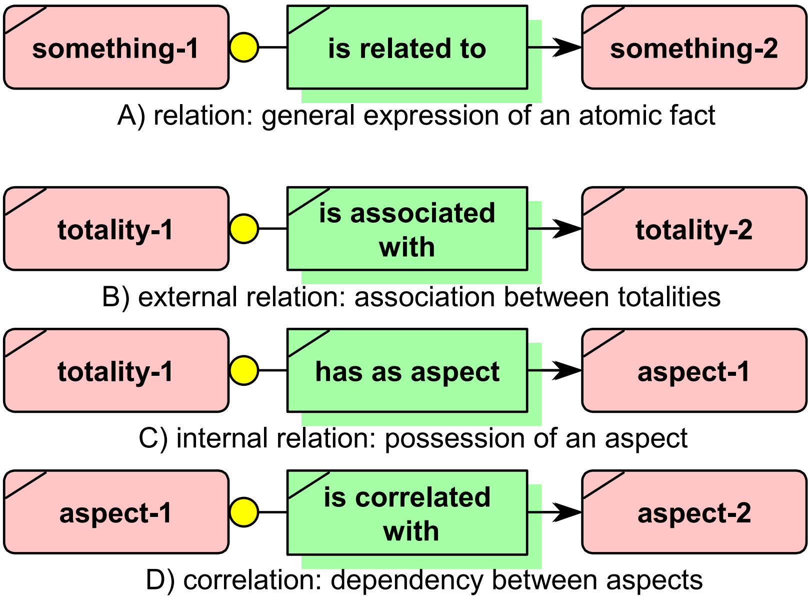


Figure 39, Representation of kinds of atomic facts

The general case for a relation between things is given in Figure 39A, whereas Figure 39B, C and D represent specializations (subtypes) of the general one. Relations of different kind require roles of different kinds. Each of those kinds of roles can only be played by a particular kind of thing. That is the reason why different things are mentioned on both sides of the relations of the different kinds:

- An association is defined as a relation that requires two roles that can only be played by totalities.
- A 'possession of aspect' relation is defined as a relation the requires a role of possessor that can only be played by a totality and requires a role of possessed, which role can only be played by an aspect.
- A correlation relation requires more than one role of being correlated, which roles can only be played by aspects.

  Note that the roles in a correlation can usually by played only by aspects of the same possessor. This fact is expressed by another relation that expresses that the correlation is valid for the aspect of that particular possessor. That relation is not shown in Figure 39. For example, the correlation between a mass, a volume and a density is only valid if those three aspects are possessed by the same physical object. This constraint can be expressed by a relation between the correlation and the physical object.

The names of the things in the boxes in Figure 39 (with the added numbers) indicate that the boxes represent individual things, which is also indicated by the line in the top left corners. If we generalize this to the general cases, then the left hand and right hand object of Figure 39A become 'anything' and of Figure 39B, C and D the kind of things become totality and aspect.

This is illustrated in Figure 40, which represents the definition of kinds of relations, which require different roles, whereas each of those roles may either point to the same kind of thing or to different kinds of things, dependent on the semantics of the kind of relation. This is reflected in the usual convention in graphical representations of conceptual data models where different kinds of roles of the same kind of relation may point to the same kind of thing.

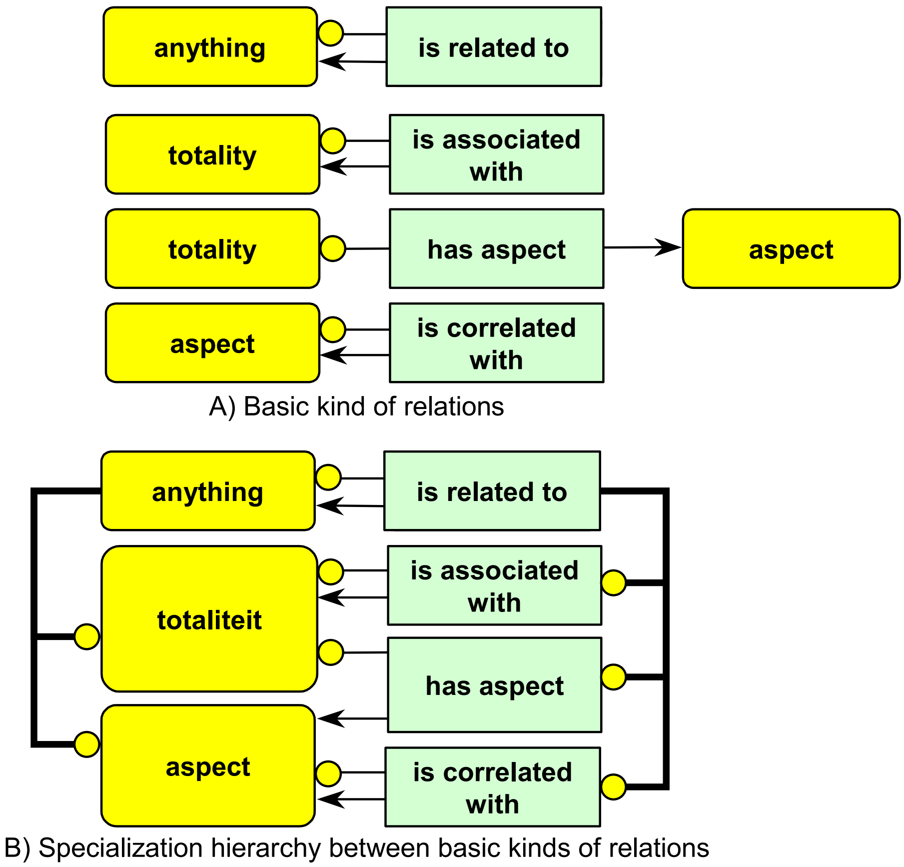


Figure 40, Models of binary facts (general case)

Often something can at the same time only have one or a limited number of relations of the same kind with something else of a certain kind. This means that such a thing has a constraint on the number of roles of the same kind. This so called simultaneous cardinality constraint is expressed in Gellish as a constraint on the number of roles of a kind that is specific for the kind of relation. Such a cardinality constraint can be context dependent. For example, in a particular context the number of pumps that are allowed in a pump system may differ from the number of pumps in a pump system that is allowed in another context. The modeling of cardinality constraints is discussed in more detail in section 3.3.9.3.

3.3.3 Elementary facts and roles in relations

The expression of an atomic fact is the smallest complete proposition. Nevertheless an atomic fact is usually composed of four or more elementary facts. This is illustrated in Figure 41.

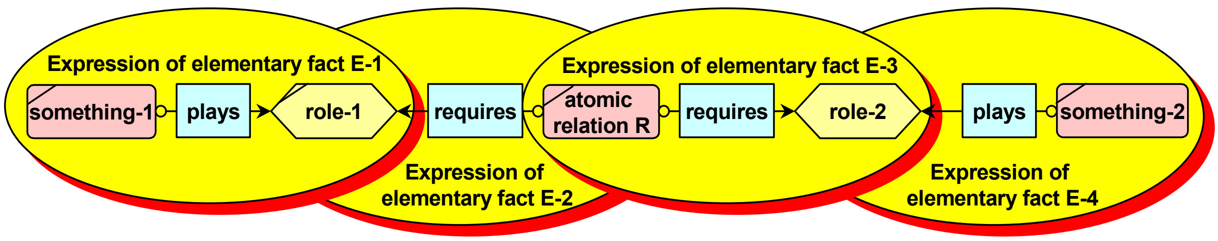


Figure 41, Elementary facts in an atomic fact

An elementary fact is expressed by an elementary expression, which expresses that a role is played by a thing that participates in the fact.

An elementary expression is *the smallest meaningful proposition about what is (or what is not) the case.*
There are two kinds of elementary facts:

1. an elementary fact that indicates that something plays a role (in a fact).
2. an elementary fact that indicates that a fact requires a role (of a particular kind).

Elementary facts are expressed as binary elementary relations. The first elementary fact can be *expressed* as a relation between the thing that plays the role (in the fact) and the role that is played by that thing. The second elementary fact can be *expressed* as a relation between an atomic relation and a role that is required in that relation.

This means that the following two kinds of elementary facts can be distinguished:

- something plays a role (elementary fact E1 and E4 in Figure 41)
- a relation requires a role (elementary fact E2 and E3).

For example, the fact that wheel-1 has the role of part in composition relation R1 can be expressed by the following two elementary relations:

- wheel-1 plays role-1
- relation R1 requires role-1

In both relations, role-1 is of the kind 'part' and in the second relation R1 is of the kind 'is a part of' (indicating a composition relation).

The *definition* of an 'is a part of' relation states that such a relation requires a (left hand) role of the kind 'part' and a (right hand) role that is of the kind 'whole', whereas the two roles shall be played by two different individual objects. This definition can be used to verify the semantic correctness of the elementary relations.
The two elementary relations share the use of the same role-1, so that they can be combined into a consistent part of the partial atomic relation (E1 + E2):

- wheel-1 has a role as part in an 'is a part of' relation

The combination with the equivalent expression(s) for the roles of the one or more other objects in the relation (in this example E3 + E4) results in an expression of the complete atomic fact (E1 + E2 + E3 + E4):

- wheel-1 has role-1 in relation R1 which requires also role-2 played by car-1

By classification of relation R1 as an 'is a part of' relation and of wheel-1 as wheel and of car-1 as car, the semantics of the expression is defined completely, because the definition of the 'is a part of' relation specifies that role-1 is a part and role-2 is a whole.

The semantics of the complete atomic fact can be expressed in abbreviated form (where the roles remain implicit) as:

- wheel-1 is a part of car-1

or in general:

- something is a part of something else

Linguistically such an expression can be further decomposed in smaller components, such as words, characters, character groups, phonemes or glyphs. However, such parts are grammatical components and not semantic components (although some duplicate semantics seems to be present in some of those components), whereas those components form incomplete expressions of the elementary facts. Therefore, we consider an elementary fact as the smallest *semantic* unit, which can be called a *'semantic primitive'*. Elementary facts occur hardly in isolation. They are always a component of an atomic fact. In binary facts, we usually don't need to make them explicit, because the relation (with its directed expression) defines unambiguously which roles are played by which objects. Therefore, we make elementary facts only explicit in the expression of ternary or higher facts, especially to specify the roles of objects in occurrences and in correlations.

The semantic components of *kinds of* atomic facts are defined by the kinds of roles that are required by instances of those kinds of atomic facts and the kinds of objects that can play those roles. Therefore, these *definitions* require the elementary facts to be made explicit.

In the practical application of the Gellish language the kind of relation (relation type) determines whether the relation is an elementary or an atomic relation. This means for example that in graphical representations one binary atomic fact can be expressed either by three related boxes or by five related boxes, dependent on the kind of relation that expresses the fact and the level of detail that is intended to be made explicit.

3.3.4 Complex facts, messages, product models and templates

The explicit *expression of information* that is of greater complexity than atomic facts can result in a model that is a large and complex network of atomic facts. In practice it is experienced that certain kinds of atomic facts often appear in groups. Together such a group expresses a complex fact. An example of a group is the fact that something has an aspect, which aspect is classified and is also quantified by a value on a scale. This is a recurring pattern that is recognizable in the following example:

p-1234	has aspect	d-1		
d-1	is classified as a	diameter		
d-1	is quantified as	20	on scale	mm

Such a pattern in information about an individual thing can be derived from a definition of a collection of facts that form a pattern on a conceptual level. The latter is a conceptual model that expresses knowledge and is thus an example of a little knowledge model. The knowledge model from which a pattern for the above example can be derived is:

physical object	can have as aspect a	property
property	can be classified by a subtype of	property
property	can be quantified by a	number
quantification	can be qualified by a	scale

In the Gellish ontology it is defined that the relation type ‘has aspect’ (as used in the first table) can be a realization of the conceptual relation ‘can have as aspect a’ in the second table. This is specified as follows:

- ‘has aspect’ can be a realization of a ‘can have as aspect a’

Similar realization relations are defined for the other types of relations.
With that additional knowledge it is possible to derive templates for automated support for the generation of instances. For example, the pattern of the second table can be used as a template for the generation of instances, such as the one in the first table.

Note, that in a Gellish Table the column with the word ‘a’, which in the above example precedes the terms physical object, property and quantification, is not shown, because it only appears for grammatical reasons that are typical for the English (and other) grammars. Semantically, the phrase ‘can have as aspect a’ means that ‘a’ thing of the kind ‘physical object’ can have the role of possessor in a relation of the kind ‘can have as aspect a’.

Such typical and repeatedly recurring patterns of atomic facts can be called molecular facts, as analogy with molecules that are composed of fixed patterns of particular kinds of atoms. Such a pattern can be used as a *template* for the derivation of specific instances.

The *completeness of a complex fact or message* cannot be determined on beforehand in an absolute way, without a specification of requirements for the extent of the information (= the extent of the network). This is caused, among others, by the context in which the complex fact is embedded during its interpretation. The complex fact can be a fragment in a database, in which case the user can query for further contextual information. The complex fact can also be a complete message, or it can be a fragment of a message that will be interpreted in the context of a receiver. In the latter two cases the sender of the message can only guess whether the criteria for completeness that he uses are sufficient for correct interpretation by the receiver. The context of the receiver forms the context for interpretation of a message by the receiver and determines whether the sender has expressed sufficient information to enable or ensure a correct interpretation by the receiver. For example, if in the above example the receiver does not know for which object the property is provided, he cannot correctly interpret the message. In that case the receiver has to ask for additional (context) information intended for clarification until the interpreting party can unambiguously interpret the meaning and objective of the message. Such a question for (additional) information as part of a communication process can also be expressed in Gellish as is discussed in section 6.3.3.

The expression of information that describes a certain object as a model with a particular level of detail is called an *object model* or a *product model*. Such models are composed of atomic or molecular facts. Large models of complex objects, such as pieces of equipment or even industrial facilities, can result in very large collections of atomic facts that can be seen as integrated networks. For example, a collection of facts can form the specification of something that is procured, or it forms the fact how a standardized thing is defined, or it is the result of the design of a facility. As far as such collections of facts describe generally valid (conceptual) object models, these models can be characterized as scientific discipline specific partial ontologies, although these models are often not recognized as partial ontologies. Furthermore their scope is outside the philosophical discipline. However, it is important to understand that the methodology to document generic philosophical models and to document less generic technical (ontological) models should be the same, in order to be able to integrate them into one consistent model of reality. This can be achieved by using a common grammar and a common dictionary, which the Gellish language intents to provide.
For a further discussion of product models see my chapter 'The development and use of product models' in "The Gellish English Application Manual" (Andries van Renssen, 1999).

In business it is common practice to make use of standardized *'fill-in-the-blanks' forms* or their electronic equivalent 'screens' or 'windows'. Such types of templates are mainly created to simplify the repeatedly occurring specification of requirements for similar things of the same type. For example, to simplify the specification of similar artifacts, such as pumps, pipes, motors, etc. in various dimensions and capacities. Such a form contains a number of explicit facts that form requirements (and often as many implicit facts) which specify the amount and structure of data about a kind of object or activity (in a certain context), fixing some data and leaving a number of options and quantifications open for variation.
For example, an empty 'data sheet' for a certain kind of electric motor fixes the data that need to be quantified in the context of purchasing a new electric motor of that kind and may fix the standard preferences of a certain company. For example on an electric motor data sheet there may appear a field accompanied by the text 'voltage'. A human being will probably interpreted this as the requirement that (in this context) an electric motor shall have as aspect a voltage and the value of that voltage shall be specified. The Gellish equivalent of such a template consists of a number of facts that express requirements in a certain context. This means that each fact is only valid in a specified context and such facts express requirements of the form '… shall be …' or ' … shall have …'

For example, Company ABC may specify its requirement that in the context of 'procurement by Company ABC'

- (an) electric motor shall have as aspect a voltage

etc.

Such a *requirements model* actually is a partial model for an object of that kind, which specifies requirements for some aspects of the objects of the specified kind and leaves the unspecified aspects free for the one who interprets the requirements, such as the supplier. Such a requirements model can

be integrated in the total collection of facts about the kind, because the general knowledge about such a kind of object remains valid. The definitions of the concepts used in the specification remain valid as well.
Requirements models are usually specialized subsets of knowledge models. Therefore, it is possible to derive templates for instances also from requirements models.

As stated above, such standard forms usually contain a lot of implicit information. Often the objects that are related because of such implicit facts have no explicit relations with other objects on the form. In practice the form is the context for the interpretation of the facts and they cannot be interpreted properly outside that context. For example, only the simple fact that some fact is presented on the form or is presented on a particular position on the form, implies that that fact has something to do with other facts on the form. In order to make those facts computer interpretable and to be able to integrate those requirements in a consistent object model, it is required that such facts are made explicit and become expressed as part of a coherent network of relations.

3.3.5 Ordinary facts (static facts)

This paragraph discusses the expression of static situations or ordinary facts as opposed to changing situations or occurrences, which are discussed in the next paragraph. This reflexes the difference between what is the case and what is the case when something becomes the case. An ordinary fact, which we can also call a static fact, is something that is the case without the occurrence of something that changes that situation. Ordinary facts can be further distinguished in facts that express that something has happened and facts that express that something is the case during the validity of the expression. For example, the statement that something is painted is ambiguous, because this statement can be interpreted in at least two ways: (1) as the fact that an activity to paint something is finished, or (2) as the fact that paint is present on the surface of the thing. Both interpretations describe a static or ordinary fact, but those facts are different facts. To be unambiguously expressed they require different expressions in Gellish.

The expression of the two facts and their relationship is illustrated in general terms in the model of Figure 42.

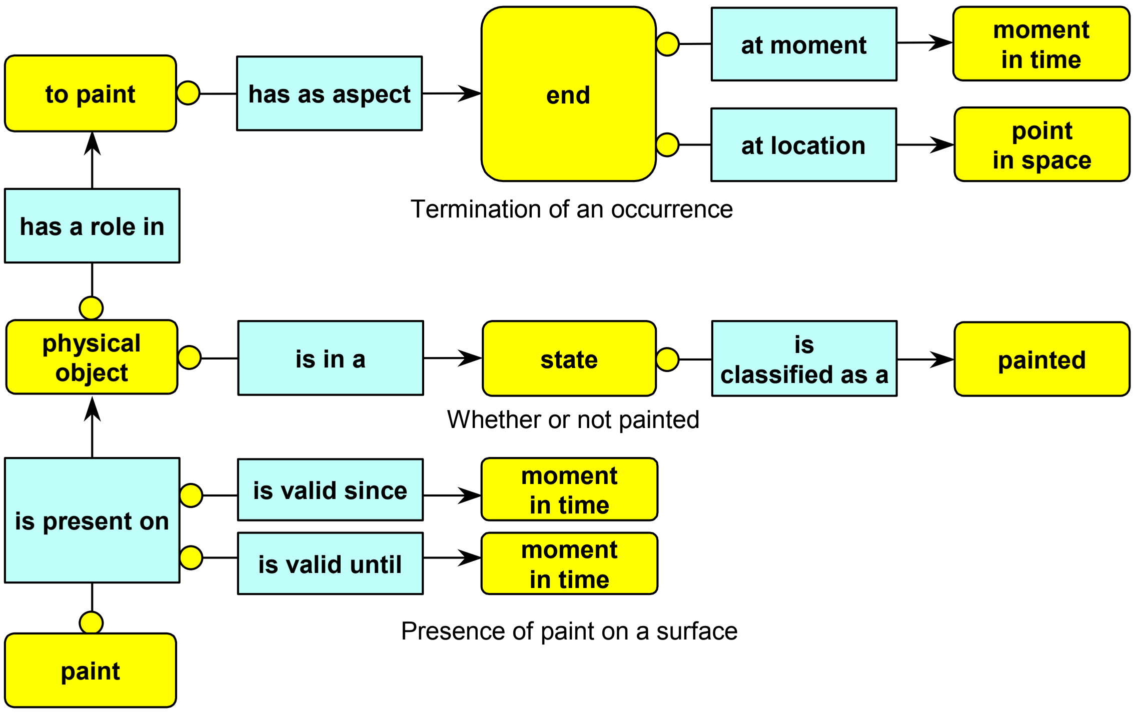


Figure 42, Fact about being painted or not

The first fact expressed in Figure 42 is not influenced anymore by the occurrences after the completion of the activity; it is a *historic fact*, independent of further expiration of time. The first fact is actually a fact from the past. It is described by the *activity* of painting and by the involvement in that activity of the thing that is painted. The fact in question is the fact that that activity is completed. In the Gellish language, such a fact is therefore expressed as an aspect of an occurrence (the activity) that indicates the completion of that occurrence. This aspect can be either a qualification of the state of the activity as being 'completed' or by explicitation of the completion (the end) of such an occurrence, as is illustrated in Figure 42. An explicit completion of an occurrence enables to add further information about the occurrence, for example by indicating the location and time of the completion, or the fact that it triggers the start of another occurrence.

The second fact that is expressed in Figure 42 is the 'being in a *state'* that continues to be the case until the paint is removed or until the existence of the painted thing is terminated. So, the second fact describes a situation in the present, measured at the moment that it is concluded that the state is the case. The state was created in the past and will be terminated in future. It has a period of validity. This second fact is expressed by recording that the painted object possesses a state, which state indicates whether the object is painted or not. That aspect is then classified by a qualified kind of aspect, the value 'painted'. To have such kinds of states available as standard (defined) states in the Gellish language it is required that the kind of aspect as well as the allowed values (the 'domains') for those kinds of aspects are present in or are added to the Gellish vocabulary and thus are included in the dictionary / taxonomy. In the above example, this means that the conceptual aspect 'being painted or not' and the qualifications: painted, not painted and partially painted are included.

Figure 42 also illustrates that the second fact ('the object is painted') can also be expressed in another way, being by an association between two totalities: the paint and (the surface of) the painted object, which association expresses that the paint is present on the surface. The latter kind of expression is not very common for this kind of situations, except when more detailed facts need to be recorded about the second object, the paint. For example, when the thickness of the (layer of) paint has to be recorded or if it is required to describe precisely where the paint is present on the surface and where not. Such expressions require that the paint is explicitly recognized as an object, whereas in the first two expressions the paint does not occur. Apparently there is a correlation between the concept 'painted' and the expression that paint is present on a surface. The one is a definition of the other. By making that correlation explicit, it becomes possible either to generate one expression from the other or to verify the consistency of two different expressions.

3.3.6 Occurrences as dynamic facts

This paragraph discusses the expression of changing situations, or occurrences. An expression of an occurrence is an expression of the fact that something is the case that terminates what was the case and that causes that something else is becoming the case. In other words it is a (dynamic) transition state, between a preceding state and a final state. My definition of a fact as being something that is the case, at first glance seems to be a definition of a static world and seems to ignore that the world is basically dynamic. The whole reality exists only in time and at subatomic level everything is in motion. It often only depends on our perspective, whether we want to consider something from a relatively static or from a dynamic perspective. For example, each thing on earth is moving through the universe, but usually we can ignore that; in a steady stream of water many properties don't change, but some do; the foundation of a bridge give static support, but they move under forces of wind, traffic or earthquake; the storage of a fluid or solid item is pseudo static because storage often influences the composition and quality of the stored object by deterioration, such as forming of deposits or corrosion. So we have to take into account that many 'states' are continuously changing and that many ordinary facts are only valid during a particular period in time. Therefore, my definition of a fact includes the dynamic character of reality. In other words: the definition of a fact as something that is the case includes things that are the case over time, even if they are changing during time.
To include the description of the dynamics of the world in our ontology we have to incorporate occurrences and kinds of occurrences. It also means that basically each fact is related to time, because it started to become valid at a moment in time and remains valid during a period in time. The only exception might be the 'laws of nature' that are considered to be valid as long as time exists. Nevertheless, dynamic states are often described as if they remain (by approximation) the same during a particular period in time and can thus be qualified as a pseudo static fact. This is usually done by taking a macroscopic view or by abstraction from time and validity period.

So, an occurrence is the fact that something is *subject to the effect of time*. In other words, an occurrence is the influence of time on things (plural!) that are participating in the occurrence. It can only be described as an *interaction* between the things that are involved in the occurrence. Therefore, the expression of an occurrence requires the expression of several facts, each of which describes the *involvement* of one thing in the occurrence. Each thing that is involved plays its own *role* in the occurrence, which means that each involved thing has its own function or contribution and should be suitable for the role that it plays.

The fact that something occurs does not necessarily mean that changes take place on a macroscopic level or that everything changes. A typical example is a *'steady state'*, such as a constant and continuous flow of fluid, as approximately 'occurs' in a river. Such a steady state is a dynamic situation that can be described by parameters whose values don't change, but the state is not static. These kinds of states often occur in certain environments. For example, in the process industries, a continuous steady state process, in which a constant flow of fluids enter and leave a facilities is a process that can be described by parameters, such as flow rate, pressure and temperature, that do not change (on a macroscopic level). However, there are certainly changes, for example, the position of the parts of the fluids, the heat that may be transferred and the integral of the fluid that is transported.

The conclusion can be that an occurrence can be regarded as the counterpart of an ordinary (static) fact. Each ordinary fact implies (requires) that every thing that is involved in the fact plays a role in an elementary relation that expresses that the thing is involved in the fact in a particular role. Each role differs from the roles of the other things and is an aspect of the thing that plays the role. Similarly, an occurrence is an interaction between the things that together plays their roles in the occurrence and those roles are aspects of the role players. An occurrence can therefore be regarded as a (dynamic) fact. The fact that a thing is involved in an occurrence can therefore also be expressed as an elementary relation, whereas the semantics of the expression become complete by classification of the things, the roles and the occurrences. In general this can be expressed as:

- a thing can have a role in an occurrence

or in a specific case:

- thing T1 has role R1 in occurrence O1.

The Gellish language allows for the expression of occurrences in various levels of detail. This is illustrated in Figure 43.

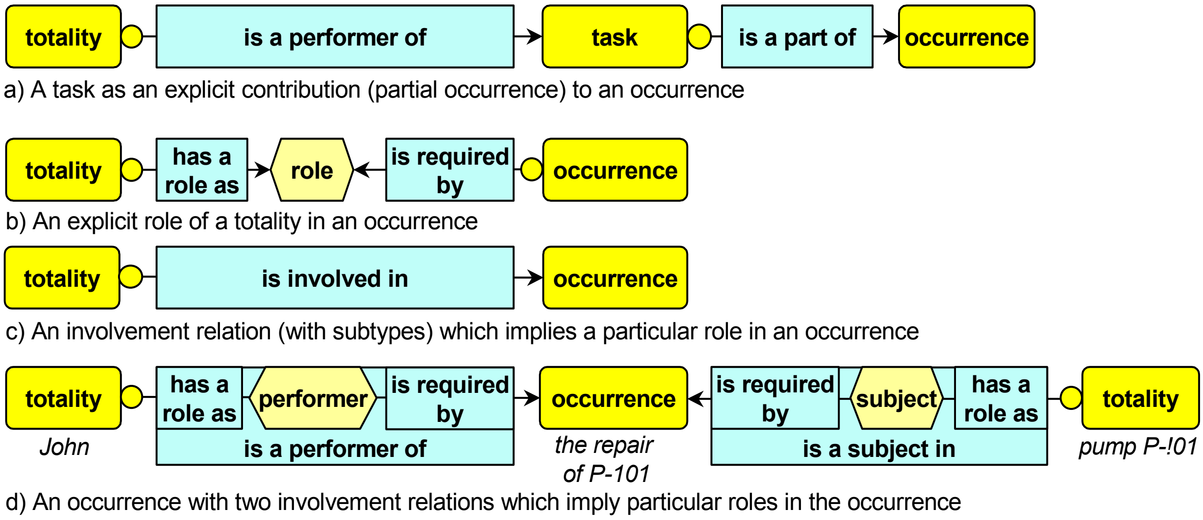


Figure 43, Totalities and their roles in an occurrence

The first expression in Figure 43A illustrates that it is possible to break down an occurrence in partial occurrences, each of which is performed by a particular party. For example, various people that together perform an activity, whereas each of them performs his or her own task. Such a task is also an occurrence, a part of the total one.
Total occurrences as well as partial occurrences can involve several totalities, each with its own role, such as performer or subject or tool or input and output. At a high level, it can be specified that something is involved in an occurrence without stating in which role. For example, Figure 43C illustrates the expression:

- John is involved in repair R

without specifying the kind of role that is played by John. Figure 43B illustrates how the role can be made explicit. The combination of B and C defines the relation 'is performer of', which is a subtype of 'is involved in'. This is expressed in Gellish in the left hand part of Figure 43D as follows:

- John is performer of repair R

The 'is performer of' relation is defined as having by definition two roles: the left hand one is 'performer' the right hand one is 'performed'. So the latter expression implicitly defines the role played by John.

A more complete expression of the occurrence requires also the right hand side of Figure 43D. This expresses the involvement of the other thing that plays a role in the occurrence R:

- P-1 is subject in repair R

The two elementary relations together express the natural language expression (or atomic relation):

- John repairs P-1

Note that the latter expression seems to ignore the existence of the individual occurrence R. However, in full Gellish the individual occurrence also has an explicit identifier, whereas it is recognized that in the above example 'repairs' is a kind of occurrence that classifies that individual occurrence R.

Figure 44 illustrates the main kinds of roles that can be played by objects (totalities) in an occurrence and their accompanying involvement relations. Most other kinds of roles and kinds of involvements are specializations of these types.

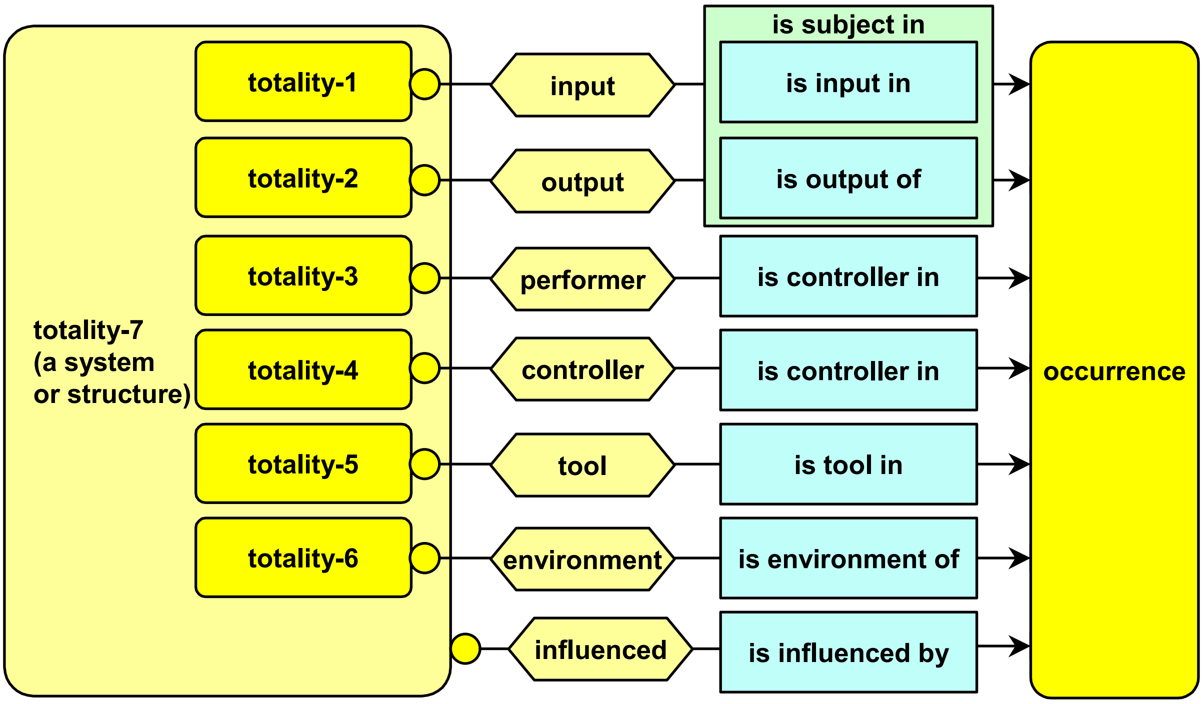


Figure 44, Kinds of roles and involvements in an occurrence

Each totality plays its own kind of role in an occurrence, but in addition to that, all involved things together, including also the thing that has the role of environment for the occurrence can be considered to form a 'system' of things that are related in a structure. The system as a whole has a pre-state before the occurrence commenced and that is influenced and changed by the occurrence (the transition state), resulting in a post state.
These Gellish expressions about an occurrence recognize more roles than the 'Integration Definition for Function Modeling' (IDEF-0) method which focuses mainly on "input, output, control and mechanism", see NIST (1993), whereas in Gellish additional types of roles can be added when the prime ones provide insufficient semantics. The Gellish model for occurrences appears also suitable to represent the various kinds of 'acts' and the roles of various parties in business communication and production as are recognized in the DEMO method (see Van Reijswoud (1999) as is discussed in section 6.3.3.

3.3.7 The relationship between ordinary facts and occurrences - to be and to become

There is a relation between occurrences and facts, because facts are caused by occurrences and their existence is terminated by occurrences. This section discusses how those relations can be expressed in Gellish.

In general, a fact will start to be the case, it will exist, and it will cease to exist. This means that 'the life of a fact' generally goes through the following stages:

1. Occurrence - creation of the fact. This is an occurrence, being a process that causes the fact to become the case. Such an occurrence commences with a state (a situation) in which the fact is not the case, whereas the occurrence is the transition to a new state in which the fact actually is the case.
2. Ordinary fact – a static situation. The transition is followed by a state in which the fact remains the case (possibly for an infinitely short period).
3. Occurrence – termination of fact. The static situation is followed by an occurrence that is a process that results in the termination of the fact. Such an occurrence terminates in a state (a situation) in which the fact is again not the case.

A static view on reality neglects the creation and termination occurrences. Consequently, only the existence and sequence of facts is registered and modeled. This may include the registration of the moment of start and termination of the validity of the facts, which means that validity and history of facts is included.
In a dynamic view on reality the focus is on the transition occurrences from one state into the other (state transition). Something happens in this view.
These two views form each other's complement in one integrated ontology. In Gellish these two views are combined by being expressed in a compatible way, using a single language.
That is done as follows.

An ordinary, static fact exists in time. This means that it is a static aspect of a state. For example, the fact: "the water of the river Rhine flows steadily" is an expression of a continuous state, which does not change during a certain time. It is continuously the case, true. Nevertheless, such a static fact is an expression of a dynamic occurrence. The same holds for other, less dynamic situations, such as the connection between physical objects (a painting that is attached to a wall; a component that is stored in a warehouse, etc.). All these static facts actually are occurrences that happen in time and do not change at a macroscopic level, although microscopically and in exceptional cases (such as during an earthquake) they do demonstrate dynamic behavior.

The above considerations lead to the conclusion that facts about the relations between physical objects are actually interactions that occur in time, in a similar way as the more dynamic occurrences, activities and processes.

As a consequence, ordinary facts and occurrences can be arranged in a sequence of static and dynamics states. In many cases such a sequence of states has the nature of a *cause and effect* succession. An occurrence can be the cause of the start of validity of a fact, which fact can be terminated by a second occurrence. The Gellish phrase that expresses such a 'cause and effect' kind of relation is:

… is cause of … or … has as effect …

A relation of this kind relates an occurrence to a *start or end* of a state (a fact or an occurrence). In separate relations the start or end is related to the fact or occurrence.

This is illustrated by the following example: The process 'connect A and B' results in connection C (fact or state), whereas that state is terminated by another process 'disconnect A an B'. This can be expressed by the following relations:

- connect A and B	has as effect	start of connection C
- start of connection C	is a begin at point in time	T1
- start of connection C	is the begin of	connection C
- end of connection C	is the end of	connection C
- end of connection C	is an end at point in time	T2
- disconnect A and B	has as effect	end of connection C

In this example, the begin and the end of the state are explicitly modeled to enable to express where and when the begin or end took place. This is expressed by a relation with a moment in time and with a location, if required. The 'connection C' is the name of a fact, which can be expressed as:

- A	is connected to	B

Instead of the fact 'Connection C', it is also possible that the effect of Connect A and B is the start of another occurrence, for example a 'Communication C'. So occurrences can trigger the start and termination of other occurrences or they can determine the begin or end of validity of a fact.

The Gellish language also contains standard shortcut kinds of relations that express that an *act* has as result the start (or termination) of a fact, irrespective of the question how, where and when that is caused. These relations express a sequence of states. These relations leave the start and end of a state implicit and do not allow for the expression of time and location. When we use those relation types, the above example becomes:

- connect A and B	has as result the start of	connection C
- disconnect A and B	has as result the end of	connection C

3.3.8 Functions, roles and occurrences

Design methods often recommend to start a design with the definition of the 'functional requirements' or the 'functional specifications' which include a functional decomposition whereas only in a second stage of the design this should be followed by a 'physical design'. See for example the use of the concepts Functional Unit and Technical Solution in the General AEC Reference Model (GARM), often referred to as the 'Hamburger model', Gielingh (1988). However, it is often unclear what is exactly meant with a 'function'. Apparently it is expected that it is intuitively clear which aspects are functional and which aspects are physical. For example, the implementation of the SAP ERP system can cause extensive discussions on the meaning and required instances of attributes of the 'functional location' entity in the system. Such discussions are illustrative for the fact that the meaning of the concept 'function' is far from intuitively clear. Often it is tried to clarify the concept by stating that a criterion for the difference is that the technical solution can vary, while the functional requirements remain the same. However, this criterion is also applicable for a specialization of a technical solution, because a supertype technical solution can also remain the same when another subtype is chosen. A precise definition is very important, because especially with computer assisted design and engineering the question is essential whether design data about the functions and design data about the physical objects should be separated from each other or not and how the relation is between functional data and technical data or data about the physical objects.

So, what is a function? What is a functional design and what is a functional decomposition?

3.3.8.1 What is a function?

The word 'function' in English has several meanings. It can occur as a noun, but also as a verb (in 'to function'). If somebody asks about a thing: What is its function? Then this might be paraphrased as a capability or as a question about the occurrence that it should perform: "What can it do?". In both cases, noun and verb, the expression points to a role of a thing in a bigger totality. If something functions, it always operates in relation to something else. Also when we talk about the function of something, we refer to the contribution of a role player in the bigger context of an occurrence in which also other things participate.

The unclarity increases because of that fact that kinds of artifacts are often named after the intended function or role for which they are designed. For example, a blower is called a blower because it is designed to be suitable to blow. Similarly for a vacuum cleaner, a controller, etc. The English words ending with 'er' (as in blow**er**) typically refer to a role or function.

This is hardly the case for non-artifacts: a tree has no reference to a verb, role or function. Probably this is caused by the fact that non-artifacts are not designed by us, although apparently they do have a function. When people gave names to non-artifacts they seem to have thought in a less purpose-oriented manner.

The word 'function' is also used to refer to a correlation between aspects of things or between parameters in a mathematical equation. For example, the expression 'density is a function of temperature' refers to a correlation between physical aspects.

The various usages of the term 'function' means that we can distinguish at least five different concepts, each of which can be named with the term 'function'. Those concepts are:

1. An **occurrence** (activity, process or event).
2. A **totality** in a particular role or designed or made for an intended role.
3. A **role** of a totality (usually of a physically qualified thing) played in an occurrence.

4. A **correlation**, usually as a physical coupling between aspects (if the magnitude of one aspect changes, then the magnitude of the other necessarily changes as well, in a particular way).
5. A **mathematical relation** between numeric objects, which specifies a mapping.
   Note that a mathematical relation (such as y = a.x + b) differs from a mathematical assignment (such as: allocate the value 5 to a variable). The first is a kind of relation, the latter is a kind of activity, which therefore belongs to the first category.

The following paragraphs will discuss these five different meanings of the word ‘function’ and thus they discuss five different concepts. This should clarify what is meant with ‘the’ function, functional decomposition, etc. and how the relationship between those concepts are. This will also indicate how each of those concepts should be modeled in an integrated way.

3.3.8.2 The term function used to indicate a subtype of occurrence

The term ‘function’ in the first meaning, being something that can, shall or will occur, is frequently used in technology. Functional design (or ‘the design of functions’) is often contrasted with physical solutions, because the specification of functions that do not prescribe the kind of physical objects that should perform those functions is more flexible and increases the chance to lead to the evaluation of a larger variety of technical solutions than with a direct design of alternative technical solutions.
However, there is a lot of obscurity about the way in which these functions should be modeled. Therefore we will analyze the use of the terms function and functional design in the context of engineering to determine how those functions should be properly modeled.

The term function is for example often used in control engineering. In an early stage of the design of a control system, the system is described as a sequence of ‘functions’, which have ‘inputs’ and ‘outputs’ that realize the ‘connections’ between the ‘functions’. See for example the standards IEC 1131-3 and VDI/VDE 3696 (ref. IEC 1131 (1985) and VDI/VDE (1995)). In such an early stage of the design it is not yet chosen *which* physical objects will perform the control, although it is already clear *that* there will be physical objects that will be in that role (possibly software, but software is also a physical object, being an encoding on an information carrier). Apparently, in such a design there will be also physical objects that have a role as input or output (input and output are subtypes of role). The first option is the possibility that in this context the term ‘function’ refers to the roles of the (later defined) physical objects that will be required for the operation of the process. However, this would mean that an input would enter a role (a function). But inputs cannot enter roles. Therefore, the term function in this context cannot refer to a role. The second option is the possibility that the term function here refers to the performer physical object. However, in this stage of the design the physical objects that will play roles as performer of the control are not yet chosen. Therefore, the word ‘function’ cannot refer to such a performer physical object either. This leads to the conclusion that the word ‘function’ in this context must refer to (a part of) the control process itself. This appears to be consistent with the other information that is recorded about the function, such as the algorithm that describes how the function (the process) has to be performed. So, the word ‘function’ in control engineering refers to a process, in other words it refers to an occurrence.

This is illustrated in Figure 45.

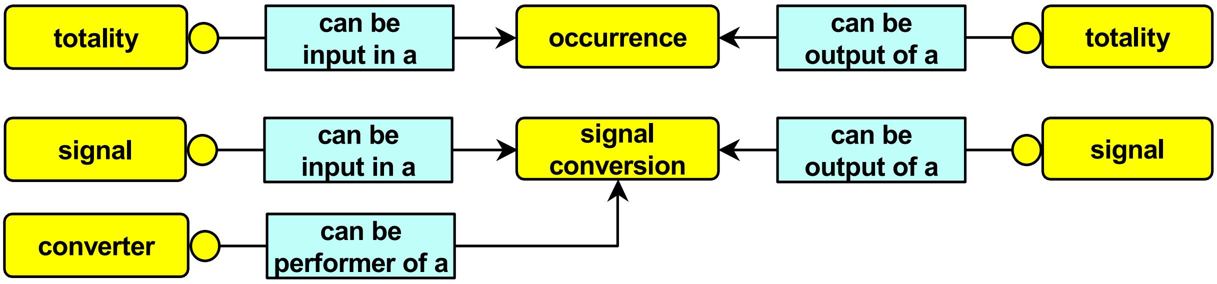


Figure 45, A 'function' being an occurrence (a conversion of input into output)

The upper part of Figure 45 presents the general case of an occurrence, where totalities are subjected to the occurrence with a role as input or are the result of the occurrence with a role as output. The middle part of the figure shows the specialized case of a signal conversion function that refers to a process that converts an input signal into an output signal, whereas signals play a role as input or output. Note, that the process is actually a conversion of the input signal irrespective of the converter that enables or performs that process. Such a totality with the role of converter (or performer) can be added (later in the design process) by an additional relation with the signal conversion process as is illustrated in the lower part of the figure.
It should be noted that in many cases a design is described as if a process variable, such as a pressure, is converted, although the process variable is not converted, but a physical signal with its characteristics, that represents e.g. the magnitude of a pressure, is converted in another signal (with other characteristics). So, a model that represents reality in an accurate way has to model the signal explicitly as a physical object and role player.

When the term (pure) *functional design* is used, this unjustly suggests that only functions (occurrences or processes and roles) will be described and no physical objects. On the contrary, the processes are phenomena that convert input physical objects into output physical objects. So, the properties of the input and output physical objects shall be described in functional designs as properties of physical objects and not as properties of the functions! For example, in control engineering, the measured and controlled things are physical objects and the control processes are physical phenomena that can only occur in or with physical signals. *This means that the term 'functional design' refers to the design of (partial) processes and their coherence and the specification or prediction of the properties of the physical objects in which the processes should take place and that are transported from one partial process to the other.* Usually, these physical objects are streams, such as liquid streams, gas streams or moving electrons, whereas the physical objects that will have a role as performers or enablers are not yet specified. In other words, a functional design is just an ordinary physical design of a process that occurs or should occur in the physical objects that are subjected to that process. Therefore, it would be clearer to use the term '(control) *process design*' (being a design of a process), rather than the term 'functional design'.

This would be in line with the terminology in a related field of technology, being process engineering, a part of physical or chemical engineering. In that discipline the design of chemical plants always starts with the 'process design'. Such a process design is documented mainly in 'process flow diagrams' and 'material balances'. On those diagrams, the lines represent the streams of fluids (being fluids that are transported from one sub-process to the next one) and other symbols that represent the transformation of or the processes in those physical streams of fluids. In that phase in theory no choice is made which physical object will be the enabler or performer of the process. Strictly speaking that is not yet needed, because a process (a 'unit operation') such as a chemical reaction in a fluid, can be described irrespective of the apparatus in which it occurs. For example, such a reaction can take place in a tube, in a vessel or in a reactor. However, in practice the apparatus in which the process takes place does have an influence on the proceeding of the process. For example, heat transfer usually cannot be calculated without knowing the size of the heat transfer area. Therefore, also during process design often some properties of the (imaginary) enablers or performers are already specified. This is not a problem when the design is expressed in Gellish, because Gellish allows to allocate properties to physical objects, even if those physical objects are not yet classified by a subtype of 'physical object'.

In actual practice a lot of problems occur because of the fact that no clear distinction is made between the processes and the processors, being the physical objects that enable or perform the processes. Neither is often distinguished between a material stream or signal and the role that is fulfilled by such a stream or signal in the various processes.
An example of confusion is when the term 'input' is used as classifier of a stream or signal (instead of a classifier of its role), because that same stream or signal has a role as output in the process where it was created. Therefore, the stream is not an input, but has a role as input. Similarly, the term 'feedback' is an example of a role that is played by a signal (the term signal refers in this context to a physical modulated wave, irrespective of the role of that wave).

This points to the general issue, that in practice often no clear distinction is made between the nature of something and the role that is (temporarily) played by that thing. For a high quality ontology and for proper modeling this distinction is however of great importance. Neglecting the distinction makes it unclear whether a kind of physical object is meant or that there are still degrees of freedom to select a kind of physical object that can play the role in question.

3.3.8.3 The term function used as a reference to a totality in a role

The term 'function' with the second meaning, being a totality that plays a role, thus a role player, also often occurs.

For example, Peter Kroes (1996) states in his theory of the dual nature of artifacts: "technical artifacts are social constructs and at the same time physical objects. Each technical function… is a social construct, but in order to become a real artifact it has to be realized, in other words, it requires a physical 'bearer'."[39]
With this statement Kroes indicates that each technical artifact is made with the intention to have a particular function. Such a function does not need to be primarily a technical or economic function, but can also be an esthetic function. In Kroes' view that function is a central concept when he qualifies technical artifacts primarily (in their essence) as social constructs.

I agree with Kroes that technical artifacts cannot be isolated from their functions, especially not from their intended roles or purposes for which they are created. However, I am of the opinion that those functions need to be distinguished from the totalities themselves. Functions are aspects of the totalities and those aspects are possessed by those totalities. So, in my opinion, a totality and in particular a technical artifact, is not synonymous with the (social) function that it fulfills or is intended to fulfill. The artifact has a physical nature and its function has a role nature, being a role towards a social entity.
In the Gellish ontology Kroes's concept is represented by two different concepts: (1) a technical artifact is a subtype of physical object, which is a subtype of totality and (2) a social construct is a role (that can be played by the artifact towards a social entity). The ontology also makes a distinction between the intended role for which the artifact is or will be made and the actual role that is played or will be played.

The concept which Kroes refers to as 'function' seems to coincide with the concept called 'functor' as used by Hart (1984), although Kroes limits his use of the concept to technical artifacts in a social context. Hart summarizes his integral ontology with the statement that the world consists of 'functors functioning in relation'. With that expression he intents to say that each totality only exists in interaction with others in his environment. A totality in isolation, without environment, in other words a totality that is not a functor, does not exist in this world. I agree with Hart that every totality appears to be in interaction with others. But I am of the opinion that the term functor is not an adequate term to refer to the totality, because the term functor suggests that the concept that is named with the term is a specialization of 'role', whereas it is important to distinguish between a totality itself with its own

[39] That an artifact is a (or has a role as a) social entity also holds during the design phase, when the 'bearer' is not yet 'materialized' as is the case after fabrication, because an imaginary artifact is an imaginary social entity as well as an imaginary physical object.

nature and the roles that it plays in the interaction with others. This is strengthened by the fact that a totality has always more than one role, whereas the number of roles vary in time.

My conclusion is that the term function as an indication of a totality in a role is a 'shortcut' for a reference to two underlying concepts that are related by a 'possession of role' relation. In a precise ontology these concepts should be distinguished and therefore modeled as two separate concepts.

3.3.8.4 The term function used to indicate a subtype of role

The term function with the third meaning, being a subtype of role, seems to me the most natural meaning of the term. When we talk about the function of something, then usually we do not primarily point to what something is, but we point to the role that is fulfilled in a particular occurrence. A totality is not a function, but can have or fulfill a function. Furthermore it can be the case that, although a totality does not have a particular function, it nevertheless has or should have the ability to act in that role, in other words: it potentially has the function. So, there are two kinds of relations between a totality and a function: (1) a totality has a function and (2) a totality has a capability to have a function.
The term function is usually not used to refer to a role in a relation that expresses an ordinary (static) fact. Therefore, in the Gellish ontology we use the term function in this context as an indication of a role in an occurrence, as opposed to a role in a relation that expresses an ordinary (static) fact.

A complete occurrence (or an activity) is always an interaction of a number of participators, each of which fulfills its own role: performer, subjected, tool, catalyst, etc. In an accurate model it is important that always a clear distinction is made between the qualification of the own nature and the qualification of the own role of each of the participating totalities. Their nature is internally oriented, according to the own essence of what the totality is, whereas its role is externally oriented, according to its contribution in the interaction between the participating totalities.

In natural language, it often occurs that technical artifacts are classified as being of a class that is named after the *intended kind of role*, in other words, a class that can classify physical objects but that has a name of a kind of role for which the artifact is intended (and is made suitable).

> Note: Also non-artifacts can have an intended role (or function). For example, an eye has an (apparently) intended role of 'viewer' in a process of seeing. Nevertheless non-artifacts are rarely classified by a class that has a name that indicates a kind of role.

That intended kind of role is essential for a technical artifact. It determines the requirements for its characteristics. Because of this role it is what it is and it possesses the aspects that are essential for the fulfillment of that role.
In addition to that, it is possible that a technical artifact has an *actual role*, which may or may not be in line with (classified the same as) its intended kind of role. In those cases where such a technical artifact fulfills an actual role that is classified by the same class as the intended kind of role for which it is designed, confusion may arise, because in natural language often there are no separate words to distinguish between a kind of technical artifact and its intended kind of role.

This double use of the same term (being a use as homonyms) is illustrated in Figure 46.

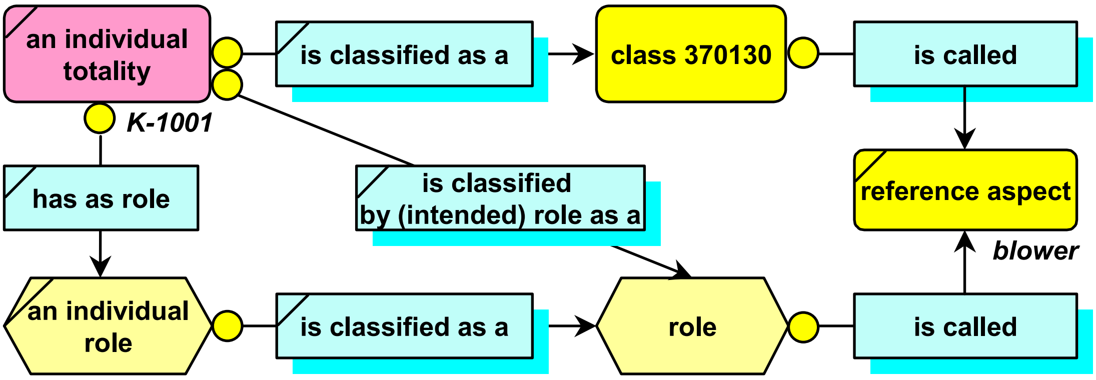


Figure 46, Classification of an artifact and of its role by different classes with the same name (a homonym)

The individual totality in the above figure is classified by its nature as being of a class that is a subtype of artifact (such as 'blower'), whereas its role is classified by the nature of the role as being of a different class that is a subtype of role (also called 'blower'). In addition to that the figure illustrates a 'cross classification', which expresses that the totality is 'classified by its intended kind of role' by the same class as is used to classify the actual role. These two different classes are named with the same name. For example, if pump P-101 functions as pump, then it *is* a pump and at the same time it *has* (or will have or had) a role as pump (the artificial term 'pumper' might have been more precise, but does not exist in English) and thirdly it may be expressed that it *has* an intended or possible role as a pump ('pumper'). The term pump that classifies the nature of the artifact *is copied from* the name of a role. This copied name is now also the name for a class that is used to classify totalities. The original use of the term pump to classify a role occurs for example, in an expression that is used to clarify the role of a human hart by stating that a hart is a (or has a role as a) pump.
Most technical artifacts are especially designed to function in a particular kind of role. Therefore, it is often the case that technical artifacts are qualified by their *intended* role.
Linguistically, a term can originally refer to a kind of role, but such a term can also be the name of a kind of physical object. It would have been simpler for an artificial language that is based on natural language (such as Gellish), if the natural language would consequently use different terms for artifacts and their intended roles (such as 'pumper' next to 'pump').

When a term that is derived from a role is used as a name of a kind of physical object, such as is the case with the term pump, this does not imply that the definition of that kind of physical object only specifies the role. On the contrary, the definition of such a concept implies all aspects that are required to fulfill the role. For example, in the definition of the concept pump, it is included that it is made of solid material, that it has a mechanism to push fluid and that its shape has certain constraints, etc. On the other hand, the other (role) concept is defined by a contribution to a process only, irrespective of any material aspects.

So, we can conclude that in the ontology we have to distinguish two concepts: physical object (in particular its subtype: technical artifact) and role or function of or for a physical object. These distinct concepts may or may not be named with the same name.
Thus the term pump in English is a homonym.

In actual practice of modeling the issue that is described above is often not a real problem, because the role may remain implicit or the role may be classified by a more generic kind of artifact, such as 'performer'. For example, if we classify a certain role of a pump as performer in a particular process, then for a human being it is usually clear from the classification of the process as a pumping process, that the 'pumper' role is meant.
However, there are cases that the distinction of the two concepts is essential for a precise expression of the meaning. This is especially the case when there exist dedicated artifacts that are especially made for a particular kind of role, whereas at the same time there are general-purpose artifacts that among others are also suitable to be used for that particular kind of role. For example, dedicated controllers as

well as general purpose controllers are on the market. The first kind of artifact is only suitable as controller of one kind of parameter, for example level or pressure. The concept ‘level controller’ or ‘pressure controller’ can therefore be the name of a particular kind of physical object. But that same name can be the name of a kind of role that can be used to classify a role that can be played by a general-purpose controller.

Finally it should be noted that objects have roles in time. This means that an individual physical object is planned to have a future role or roles (such an ‘intended role’ has to be distinguished from the intended kind of role mentioned before), whereas it actually has another role and in the past it had again a different role. These roles may be imaginary roles, played by imaginary physical objects, for example when they are part of a design, or they may be real roles of actual or historic materialized (fabricated) physical objects.

One role can be played by several objects together, for example by two pumps operated in parallel, or by several persons that cooperate to lift an item. In those cases each object has its own partial role, being its individual contribution, whereas together they fulfill the complete role. This illustrates that roles can be decomposed into partial roles (this is discussed further in Functional decomposition – decomposition of functions (occurrences).

The above is reflected in the Gellish ontology, by the inclusion of a hierarchy of kinds of totalities and a different hierarchy of kinds of roles and by allowing that they may have identical names (i.e. that their names are homonyms).

3.3.8.5 The term function used to indicate an expression of a correlation

The term function is also used to indicate a correlation between aspects of totalities. In this paragraph we explore the nature and kind of correlations and thus their position in the ontology and the way in which correlations should be modeled in Gellish.
Correlations between aspects of totalities also deal with coherence between things. Often it regards a relation between properties of the same thing, but a correlation may include aspects of different things.

In general, *a correlation is the fact that two or more aspects appear to be related in such a way that the magnitude of one aspect cannot change without a corresponding change of the magnitude of the other aspect(s).*
Most correlations are expressions of a *physical law*. But sometimes, human beings have discovered derived aspects or properties that are defined by the correlation between other aspects. For example, sometimes it appears that a ratio between two aspects is of special interest, because of its behavior, such as being constant under specific circumstances. In such cases it appears to be attractive to define a derived aspect, being the ratio between the observed aspects. For example, a density is defined as the inverse of the specific volume of a physical object, which means that it is defined as its mass per unit of its volume. Because of this definition, the density indicates a correlation between the mass of a thing and the volume of the same thing. This correlation therefore is a subtype of (physical) function, which is a subtype of relation. In other words, the density is a function of the mass and the volume of the physical object; expressed in formula: $d = f(m,V)$ and when we make the kind of correlation explicit this becomes $d = m/V$ or $d – m/V = 0$. Functions or correlations of similar kind are relations between aspects (or properties), of which at least one property is a concept that is derived by human beings.
Other functions or correlations do not relate derived aspects, but they relate aspects that are dependent on each other according to fixed patterns, which we call physical laws. So, a physical law is a *kind of* correlation or function that classifies how particular aspects relate. Such a law prescribes how the magnitude of aspects of a particular kind relate. The physical object, and thus its aspects, are subjected to the physical law. Therefore, in Gellish those aspects are defined to have a role as ‘correlated’ in the correlation, whereas the physical object that possesses the correlated aspects have a role as ‘subject’ towards the correlation. This is in line with the Philosophy of the Cosmonomic Idea (Dooyeweerd), in which the subjected physical object is said to have the role of ‘subject’ in the law.

Many correlations describe a situation or state independent of time. They assume a static situation that is often called an equilibrium, which is either a theoretical state or an approximation of reality.
Nevertheless, they are sometimes said to describe the ‘behavior’ of a physical object. If the correlation

includes time, it is an expression of a (real or imaginary) occurrence, but if it does not include time it nevertheless may describe behavior, because of a general physical law that states that ‘systems’ tend towards equilibrium. The difference between the actual state and the equilibrium is the driving force for the change towards the equilibrium. This tendency can be so strong, thus requiring only a small deviation, that the deviation between the actual state and the equilibrium can be neglected. This means that by approximation the system can be assumed to remain always in accordance with the physical law, which is actually only valid during equilibrium. Therefore, because of the change in one property of a ‘system’ (a physical object) the law describes how other properties will change in magnitude. But it neglects that this takes an unspecified amount of time.
An example of such a physical law is a correlation that describes by approximation the changes of a thermodynamic state of a particular quantity of material while remaining in thermodynamic equilibrium. Such a law can be expressed (or approximated) as a relation (or correlation) between the pressure (P), (molal) volume (V) and temperature (T), which in formula form can be described by the equation (or function) $P.V - Z.R.T = 0$, in which R is the universal gas constant and Z is called the compressibility. This compressibility is determined by the atomic composition of the material. The form of the correlation illustrates that it is not relevant which property is the cause of a certain change and which properties are the effect.

> Note: When a quantity of material is in a particular (thermodynamic) state and that state can be described by a collection of coherent properties, such as P, V and T, then that state is sometimes called a PVT-point (the other two are considered to be constants). The atomic composition and structure of the material implies that those properties are coherent in such a way, that a change of one property causes that the other properties will change as well, approximately according to the above correlation. Therefore the correlation is also called an ‘equation of state’. There is no reason to model this ‘point’ to assume the existence of a ‘compound property’ that would represent the collection of the three properties, because the ‘point’ is not a particular property, but actually it is one particular state of a continuous realm of states, at which each of the properties have a particular value.[40]

The kind of relation between the aspects that are related ‘because of a definition’ (such as density, mass and volume are related because of the definition of density as a quotient of mass and volume) as well as those that are related ‘because of (an approximation of) a physical law’ should indicate that the aspects are correlated. Therefore the term *correlation* is a suitable type of relation to refer to both types of correlation between aspects.

Finally we should distinguish between a function as defined from a physical perspective (a correlation) and a function as defined from a mathematical perspective. A *physical function* describes a *correlation* between physical properties that are (usually) not dimensionless. Mankind has discovered the existence of common correlations between objects of the same kind. Those generalized correlations are called physical laws or generally valid correlations between kinds of objects. These laws or correlations do not belong to the domain of mathematics. These generally applicable correlations are derived from observation of many individual physical correlations between (individual) aspects of individual objects. Also these correlation between quantitative aspects of *individual* objects, such as the correlation between a quantity of material and the space it takes, is usually not regarded as belonging to the domain of mathematics.

3.3.8.6 The term function used to indicate a mathematical equation

A *mathematical function* is a relation between mathematical objects that are by definition dimensionless and are ‘universals’. A physical function and an equivalent mathematical function differ fundamentally, because of the fact that the first one is a correlation between physical phenomena and the second one is a relation between numbers. In other words, the latter is a depicture of numbers (or

---

[40] Often, a correlation between two properties is represented by a curve. A point on such a curve then suggests to represent a compound property. But physically such a curve represents a (continuous) correlation between the pair of properties, of which continuously a pair of values are valid at the same time. Mathematically, it is common practice to represent a point on a curve by a pair of values on the axes that are depicted on each other, for example as (x_1, y_1). However, from a physical perspective there is no third, compound property, because a state is not a property, but is described by properties that are dependent on each other.

spaces) on other numbers (or spaces). The term *equation* is more suitable for the *mathematical expression* in the form of a mathematical formula. Such a mathematical equation can correspond to a physical correlation, provided that a defined collection of units of measure is used to map the physical phenomenon to numbers. For unambiguous computer interpretation it is important that a clear distinction is made between a physical correlation and its mathematical expression in the form of an equation, although both may be referred to as a function. In natural language the context usually provides sufficient information to know which of the two meanings is meant and there is little confusion when the term function is used as a name for both concepts that are described here, however in the Gellish ontology they are clearly distinguished as relations between (physical) properties and as relations between mathematical variables (or 'mathematical spaces').

Furthermore, there is another related category of things, which are mathematical operations. A mathematical operation is act (an occurrence) with mathematical objects as inputs that generates numeric outputs, and that follows a procedure, being an algorithm (which is a qualified kind of occurrence). Such an algorithm has a large similarity with a function, but the algorithm prescribes a sequence of assignments, whereas a function only states what is the case. So a mathematical operation happens and is therefore dynamic, whereas a mathematical function is static.

The relation between physical correlations, mathematical functions and mathematical operations is illustrated in Figure 47.

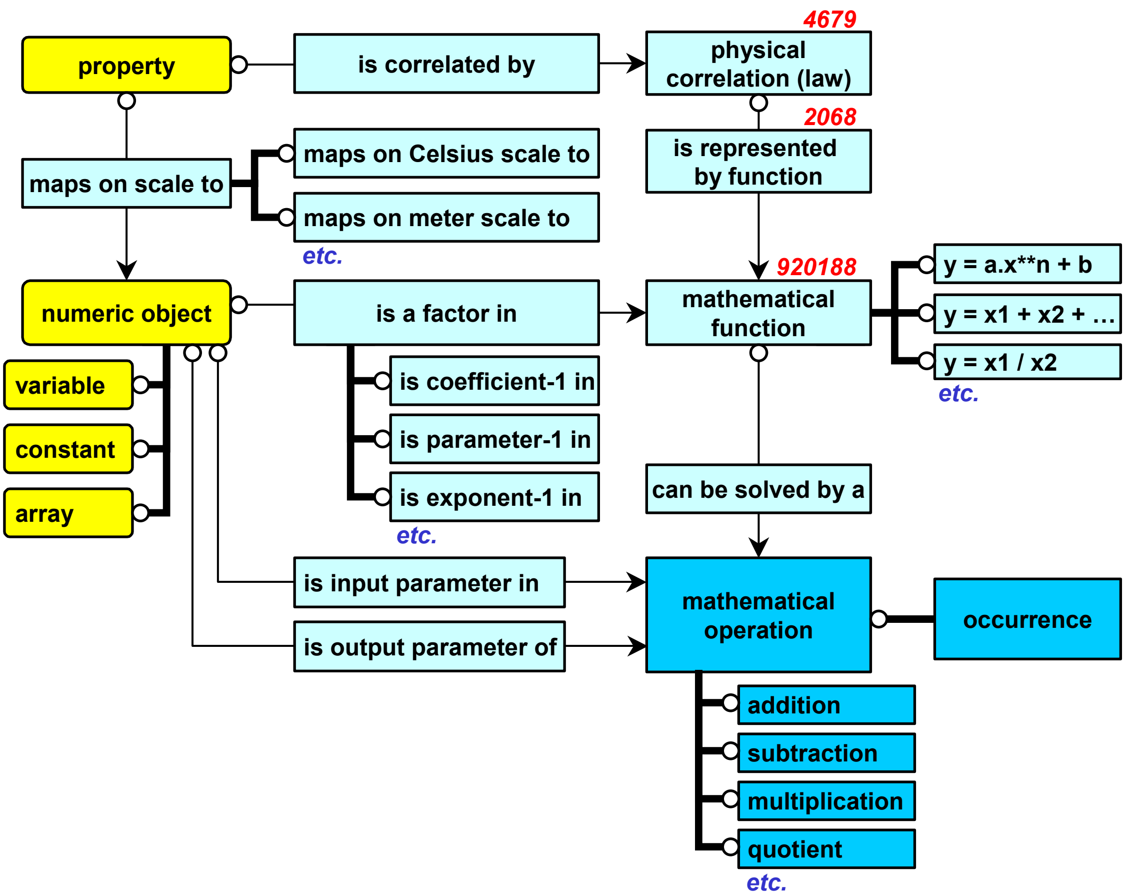


Figure 47, Mapping between physical correlations and mathematical functions

A physical correlation between properties can be described by a corresponding mathematical function between numeric objects provided that each correlated physical property is mapped to a dimensionless numeric object, using a particular scale.
A *scale* is a mapping convention to represent the magnitude of a physical property on a numerical value. This implies that a scale is a relation type between a property and a mathematical space, usually a number. This is discussed in more detail in section 3.3.9.5.

3.3.8.7 Functional decomposition – decomposition of functions (occurrences)

Given the above analysis of the various concepts that are referred to by the term function, we can now clarify what 'functional decomposition' means.

Functional decomposition is a technique that is often recommended as a first step in methodologies for the design of technical artifacts. The idea is that the overall intended 'function' of the intended complete artifact has to be decomposed. When it is unclear which of the above described concepts should be decomposed, there is a risk that the decomposition becomes a mixture of artifact decomposition, role decomposition and occurrence decomposition. For example, it can be the case that roles are decomposed, without identifying and decomposing the activities and processes in which those roles are required. It may also result in a decomposition of a role, on the basis of the decomposition of the artifact that fulfills the role (as is the case in the 'hamburger model'), while ignoring the other objects that participate in the activity or process.
A proper functional decomposition should in my opinion be primarily a decomposition of (intended) occurrences (processes, activities and events), by decomposing what should happen. This should be followed by an identification of the roles that are required in those occurrences and partial occurrences. This means that the decomposition of occurrences is in line with the roles that are played. However, the roles follow from the occurrences and not the other way around.
The essence of a functional design phase is that, together with the occurrences, only the input, output and control role players need to be defined, and that the players of performer or enabler roles are largely ignored.
This means that a methodology of functional decomposition should result in an occurrence decomposition as well as a role 'decomposition' with the relations between them.
The term 'role decomposition' seems to be slightly inaccurate. Roles are combined, but not assembled. A combination of roles consists of a number of roles, each of which is performed by a separate role player. For example, when two machines perform two roles at the same time such that one process takes place, then the roles together are not one composed role that is decomposed into partial roles, but the roles together remains a collection of roles that are performed in parallel. This illustrates that we should not talk about decomposition of roles, but about collection or combination of roles.

In the Gellish language, the concept of composition, being a supertype of assembly and collection and combination, is defined at a high level in the hierarchy, so that it includes (de)composition of occurrences, as well as the combination of roles and the assembly of totalities that fulfill the roles.

3.3.9 The ontology of kinds of relations

In the above paragraphs it is described how the facts in reality and imagination can be expressed by a combination of atomic relations, that themselves are composed of a structure of elementary relations.

A further analysis of the kinds of relations revealed that also kinds of relations can be structured in a specialization/generalization hierarchy. This means that semantic definitions of the supertype relations are inherited by the subtype relations. For example, subtype relations inherit the roles that are defined to be required by their supertype relations. Furthermore, it appeared that the number of kinds of elementary and atomic relations that are required and used in a technical and business practice is a collection of a limited size. This hierarchical set forms a semantic system of kinds of relations. It has some analogy with the 'periodic system of elements' in physics.

Figure 48 presents an overview of the Gellish upper ontology of kinds of relations. Other figures in this document provide an overview of further subtypes of these kinds of relations. Details of the definitions of the kinds of relations are provided in the Gellish dictionary / taxonomy of which a summary is presented in Appendix B. Those details include also the kinds of roles that are required by those relations and the kinds of objects that can play roles of those kinds.

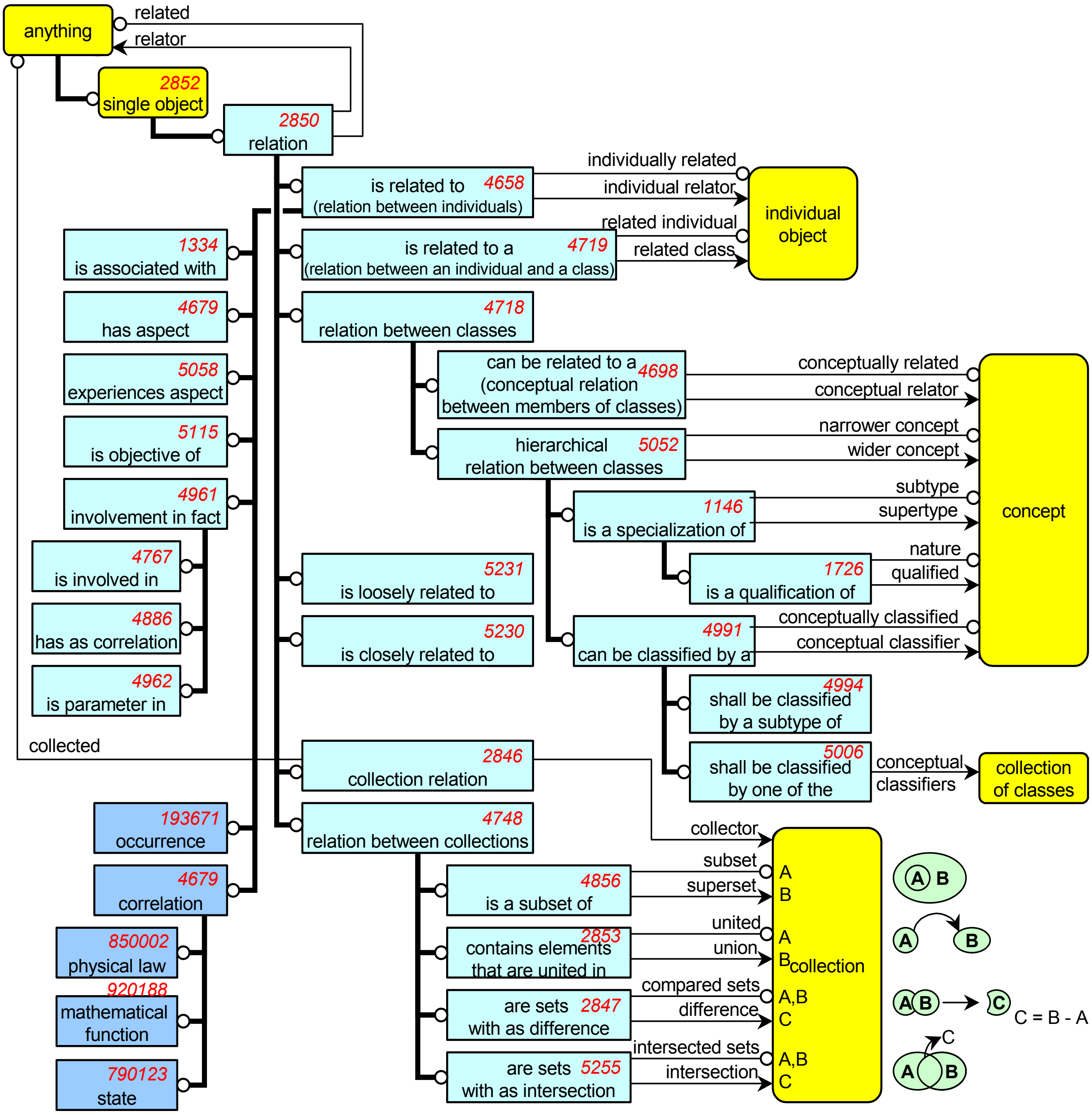


Figure 48, Upper ontology of kinds of relations

Figure 48 illustrates that relations can be of different kind, depending on what kind of fact they represent. The following main categories are distinguished:

- **relations that represent individual facts** about the relation between individual things.
- **relations that represent generic information** about individual things. These relations relate an individual object to a kind of object. For example, a classification of an individual object, such as P-1 is classified as a pump, relates an individual object to a kind of object. Another example of a relation between an individual thing and a kind of thing is the fact that a particular person can execute activities of a kind, such as ‘John can paint’.

- **relations that represent knowledge about members of classes**. This knowledge is generic and valid for objects of particular kinds. For example, the fact that a wheel can be part of a car.
- **relations that represent generic knowledge about relations between kinds of objects**. For example, a specialization relation (or subtype/supertype relation) represents knowledge about a kind of thing in relation to another kind of thing.

Each of these kinds of relations and their further subtypes is a concept that has a unique identifier (UID) that is sufficient to refer to the concept. However, for human readability, each concept has not only a name in English, but it has also at least one phrase that can be used in Gellish sentences. It has also at least one inverse phrase in case the sentences are written and read in the inverse direction. It can also have one or more synonym names and names and phrases in other languages.

Figure 48 presents only the top part of the hierarchy of kinds of relations. A certain status of the full hierarchy of kinds of relations is described in Appendix B, Upper ontology of relations . The current full set is published regularly as new versions of the ‘open source’ Gellish language definition table. The kinds of relations in that Gellish Table form the standardized grammar that defines the Gellish language and are an integral part of the Gellish dictionary / ontology. That hierarchy of kinds of relations appears to be reasonably complete for the expression of facts and knowledge, including those about physical and business processes at least in an engineering, procurement and facility operation and maintenance context. Furthermore, the Gellish language is ‘alive’, which means that additional concepts can be added and (synonym) names can be added as and when required. However, concepts will not be deleted, they can only be ‘pensioned’ and succeeded by a different and better-defined concept. So the UID’s cannot change and a successor name or concept is accompanied by a date, so that the language history of Gellish is defined and old expressions remain interpretable. This makes that Gellish is a stable and time resistant language.

The above-mentioned main direct subtypes of ‘relation’ are discussed in further detail in the next sections.

3.3.9.1 Relations between individual things

The first subtype of relation is a relation between individual things, which expresses *that* individual things are related, whereas its further subtypes express *how* they are related. In natural languages these kinds of relations are expressed with phrases that make clear that individual things are related. For example, ‘A is connected to B’ expresses a connection relation between the individual A and the individual B. This is different to an expression such as ‘an A can be connected to a B’, in which A and B are apparently kinds of things. This difference in semantics is often not made in information modeling, where it is common practice that relations between kinds of things are modeled and where ‘instances’ of those relations are assumed to have the same semantic meaning. In the Gellish ontology however, there are two different partial hierarchies for these different kinds of relations. The Gellish names of the relation types that relate individual things usually start with ‘is’ or ‘has’. For example, ‘is a part of’ or ‘is a performer of’ or ‘has aspect’.

As is described in section 3.3.3, there are two kinds of ‘elementary relations’ that express two kinds of facts. For a relation between individual things these two relation types are:

1. Having a role: which expresses the fact that something *plays* a particular role.
2. Implying a role: which expresses the fact that a relation *requires* a role.

These are subtypes of the kind of relation ‘relation between individuals’.

A number of other subtypes of relation between individual things (individual objects) are shown in Figure 49 and Figure 50. Those subtypes are arranged in a subtype / supertype hierarchy, but to reduce complexity of these figures that hierarchy is not shown here, but is presented in Appendix B.

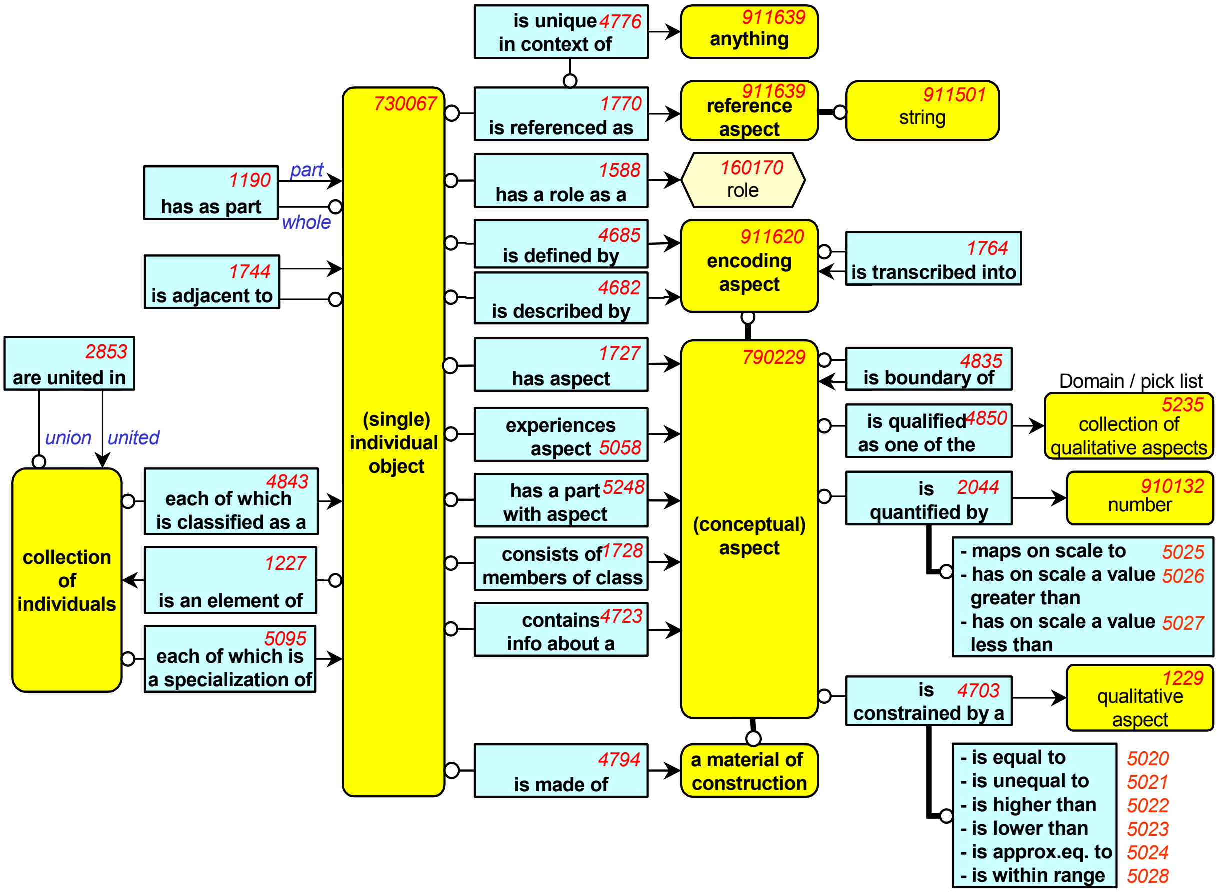


Figure 49, Relations between individuals: Aspects of an individual or of a collection

Figure 49 illustrates that relations of a particular kind (by definition) relate things of a particular kind or relate things that are classified by subtypes of those kinds of things. That information is often sufficient for an intuitive understanding of the semantic meaning of the kinds of relations. Therefore, the definitions of all these kinds of relations and their hierarchy are not discussed here. A more detailed definition of these Gellish ontology concepts is documented in the Gellish database and is summarized in Appendix B. In that database, each kind of relation has a textual definition, which extends what it inherits from its supertype kind of relation by adding constraints that distinguishes the subtype from the supertype and from its 'sister' subtypes. In addition to that, each kind of relation is further defined by the kinds of roles that it requires and by the kinds of objects that can play the roles of those kinds, whereas those kinds of roles and role players are defined in a similar way (by text and supertypes and other relations). That further definition describes the semantic constraints that are applicable for the kind of relation. For example, the kind of relation with the name 'assembly relation' and the phrase name 'is a part of' has as roles 'part' and 'whole', whereas the players of both those roles are members of the kind individual object. In other words, the Gellish expression: 'A is a part of B' implies that A as well as B is classified as an individual thing or as a subtype of that.

Figure 50 as well as Figure 49 illustrates that some subtypes of relations between individuals can be further specialized to define in a semantically more precise way which kind of relation is meant. For those subtypes it is not explicitly indicated in Figure 50 which kinds of things are related, as they inherit that from their supertype kind of relation.

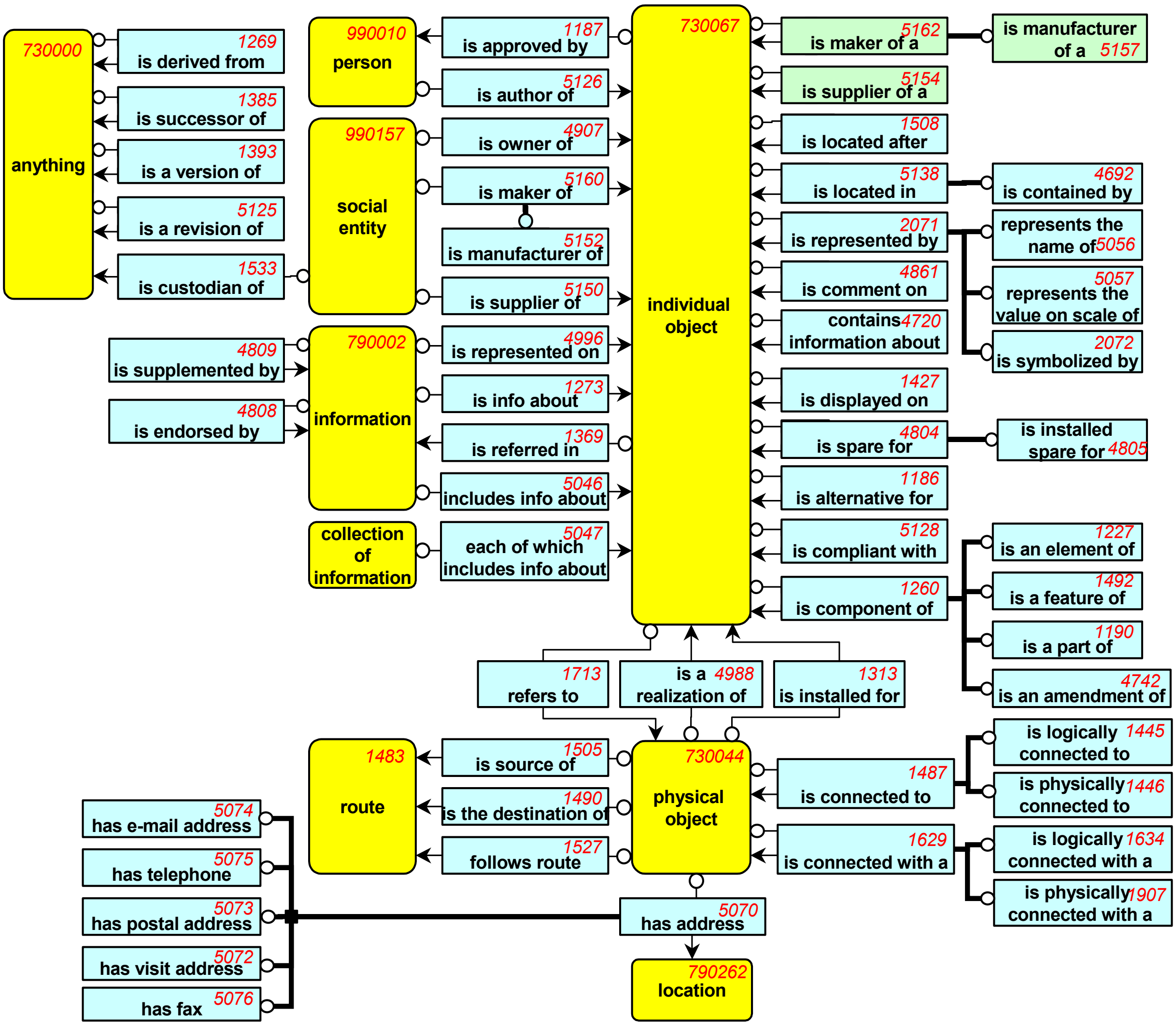


Figure 50, Relations between individuals - continued

As said above, the hierarchy of these, and other, relations between individual things is illustrated in Appendix B, Upper ontology of relations . That hierarchy includes for example two main ‘branches’, which indicate two series of subtypes of relations between individual things:

- is associated with (association relations)
- has aspect (possession of aspect relations)

The first kind classifies relations between a totality and other totalities in its environment; the external relations.
The second kind classifies relations between a totality and its intrinsic aspects.

Further specializations of these kinds of atomic relations are only useful when further constraints are applicable for the subtypes. For example, the connection association (‘is connected to’) has further subtypes depending on the kind of connection: loose connection, fixed connection, electricity conducting connection, etc. In other cases the kind of related things makes sufficiently clear what kind of relation is meant. For example, the expression that a physical feature ‘is a part of’ a totality is semantically sufficiently defined, although the subtype relation phrase ‘is a feature of’ could be used, but is unnecessary.

3.3.9.2 Relations between kinds of things – expression of knowledge

This section discusses the second part of the total semantic system, which includes *relations between kinds of things*. Relations of this kind express knowledge as will be illustrated below.

We defined a fact as 'that which is the case'. Similarly we can define that a *kind of fact* is *'that in which facts have resemblance'*, in other words it defines a *commonality* of facts. Such a kind of fact is expressed in Gellish as a relation between kinds of things, which specifies that members of those kinds of things have relationships of that kind in common.[41]

3.3.9.2.1 Conceptual relations

A category of kinds of relations between kinds of things expresses *possible facts* between individual things of those kinds. These are called *conceptual relations* (between members of classes) that express that members of a kind of thing *can have a relation of this kind with* members of the other kind of thing. A conceptual relation is therefore an expression of possible relations between individual objects of the related kinds.

If *each* individual thing of a particular kind *has* something that has a commonality with what the other members of the kind have, then there is apparently a common fact for all members of the kind, which is expressed as a *common relation*. Such a common relation is a subtype of a conceptual relation with a cardinality constraint that is greater than zero. Such a relation expresses that something is *by definition* the case for members of the kind.
Common relations often express not only a commonality in nature, but also a commonality in magnitude or intensity. Then they can be called *common qualitative relations,* because there is also commonality on the related qualitative aspects, or on their subtype quantitative aspects (values). An example of a conceptual fact is the fact that each quantity of matter has a mass. This is not a particular individual fact (about an individual thing), but it is a kind of fact. In other words it is a commonality between individual facts. The fact can be expressed as a relation between a kind of totality ('quantity of matter') and a conceptual aspect ('mass'). The fact as such does not imply the for example my car has a particular mass, but only that 'my car' has an aspect that can be classified as mass, provided that my car is an individual totality that is classified as a (subtype of) quantity of matter. An example of a common qualitative fact is that human beings have the same average body temperature of 37°C or more precisely said: each human being has an average (individual) body temperature that is classified as 37°C. This is a relation between the class human being and the qualitative class 37°C (within a tolerance around it), which expresses that human beings not only have a commonality that they have an average temperature, but they have also a commonality in the quantitative value of that temperature.

Conceptual relations as well as common relations may have additional constraints, such as on their validity context or duration or number of occurrences (the cardinality constraints). An example of possibilities for relations is the following: consider the fact that a car *can have* a number of wheels that can vary between 3 and 5 (including the reserve wheel). This fact does not define how many wheels are part of a car. It only defines a general truth about *well-formed* cars in normal situations. These additional constraints can also be context dependent. This is usually the case when such a relation expresses a requirement that in a particular context a relation *shall be* the case. These 'shall be' relations are subtypes of the conceptual 'can be' relations.

Conceptual and common relations are not *true* relations between kinds of things, because the kinds don't have a relation; but such a relation indirectly expresses facts about relations between members of the kinds. Their semantic meaning is that an individual thing of the first kind has a relation of the indicated kind with an individual thing of the second kind.

In summary, there are *conceptual relations* that express possible relations between the members of the related concepts, there are conceptual requirements that express what kind of relations shall be present for members of a kind, and there are *common relations* that express that all the members of the related

---

[41] The explicit kinds of facts about a kind of thing together *implicitly* define 'that in which things are or may be different'.

concepts have at least one relation of the kind. The latter have a minimum simultaneous cardinality that is greater than zero.
So, *a kind of fact about a kind of thing is that which in general is or can be the case for members of the kind of thing.*

3.3.9.2.2 Hierarchical relations

Another category of relations between kinds of things are non-recursive relations. Such relation types form a hierarchy of concepts, such that there cannot be a relation between the same kinds. Such a relation type is really between kinds of things and does not express common facts about members of the kinds. An example of such a relation type is the *specialization relation*.
A specialization hierarchy is a hierarchy that relates subtype to supertype kinds of things or concepts. Therefore each subtype and supertype role in such a relation can (only) be played by a kind of thing (and not by an individual thing). This constraint is expressed in the definition of the specialization relation. This means that the ontology, with as backbone a specialization hierarchy of kinds of things, can only present the right hand 'world' of Figure 26.
A specialization relation implies that a subtype concept inherits all constraints that are defined for the supertype concept, including the ones that the supertype concept inherits from all its supertypes.

To clarify what is inherited by a subtype kind of thing from its supertype kinds of things up to the top of the hierarchy, it is important to distinguish between four kinds of conceptual possession of aspects (conceptual aspects conceptually possessed by kinds of totalities):

1. **Essential conceptual possession of aspects.**
   This includes kinds of aspects of which members are certainly possessed by all members of the kind of thing. Such aspects are called necessary and sufficient intrinsic or obligatory aspects, because they belong to the nature of isolated things of that kind and determine the nature of what the thing is by itself. In other words a possession relation of this kind defines that each member of the kind of totality *has by definition* an aspect of the possessed kind.

2. **Normal conceptual possession of aspects.**
   This includes kinds of aspects of which members are possessed by each *well-formed* totality of the kind of possessor. The members of the kind normally possess these aspects, but if those members don't possess such aspects, then they can still be members of the kind.
   For example, a well-formed human being has two legs, but without legs you are still a human being, although not well-formed. So, such an aspect is normal, but not essential for things to be member of the kind. Typical for normal aspects is that the individual things of such a kind normally have these aspects (by default), because normally individual things are well-formed. The norm of well-formedness may be context dependent, so that it may be the case that for members of a kind it may be specified that in particular cases their aspects are constrained by a particular requirement. For example, a standard specification or a buying description of a particular company may specify requirements for members of a particular kind that they shall have particular values for specified kinds of aspects.
   In other words, a normal conceptual possession of aspect relation defines that in a particular context each member of the kind of totality normally *shall have* aspects of the possessed kind.

3. **Optional conceptual possession of aspects.**
   This includes aspects that are sometimes or usually possessed by members of the kind of totality. These kinds of aspects are incidentally possessed or probably possessed, because there is a probability or chance that members of that kind possess aspects of that kind. Having such aspects or not having such aspects does not determine whether a thing is a member of the kind, nor does it determine whether the thing is well-formed.
   In other words, an optional conceptual possession of aspect relation defines that a member of the kind of totality may or *can have* aspects of the possessed kind. A general rule in Gellish is that such an expression does not exclude that the totality may have aspects that are not mentioned (see below).

4. **Unspecified conceptual possession of aspects.**
   This includes aspects about which no constraints are specified in the model, neither in positive nor

in negative sense. This implies that individual totalities of that kind nevertheless can have such aspects. This can be called implicit knowledge. Sometimes that knowledge is generally known, but in other cases the knowledge is unknown for most people or is of importance only in special cases.

An example of generally known knowledge is that material physical objects consist of atoms. An example of knowledge that is unknown by most people is that metal things generally have a crystalline structure. These conceptual facts may both not be expressed explicitly in the Gellish knowledge bases, whereas it is nevertheless possible that particular facts about the composition or crystalline structure of a physical object is expressed in the Gellish language.

Next to 'positive' facts of these kinds it is also possible that it is explicitly recorded that members of a particular kind of thing (normally or optionally) does *not* have aspects of a particular kind. This means that the denial of facts of the first three categories are 'negative' facts that are also inherited by subtypes of the possessor kind of totality.

3.3.9.2.3 Distinction between conceptual relations and individual relations

Relations between kinds of things may describe generally valid knowledge or knowledge that is proprietary and/or only valid within a particular context. Knowledge that is expressed in Gellish, using standard Gellish relation types that relate Gellish concepts or their subtypes form a Gellish *knowledge base*. The Gellish knowledge base is an example of an open source knowledge base that is expressed in Gellish English. Other knowledge bases may be public domain or private extension of that.

Figure 51 illustrates relations between kinds of things and how they relate to relations between individual things.

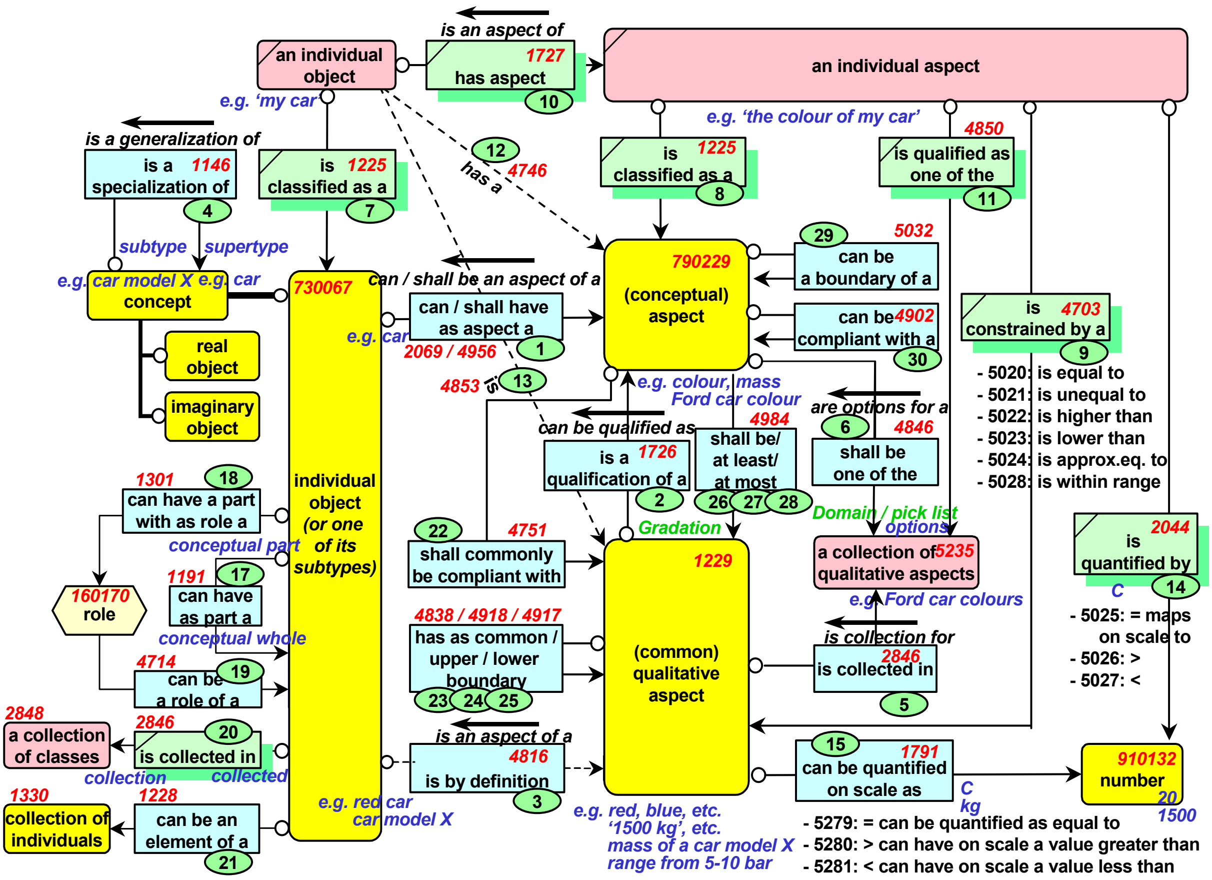


Figure 51, Relations between individual things and between kinds of things

At the top of Figure 51 it is illustrated that an individual object has an aspect. The object as well as the aspect is classified as individual object (or one of its subtypes) and aspect (or one of its subtypes) respectively. Between those kinds there is a relation which expresses the knowledge that an individual object 'can have as aspect a' aspect. Many further subtypes of the latter relation are defined in the

Gellish knowledge base. For example, the knowledge (fact) that a wall can have (and normally has) as aspect a wall thickness, that a pump can have a capacity, that a material can have a temperature, etc. Note, that all the subtypes of wall, pump, material, etc. inherit this knowledge from their supertypes!

Examples of relations that use the kinds of relations whose definition is illustrated in Figure 51 are given in the following table, in which the fact id's correspond with the numbers in the figure:

Fact id		Left hand object name	Relation type id	Relation type name	Right hand object name	UoM
1	a	car		can have as aspect a	colour	
2		red		is a	colour	
3	a	red car		is	red	
4	a	red car		is a	car	
5		red		is one of the	Ford car colours	
6	a	Ford car colour		shall be one of the	Ford car colours	
7		my car		is classified as a	red car	
8		C-1		is classified as a	colour	
9A		C-1		is qualified as	red	
9B		T-1		is unequal to	20 C	
9C		T-1		is higher than	20 C	
9D		T-1		is lower than	40 C	
9E		T-1		is approximately equal to	20 C	
9F		T-1		is within range	20 - 40 C	
10		my car		has aspect	C-1	
11		C-1		is qualified as one of the	car colours	
12		my car		has a	colour	
13		my car		is	red	
14A		T-1		is quantified by	20	C
14B		T-1		is greater than	20	C
14C		T-1		is less than	40	C
15		20 C		can be quantified as	20	C
16		my motor		is part of	my car	
17	a	car		can have as part a	motor	
18	a	car		can have a part with as role a	front wheel	
19		front wheel		can be a role of a	wheel	
20		car		is collected in	list of vehicle types	
21	a	stamp		can be an element of a	stamp collection	
22		material of model X		shall commonly	standard spec A25	
23		area A		has as common boundary	border of A	
24		20 - 40 C		has as upper boundary	40 C	
25		20 - 40 C		has as lower boundary	20 C	
26		material of model X		shall be	stainless steel	
27		volume of model X		shall be at least	10 dm3	
28		start time of model X		shall be at most	3 s	

29	a	temperature		can be a boundary of a	temperature range	
30	a	material		can be compliant with a	specification	

Note 1: In the above table the column that sometimes contains the word 'a' is added only to support the readability of this Gellish English table, because normal English sentences of those kinds require that the nouns are preceded by the word 'a'. The column is not part of a standard Gellish Table.

Note 2: Line 9B and 15 together illustrate that a quantitative property ('20 C') is a degree of hotness that can be mapped to different numbers, depending on the scale that is used. In degree Fahrenheit it would map to another number. So the string '20 C' should not be read as a number and a scale, but as a name of the degree of hotness. The right hand terms of lines 27 and 28 have a similar meaning.

Relations between kinds of things mirror the relations between individuals. This means that they can be found by replacing the words 'is' and 'has' in Figure 49 and Figure 50 by 'can be' and 'can have' respectively.

There is also a relation between each 'kind of relation between individual things' and a corresponding 'kind of relation between kinds of things'. These relations can be used to expresses that an individual relation is a realization of an earlier defined relation that expresses knowledge. However, the Gellish language allows that an individual relation (fact) is defined even if the knowledge whether that is a possible relation (fact) is not (yet) defined. For example, the Gellish language allows to express that 'my radio is a part of my car', although in the knowledge base it is not expressed that 'a radio can be a part of a car'. The knowledge that defines the semantics of the 'is a part of' relation is sufficient for the interpretation of the expression. This linguistic knowledge expresses that 'A is a part of B' is a proper linguistic expression, provided that both A and B are individual objects, irrespective of the question whether it is physically possible that A is a part of B. This is different from conventional data modelling in IT, where it is required that physically possible kinds of relations have to be defined before instances of them can be created.

3.3.9.3 Allowed number of occurrences of a kind (cardinalities)

Each of the kinds of relations between classes that express possibilities and constraints on the relations between individual things can be further distinguished on basis of the constraints on the possible or allowed number of individual things that at the same time or in the course of time can be involved in a relation of the kind. For example the fact that a wheel can be part of a car still does allow that the number of wheels that can simultaneously be part of a well formed car may vary from e.g. three to five, including a reserve wheel. But the number of wheels that can be part of the car during its whole lifetime is unlimited. The minimum and maximum number of simultaneously allowed facts of the same kind determines how many simultaneous facts participate in the definition of the constraints for the kind. The minimum and maximum *simultaneous cardinalities* and minimum and maximum *lifetime cardinalities* are expressed in the Gellish language as aspects of the roles in a fact. Each role therefore has in principle four cardinalities in the form of numbers that indicate these minima and maxima. In other words: the cardinalities indicate the minimally n_1 and maximally m_1 of a kind can have at the same time (or in the course of time) have a relation of this kind with at least n_2 and at most m_2 things of the other kind. The default for minima and maxima is zero and unlimited respectively[42].

The Gellish phrases express the possibility of a relation by the term 'can'. Especially when the minimum number of allowed simultaneous instances is zero (so it is an optional instance), then the normal English phrase also begins with the term 'can'. For example, the phrase: '*can* have as aspect a' expresses the recognized possibility for a totality of a kind to have an aspect of the indicated kind. When the minimum number is greater than zero, then at least one instance must be the case. The Gellish convention is that in such cases the phrase may also begin with the term 'can', whereas the cardinalities indicate the degrees of freedom that are still left. In Gellish English such phrases may also

[42] The life time cardinalities are usually considered to be unconstrained and are therefore ignored in a Gellish Table.

begin with ‘must’ or ‘shall’ or ‘has by definition’. Such a phrase expresses a fact that is by definition the case and expresses the requirement that a value about the aspect is required to be recorded.

Apart from generally valid knowledge, there is also knowledge of what is the case within a particular context. For example, a particular company may decide that they only produce cars with four wheels. This means that in the context of that company the product model of a car contains a similar expression as the expression of the general knowledge, but only with more stringent cardinalities.

For example:

Context	**Left hand object name**	**Relation type name**	**Right hand cardinalities**	**Right hand object name**
company X	car	can have as part a	4,4	wheel

3.3.9.4 Relations between individual things and kinds of things

A third part of the hierarchy of kinds of relations regards relations between individual things and kinds of things. It is self-evident that the ‘classification of individual thing’ relation (‘is classified as a’) is among them. The use of this relation is illustrated in the upper part of Figure 51, where the individual thing as well as the aspect is classified. There is no use in a further specialization of this classification relation, such as classification of physical object, classification of aspect, classification of activity, etc., because if the individual object is classified, for example as a pump, then from the generalization hierarchy above pump it is already defined that the object is a physical object. It would be a duplicate, and thus superfluous, if a ‘classification of physical object’ relation would be used to express this fact again. This illustrates a generic rule of the Gellish language that subtypes of kinds of relations are only useful if the more detailed semantics is not contained in the kind of objects that fulfils the roles in the relation.

Figure 51 also illustrates that in addition to the classification of the individual aspect (e.g. as a colour), the individual aspect is qualified by a qualitative kind of aspect (e.g. as red). That qualification relation has further subtypes that define whether the individual aspect is for example greater than or less than the qualified kind of aspect. For example, the expression that states: the colour of my car is qualified as red, is a qualification of an individual aspect by a qualified kind of aspect, in which red is a qualitative aspect. This qualitative aspect on its turn is a qualification of (in other words a qualitative subtype of) the concept colour. Similarly there are kinds of relations that are used to describe quantifications of individual aspects by quantification of the magnitude of an aspect by a number, usually on a scale.

Furthermore, this part of the hierarchy of kinds of relations contains kinds of relations that express that an individual object can have a relation of a particular kind with a member of a kind of object. For example, an individual object can play a role of a particular kind, such as ‘John can be a driver’, or an individual object can have a part of a particular kind, such as ‘my car can have a trailer’.

3.3.9.5 Relations about scales and quantification of qualitative aspects

The magnitude of qualitative aspects can often be mapped to a numeric space, by means of a scale. Such a scale has the nature of a kind of relation between a kind of aspect and a kind of numeric space, which typically is a kind of numeric value. A relation of that kind can be created by a procedure that transforms an observation into a number on a scale.

Figure 52 illustrates the types of relations between the involved concepts.

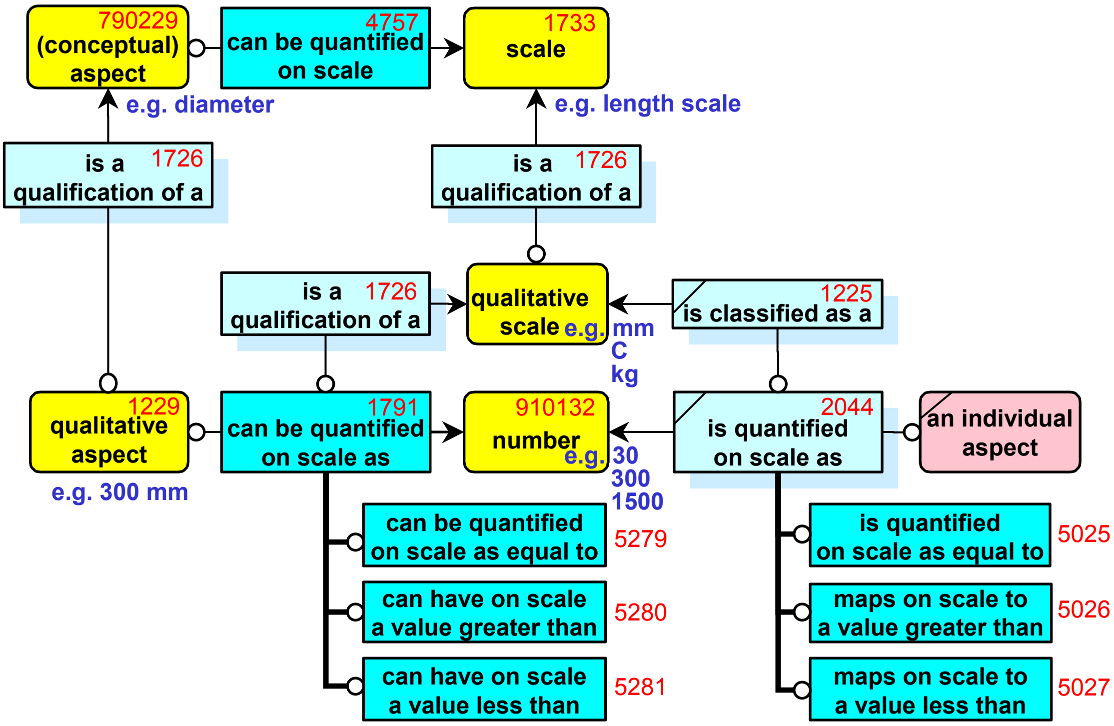


Figure 52, Relations about scales and quantification of qualitative aspects

Figure 52 illustrates that an aspect can be quantified using a scale. For example, length can be quantified using a length scale. Therefore, subtypes of scale are for example length scale, temperature scale, density scale, earth quake scale, etc., whereas qualifications of scales are the qualitative scales, which we usually call ‘units of measure’. For example, the magnitude of a temperature can be mapped on a number via the Fahrenheit or the Celsius scale. The latter scale is a mapping of the temperature of melting ice on the number 0 and of the temperature of boiling water on 100, both at atmospheric pressure. By using the expansion of a mercury column along that range, it is possible to allocate the magnitude of a temperature to a number between 0 and 100 or to its extrapolation in a way that is directly proportional with the length of the mercury column between its length at melting ice and its length at boiling water (assuming that the expansion is regular). In a similar way, all the units in the SI system, defined in ISO 31, and other (non-SI) units are qualitative scales, including derived units with or without prefix. Each of these qualitative scales are defined in the Gellish dictionary / taxonomy as a qualification of a scale. For example, m, mm, cm, dm, km, inch, light year, etc. as well as m/s, km/h, miles/h, etc. are qualitative scales. It is also recorded which scale is suitable for the mapping of which property.

Note, that the fact that a scale is defined as a relation means that a ‘unit of measure scale’ is different from the definition of a unit of measure as a (standard) reference measure, such as the length of a bar of platinum in Paris. The length of that bar is just a particular length of 1 m that is used for reference, but it is not a complete mapping method.

Figure 52 further illustrates that there is a semantic difference between the mapping of a common qualitative aspect and the mapping of an individual aspect.

The common qualitative aspect, is a kind of aspect which has an explicit magnitude. A numeric value (typically a number) is also a kind of qualitative aspect. Therefore, the relation between a qualitative aspect and a numeric value is a quantification relation between kinds of things, called ‘can be quantified on scale as’. This quantification relation is sub-typed to enable to express that the magnitude is either equal to, or greater than, or less than the numeric value. In addition to that, it needs to be indicated which scale is applied to create the quantification relation. This is expressed by relating the particular quantification relation to a quantitative scale. The nature of that relation is a qualification relation.

The quantification relation that maps the magnitude of an individual aspect to a numeric value is by nature a relation between an individual thing and a kind of thing. A particular quantification of an individual aspect is therefore also an individual relation, which has its own kinds of sub-types. Therefore, such a particular individual quantification relation is related to a qualitative scale by a classification relation.

3.3.10 Ontology of kinds of roles in kinds of relations

Each kind of relation requires its own kinds of roles to be played by involved things. A subtype of a kind of relation therefore implies that some or all of the kinds of roles of the supertype are specialized also into subtype roles of the subtype kind of relation. As a consequence, the ontology will contain a hierarchy of kinds of roles (subtypes of 'role') that is necessarily consistent with the hierarchy of kinds of relations.

Figure 53 illustrates the upper ontology of kinds of roles.

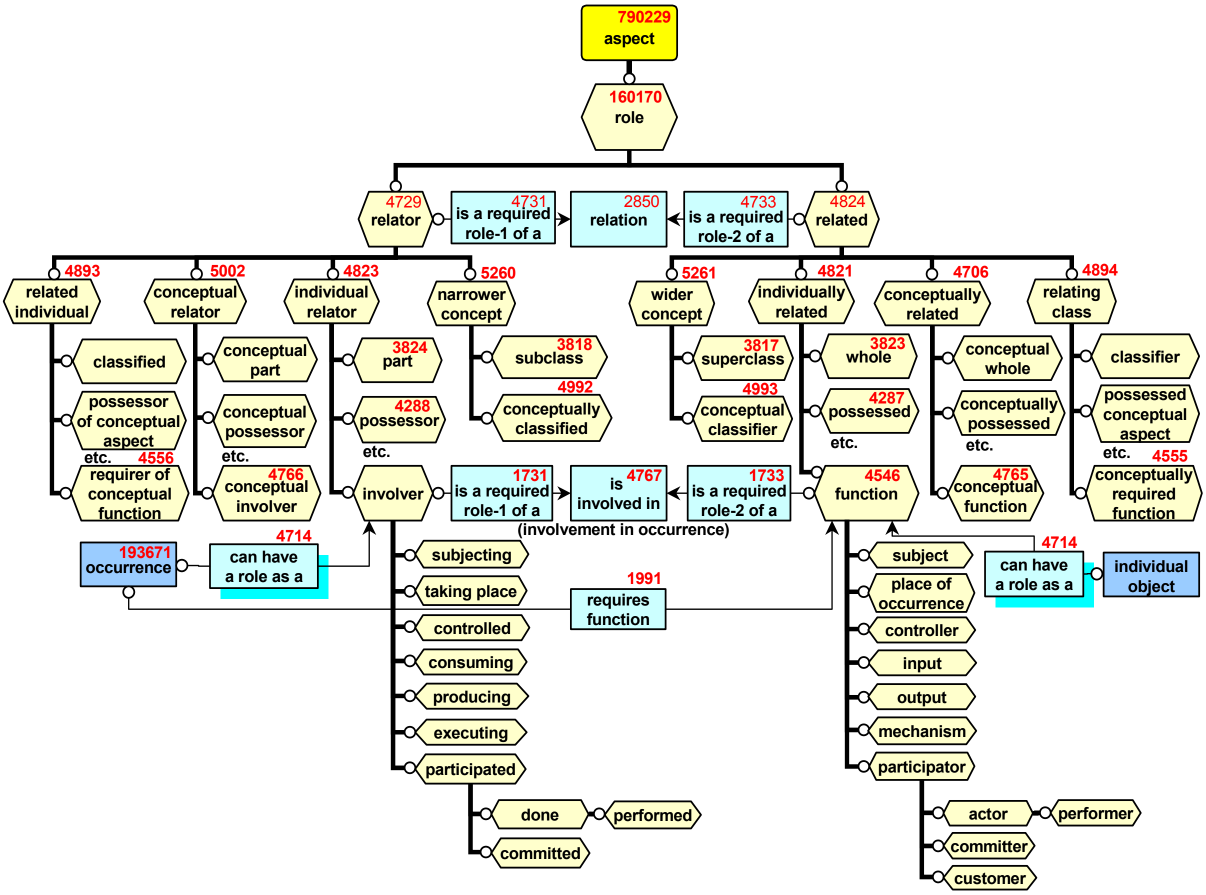


Figure 53, Upper ontology of kinds of roles

This figure shows at some places elementary relations that illustrate how the roles relate to a relation. It also shows that the concept 'role' is a subtype of aspect, and in that way it is incorporated in the total hierarchy of the integrated ontology.

3.4 Information – Implicite or explicite information

Information is an aspect of an expression. It can be further qualified on the basis of its degree of *explicitness* or *degree of expression*. This indicates whether something is *implicit information* (implied meaning) or *explicit information*, explicitly expressed (as an expression) in some way, such as in a language or as a drawing or as (an instantiation of) a model.

Physical files, books, reports, with ink on paper or magnetic 'bits' etc. are physical objects that 'contain' information. This paragraph analyses the concept information and its position in the ontology, together with the way it should be modeled in Gellish.

Particular individual patterns of any individual information carrier are recognized to 'carry' meaning. Therefore, apparently an information carrier has an individual aspect which can be classifies as a meaning. This means that the concept 'meaning' is a kind of aspect that is intended to classify the interpretation of an individual pattern or individual information. That pattern has a role as encoded information. Therefore, common information is defined as a common qualitative aspect that can classify one or more expressions. In other words, the meaning is encoded or expressed as a pattern. Various individual information carriers can carry 'the same' meaning. This means that their individual meanings have a common qualitative aspect: they express the same information.

Figure 54 illustrates how information or meaning relates to reality and to expressions.

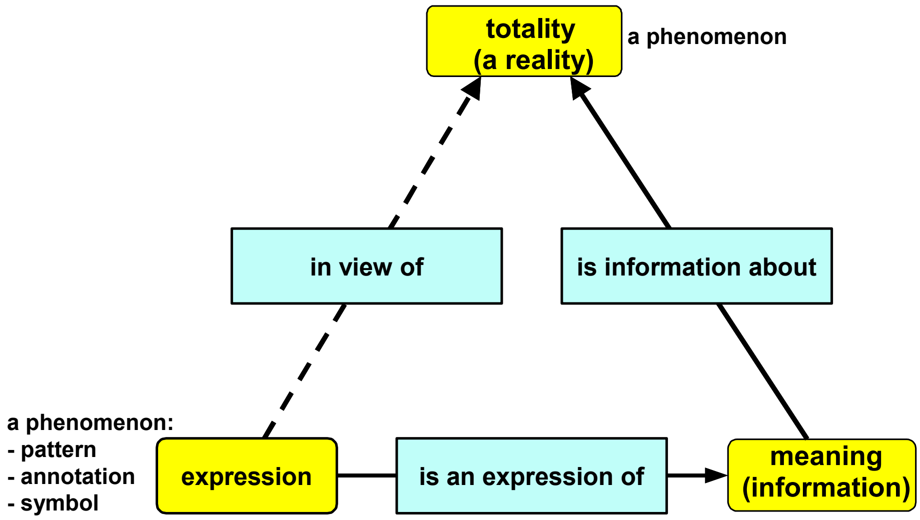


Figure 54, Relationship between reality, expression and meaning (or information)

An expression about a (real or imaginary) part of the reality (a totality) is actually a (coded) expression of information (meaning) about that part of the reality. It should be noted that the expression itself is also a part of the reality, being some aspects of sound or picture, of bits and bytes or of the human brain. This implies that each act to express information creates (shapes) a physical totality that carries an information aspect that can be recognized by a person. This means that individual information exists as aspect of an individual totality. The individual information aspects of which human beings are convinced that they express 'the same' information apparently share a common qualified information. That common information or common meaning is a qualitative aspect that 'exists' only once, and that can be used to classify various expressions. In other words, the common qualified information can be expressed in various physical forms, depending on the applied coding system. This has similarity with the fact that a qualified property can be expressed as different numbers on different scales.

This opinion about the nature of information is supported by an analysis of the *communication cycle* of expression and information that is presented in Figure 55.

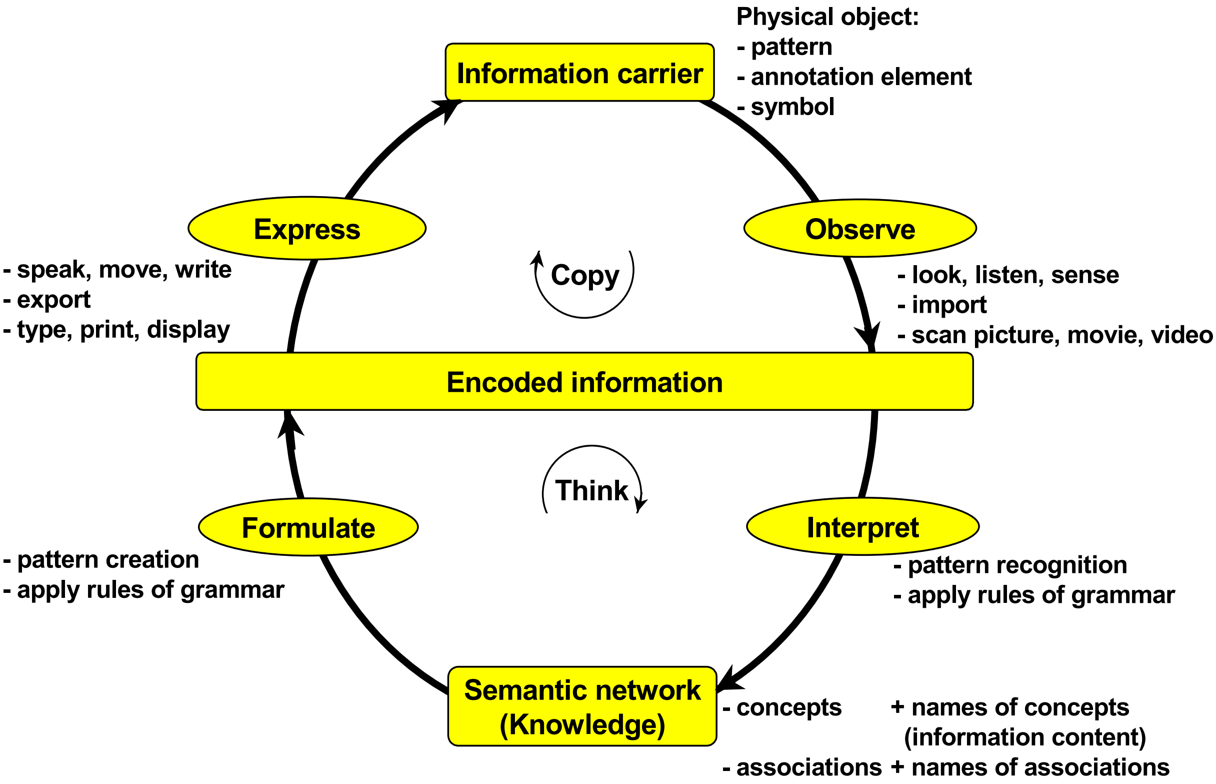


Figure 55, Expression and interpretation of information

Figure 55 illustrates that during communication, copying and thinking, a particular meaning continuously exists as a coded aspect of a sequence of different objects. During expression and observation a distinction is made between an information carrier and the (individual) encoded information. Furthermore, interpretation of encoded information results in a coding of brain cells or of a computer memory. During formulation a coding takes place that adds a pattern to a totality, by forming the totality.

From the above analysis we can conclude that information can be modeled in a similar way as other aspects. It can also be concluded that a piece of universal meaning (or common qualified information) has a unique identity (and thus should be referenced by a single UID), but that it can only exist in the form of at least one individual information aspect of an expression, which expression itself is an individual totality, formed or coded according to some coding system or language. This means that a precise modeling of a piece of information requires at least one 'is expressed as' relation with an aspect, whereas that aspect is an aspect of some information carrier. For example, the words in a particular copy of a book have an (individual) information aspect. All copies of the book contain the same common information, however, that common information does not exist anywhere. It may be even the same as the content of the master copy in a computer or of the stuff that is coded in the brains of the author. But all those versions are just different expressions in different coding systems.

The concept *information* can be defined as the *meaning of a pattern* in a physical object. This means that that physical object has a role as information carrier. The information can on its turn remain implicit or can be expressed explicitly as another pattern. This is similar to the statement that a relation is an expression of a fact.

Implicit information is to some extent hidden information. It is not expressed according to the rules of a coding system, but can be deducted by somebody who has background knowledge. Implicit information can be represented in Gellish by a single 'node object'. Such implicit information can also be classified and it can be expressed what it is about. If the latter two expressions about the implicit information are impossible, then the information node has no relations with the network; it remains isolated and lacks every context for interpretation. Therefore there seems to be no use in including the implicit information in a network model for the description of reality. For example, consider the information about how a car is composed. Such information is normally not expressed in a specification to buy a car. In a particular context it may remain implicit. Nevertheless it may be worthwhile to refer to that information, so that the 'node object' can be created and that particular information can be classified as 'car decomposition information' and it can be related to the car by specifying that the information 'is information about' the car.

Explicit information is created when the meaning is *expressed* in an *expression* in one of the many possible forms. For example, in the form of one or more written or spoken languages, as a drawing, a photo, a video, a written or presented or recorded piece of music, an instantiation of a 'data model', either or not in a database, etc. Such explicit information can also be expressed or described in the form of the Gellish ontological language. In Gellish, explicit information is expressed in the form of a network of phenomena and the relations between them.

Figure 56 illustrates such a network of coherent relations, which can be visualized as a 'cloud' of relations, which together form the expression of some information.

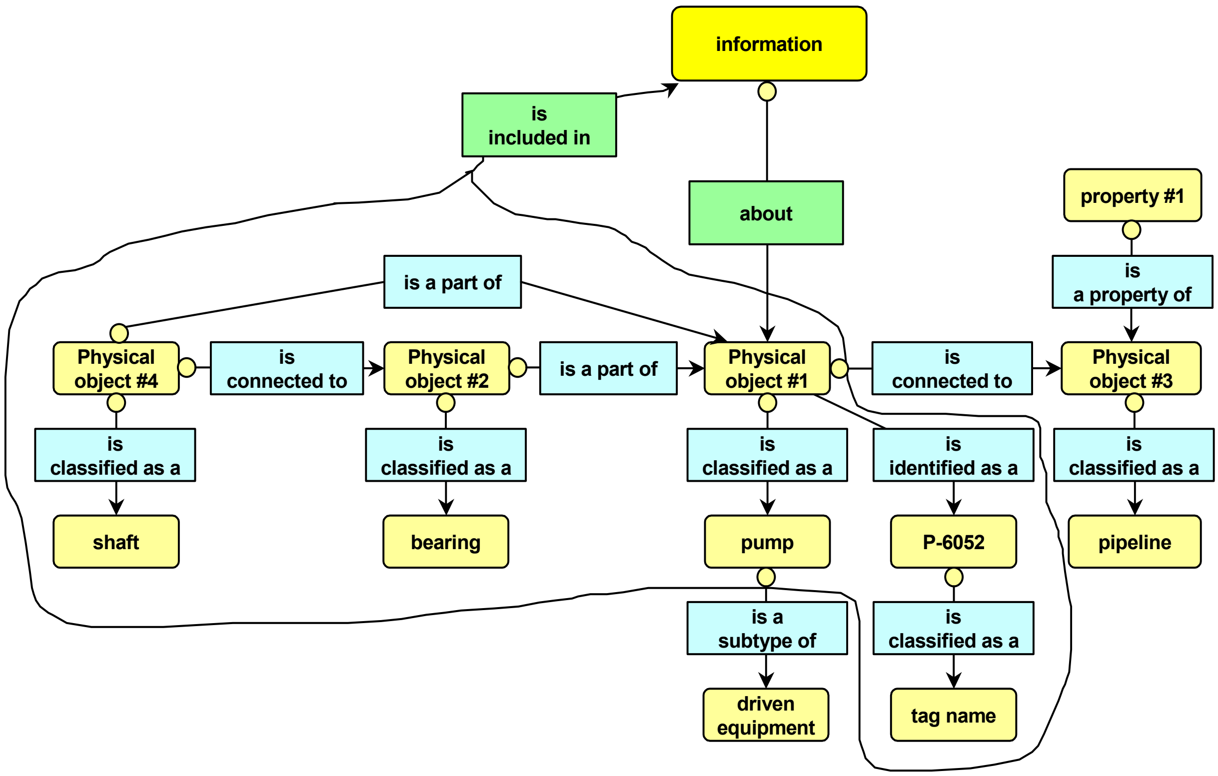


Figure 56, Example of an explicit information 'cloud' about something (a pump)

Figure 56 illustrates that we can distinguish between the information as a whole and the explicit expression of that information by a network of relations.
So, the information as a whole can be represented as a collection of relations which composition of particular relations can be expressed by inclusion relations as follows:

- relation-1 is included in info-1
- relation-2 is included in info-1
- etc.

Such a 'scope' of information may also be replaced by a rule that defines the collection of included relations.

For example, the information about a particular physical object that includes all its parts and the classifications of the physical object and its parts can be expressed by the following rule:

- information-1 about physical object #1 includes:
    - any 'is a part of' relation
    - any 'is classified as a' relation of physical object #1
    - any 'is classified as a' relation of a part of physical object #1.

Such a rule might be modeled for example in a query.

In the network we can distinguish patterns of groups of atomic relations or facts. They could be called molecular facts. A number of these can be defined as 'typical patterns' that can be parameterized and converted into templates for the expression of similar things with different values for the parameters.

Explicit information is usually expressed as a collection of (coherent) relations, informally indicated as 'clouds' of coherent relations, which expresses a 'story' about the object about which the cloud intends to provide information. The 'collection of relations' that expresses the information can be given an explicit identifier. This enables to record explicitly which relations are included in the collection. The latter implies that a number of inclusion relations ('is included in' relations) are created. Set theory relations can be used to include collections in other collections.

3.4.1 Communicative intention

Communication between people is not only informative. In other words, we express not only 'what is the case', thus strict facts. Especially in a dialog, we can distinguish other kinds of information.

This means that for a correct interpretation of information in many cases it is required to also indicate the aspect of the *communicative intent* of the information. This indicates the intention whether the information is:

- an opinion
- a question
- an answer
- a promise
- a confirmation
- a proclamation of progress
- a declaration of status or
- an acceptance or
- a rejection of a state or result.

In natural languages this communicative intent is often expressed by a different word sequence or another difference in a sentence. For example, a question usually has a different word sequence than an informative expression, such as: 'A is part of B' and 'is A part of B' (ignoring the 'melody' of a spoken expression or the question mark in a written form).

Dietz (1996) has demonstrated that in a formalized natural language it is possible to use an identical expression that is neutral to the communicative intent of the message for all the kinds of information, provided that the communicative intent is expressed separately as a qualification of the expression. This method to express the communicative intent in also adopted in Gellish, so that the intentional aspect of a relation (or of a collection of relations) indicates whether something is a question, an answer, an informative message, etc.

3.5 Reality and imagination

This paragraph discusses the distinction between real things and imaginary things as recognized in the Gellish upper ontology as is illustrated in Figure 30. Our imagination makes it possible that we can think of things that do not exist (other than in our minds) and that nevertheless have the same or similar characteristics as the real things: they are assumed to obey the laws on nature as well as the real things. This means that we can think of *realistic* imaginary things. For example, a design is a realistic imaginary artifact that is required to be realizable as well by the production of a real thing that is a 'copy' of the imaginary thing. This is discussed in more detain in section 3.5.1. But we can also think of *unrealistic* imaginary things that (partly) do not obey the laws of nature, such as is the case with the imaginary objects that are drawn by the artist Escher.

There is some similarity with the distinctions that Popper made in his three world ontology[43]. He stated that the reality consists of three realms: material entities (real objects), entities of subjective minds (mental objects, ideas) and objective knowledge (the products that express knowledge about the real world). He called them three 'worlds': world-1, world-2 and world-3 respectively. In the Gellish ontology a world-1 object is a 'real thing', a world-2 object seems equivalent to an 'imaginary thing' and a world-3 object or 'a product of the mind' is to some extent equivalent with an expression of a fact, being a relation that express knowledge about a world-1 or a world-2 object. However, Figure 30 also illustrates that Popper's distinction subdivides the world only from a particular perspective. There are also other perspectives. They are indicated by the various other branches of the upper ontology. At some of the branches a digit '1' indicates that the mentioned subtypes of that branch are mutually exclusive, which means that something can be a member of only one of the subtypes at the same time and thus belongs to either of the distinct 'worlds': it is realistic or unrealistic, but not both, although it is possible that a part is realistic and another part is unrealistic.

We know phenomena and facts by deduction from observations in the real world. Nevertheless, human beings have the capability not only to think about imaginary physical objects, but also to think of imaginary aspects and facts. The discriminating aspect that is the basis for the distinction between real and imaginary things is the *reality aspect*, with the two values: *real* and *imaginary*. On this basis the specialization hierarchy below individual thing becomes:

- concept
  - real thing
  - imaginary thing
    - realistic imaginary thing
      - design (= designed thing)
    - unrealistic imaginary thing

The imaginary things have their origin and exist in our imagination and belong to the imaginary world. Those imaginary things can be more or less realistic and can therefore be distinguished on the basis of their realism in realistic and unrealistic or surrealistic phenomena.

If it is not clear from the context what kind of thing is meant, then the object requires an additional classification. For example:

- P123	is classified as a	pump
- P123	is classified as a	real thing

On the basis of their properties, the realistic imaginary things cannot be distinguished from real things, apart from the fact that the source of information about imaginary things cannot be an observation but must be an expression of a person. This means that although we distinguish real and imaginary things, they are so similar that both kinds of things are classified by the same kinds and both can have the same kinds of aspects. Therefore the paragraphs about aspects of phenomena and relations are applicable to both categories.

Imaginary things and real things have the same position in the ontology, apart from their qualification as real or imaginary and they can have the same kinds of relations. So, they are to be modeled in similar ways.

---

[43] http://psychcentral.com/psypsych/Popperian_cosmology

3.5.1 Designs

The concept '*design*' (or 'designed thing') is defined as a subtype of *realistic imaginary individual thing* that is intended to be realized by the production of a real thing that is a 'replica' of the imaginary individual thing. This can be concluded from the fact that a designer (or group of designers) of an artifact usually builds a realistic image in his or her mind about how a later produced artifact should be, whereas the design is expressed in the documentation of that idea. Such a design has its own 'life' of being created, enhanced and modified. It (or its documentation) can be maintained and destroyed.

If it is not clear from the context that a particular thing is a design or imaginary thing, then it needs an additional classification. For example:

- P-4501 is classified as a pump
- P-4501 is classified as a design

Sometimes a designer does not create an imaginary thing, but directly creates a real mould or prototype, possibly on scale. Then the designer creates a 'design' by fabrication of a real prototype model or real inverse model (such as with a mould) from which the later real intended artifact(s) will be a copy or inverse copy, possibly with a larger size. It is also possible that more than one real thing is produced according to the design.

After the creation of realized individuals a design is usually used as the criterion to judge whether the realizations into real things are according to the design. This might trigger the idea that a design might be a *kind* of thing used to classify the realizations. However this is not the case, as the real thing usually differs from the design and the tolerances that are applied to judge whether such a deviation is acceptable partly stem from another source being the criterion whether the result is properly functioning. Furthermore, in the above-mentioned examples, where a design is a mould or a prototype, it is clear that the 'design' is an individual thing and not a kind of thing. Finally, if we make a design of a facility, for example a design of a particular building, then it is uncommon to say that the real building *is classified by* the design, but it is common to say that the real building *is a realization of* the design, whereas both, the design and the real thing both are classified as a building, irrespective of the fact that in addition to that, the imaginary building is also classified as a design and that the realized building is also classified as a real thing. For the above example of a design of a pump and a realized pump this means that the following relation applies:

- P123 is a realization of P-4501

Often designers make an insufficient distinction between the imaginary (designed) thing and the later realized real thing. The two things shall have different unique identifiers and the aspects of the one should not be possessed by the other. When a design is made, it should have only aspects of the imaginary individual object. The real thing might be replaced by another one or it might deviate from the design. Therefore, the design is not a direct description of the future real object.
This means that there is a distinction between the meaning of the following two relations:

- P-4501 has as aspect 'capacity of 10 dm3/s'
- P123 has as aspect 'capacity of 10 dm3/s'

because the first one expresses a fact that is a product of the mind, because P-4501 is a design, whereas the second one expresses a fact that must be a measured or estimated value, because P123 is a real thing.

Sometimes the future realized object is referenced as an intended, planned or expected object, whereas aspects are sometimes indicated as intended, planned or expected aspects. However, these adjectives indicate various roles of the object that plays the various roles, whereas it still needs to be indicated whether an imaginary or real object is meant. So the adjectives are indications of different kinds of relation between the objects or between the object and its aspects, but do not necessarily indicate different objects or different aspects.

Sometimes a design is called a functional object. However, an imaginary (physical) object as well as a real object has functional aspects (roles) and both have material aspects, where material aspects refer to aspects such as shape, material of construction, etc. The term functional object is unclear as is already discussed in section 3.3.8. When the term functional object is applied to a design, usually a role is meant that is played by a totality, but that itself is not a totality.

3.6 Multiplicity aspect: single or plural

This paragraph discusses the position of single and plural things in the ontology as well as the relationships between them.

Each ‘thing’ is either a single or a plural thing. This holds for individual things as well as for kinds of things. For example, a stamp is usually experienced as an individual single totality. A number of stamps (or a collection of stamps) is an example of a plural totality, which is experienced as a collection of individual single totalities, which can be seen from the fact that the name stamps is in plural. However, a stamp collection is an example of an individual composed totality that is experienced as a single totality (a single collection), irrespective of the fact that it is composed of many individual stamps. The underlying question is: when is a totality single and when is it a plural totality or is it possible that something can be single as well as plural. This is "the ancient problem of the one and the many", according to Peter Simons (in Barry Smith (1982), page 200). This is an important question for an ontology and especially for an ontological language as it should be clear which object is the possessor of aspects of the totality as well as the components, such as their mass, their price. Furthermore, what are the consequences of classifying the one, for the classification of the others and which relations express those consequences. Are there, next to plural totalities, also plural facts? In other words, does it occur that a single fact or expression by a relation implies a plurality of facts? If yes, how is that semantically defined so that it becomes computer interpretable?

3.6.1 Single individual things

To answer some of the above questions we will analyze the concept of single individual thing. Many philosophers have asked themselves: what is a unity or what is an identity? (See for example ‘The concept of identity’ by Eli Hirsch (1982) and ‘Het mysterie van de identiteit’ (the mystery of identity) by Rene van Woudenberg (2000) and, as part of that, they have studied the question: what makes a composed thing a single totality? (See for example the references in the assays of Peter Simons in ‘Parts and Moments (in Barry Smith et al (1982)).

If an individual totality is not composed of parts, then it is a *single* individual totality by definition. If the totality is composed of parts, then the kind of the connection relations between the parts determine the kind of the totality. If there are no connection relations of any kind between the parts, then there is actually not a whole, thus there is no single totality; then it is apparently a plural totality. Single individual totalities are usually still mixtures, composites or assemblies. So, a single totality is usually composed of parts, but nevertheless the composite is regarded as a single individual totality. It has a single identity. Examples of single totalities are: the atmosphere that is composed mainly of oxygen and nitrogen atoms, an atom that is composed of elementary particles, a tool that is assembled from artifacts, a living creature that is composed of cells and a solar system that is composed of components of smaller dimensions.

An assembled totality requires *coherence* between the parts that enables a combined functioning such that the parts together can play a particular role. Such coherence requires *connection relations* of any kind between the parts, or at least between a number of the parts[44]. So, composition relations, also called part-whole relations, are dependent on *connection relations*. In this context we can distinguish various kinds of connections between parts of totalities, each of which has as consequence a particular loss of freedom. When the loss of freedom is small, then the parts can still play their distinct roles in the various relationships. But connections are intended to form a larger totality that enables the parts together to play a more complex role. Thus a larger totality, is a totality in which (connected) smaller totalities form the parts. This means that an additional *totality* is created by the *connection* of two or

[44] A ‘connection’ in this context includes bindings, such as caused by attraction of masses, so that they form a system with explicit behavior as a whole. For example, the moon and the earth together.

more parts (and not by the assembly of parts and a whole). That new totality has its own identity. It is remarkable that although only connections result in a new additional totality, nevertheless people do not only recognize connection relations between the parts, but intuitively most people seem to be convinced that also composition relations exist between the new totality and its parts, although these composition relations are fictitious.

There are loose connections and strong connections. A *system* is an example of a coherent totality, in which the connections between the parts can be rather loose, but where it is intended that totality nevertheless has a clear role, played by the parts together. Therefore, a system as a whole is definitively a single individual totality.

Each single individual totality is apparently more than the sum of its parts. There are facts about that whole totality that do not hold for the individual part totalities. Examples of such facts are that the whole totality has a mass, that it has a price and that the whole totality has a capability and a location. The parts have different ones.

Various ontologies distinguish a significant number of kinds of part-whole relations. However, when the kind of things that are part and whole are already known, it seems to be questionable what the additional semantics of such part-whole relations would express. First because part-whole relations are fictitious anyway and secondly: the semantics of the part-whole relation is actually determined by the various kinds of connection relations between the parts, together with the classification of the parts and the classification of the whole either as a single totality (assembly) or as a plural totality (collection). Therefore, the Gellish ontology currently contains two composition relations: (1) an assembly relation between a part totality and a single whole totality and (2) a collection relation between an element totality and a plural totality or as Gellish expressions:

- A is a part of B
- B is an element of C

in which A and B are individual things and C is a collection.

3.6.2 Plural things or collections

For the ontology it is important to understand the nature of collections. This paragraph discusses that nature and its consequences for relations between them and between their composing elements. *Collections* consist of elements that are not arranged in a structure and don't have a clear coherence. Such collections as a whole seem to have a similar nature as single individual things. Their number is always one (it is one collection) and they are always indicated by a term in single (it is a single collection). Even when the number of elements in the collection changes, it remains the same collection. For example, a flock or a stamp collection. It is possible to express facts about such a whole collection that are not applicable for its elements. As a totality it has for example the following aspects:

- number of elements in the collection (possibly varying over time),
- a total economic value,
- an average value per element in the collection (for example based on catalogue values),
- a value distribution (a value as a function of number of elements with that value).

The latest two facts provide in an indirectly way some indication for facts about the things that are elements of the collection. This means that there can be a correlation between the facts about the collection as a whole and the facts about the elements. Nevertheless, a single collection can be called a plural thing[45].

[45] The name of a collection can also contain a term in plural. For example, the concept 'collection of stamp**s**', which seem to be a supertype of a 'stamp collection', as not every collection of stamps is a stamp collection (e.g. the stock of stamps in a post office is not a stamp collection), but every stamp collection is a collection of stamps.

Plural things are things of which the quantity of elements can vary between zero and infinity, although that number is generally greater than one. Plural things are generally referred to with plural terms (names in plural) or by the explicit mentioning of the elements. They are specified by *plural facts (see below)* each of which express something about each of the elements that compose the plurality. Plural things can be things with homogeneous elements and things with heterogeneous elements. The distinction is that the definition of a totality indicates whether the composing elements are at a particular generalization level of the same kind or of different kinds. An example of a heterogeneous plural totality is: the total stock of all things in a warehouse.
Examples of homogeneous collections are: the stock of bolts, a flock of sheep, a collection of communication devices, etc. The mutual relation between the elements in a collection can vary between 'no relation at all' to 'closely related, but not yet integrated or assembled as a single totality'. Therefore, the kinds of collections can in principle be further specialized depending on the level of cohesion between the elements. It would be valuable to further perform research on the question whether such further specialization of the collection concept is useful.

The question is whether a collection (as one) and a plural thing are different things.
For example, Peter Simons (1986) makes such a distinction. When he talks about the definition of 'manifold' and 'set as plural' he would call '25 stamps' and '30 sheep' and 'Tom, Dick and Harry' manifolds or plural totalities. When the number of elements changes, then Peter Simons calls it a different plural totality. His opinion has as consequence that for an implementation of collections, for example to record stock variations, it is required to create a series of instances: (1) the collection as single totality and (2) a number of plural totalities that succeed each other when the number of elements varies. The relationships between the first totality and the other ones specify that the first one consists of the other ones respectively. For example, a single flock is composed of '30 sheep', later of '40 sheep', etc.
To judge whether this is a valuable distinction, we should first distinguish between the concepts 'single' and 'particular'. A particular thing can be a single thing or a plural thing. If the particular thing is a single thing then the aspect 'number of elements' is not applicable, because it is not composed of elements, it is not a collection[46]. If the particular thing is a plural thing, then it has as aspect a number of elements (components) that can vary between 0 and infinity. The particular collection is one (plural) thing, and the collection has a continuity of existence, also when the number of elements varies, because the collection of facts that *define* the collection remains valid, while additional facts express that other elements are added to the collection. This means that a distinction between 'the collection as one' and 'the collection as many' is unnecessary, because both terms classify the same particular plurality. The single identity persists, because changes in the number of elements only implies changes in the composition relations and the single identity prevents that for each change a new plural thing has to be defined, including the definition of which elements belong to the collection (for those cases where that composition is made explicit).

Every relation between an element of a collection and the whole collection is classified by the relation type:

- is an element in (this specifies a membership of a plurality; $a \in A$).

Relations between plural things are of a different kind than relations between single things. Those kinds of relations when put in a hierarchy are:

- relation between collections
    - is united in (union; $C = A \cup B$),
    - is a subset of (subset/superset, $A \subset B$),
    - is an intersection of (intersection; $C = A \cap B$),
    - is the complement of (complement; $C = \wedge A$),
    - is a power set of (power set; $C = A*$).

[46] One could state that the number of elements is by definition one, but that would be a superfluous statement and the statement would be inaccurate, because the single thing is not composed of elements.

Examples of usage of the above relations are:

A	is united in	C
B	is united in	C
A	is a subset of	B
A	is an element of	AB
B	is an element of	AB
C	is an intersection of	AB
C	is the complement of	A
C	is a power set of	A

3.6.3 Single and plural kinds of things

A *single kind of thing* is one particular concept. It can be a generic concept or a more specialized concept. Examples of single kinds of things are all kinds of things with names in single, such as: physical object, nut, bolt, pump, building, atom, solar system, property, pressure, act, etc.
Many classification systems use terms in plural to refer to a particular class. For example: apples, pears, nuts, bolts, etc. However, the class ‘apples’ cannot be used to classify an individual thing, because the statement ‘A is classified as an apples’ is not a proper English expression. It appears that the plural terms are suitable to classify collections. For example, ‘Cs are classified as apples’ is a proper English expression, provided that Cs is the name of a plural individual thing. The meaning of this expression is equivalent to the two Gellish expressions:

- Cs is classified as a collection
- Cs each of which elements is classified as an apple

The latter two expressions demonstrate that a relation with a kind in plural (apples) can be replaced by a relation that express that the relation is valid for each of the elements of the collection. Thus, the single kind (apple) can be used instead of the plural kind (apples) to express the same semantics. Thus the plural kind is superfluous.
Therefore, in Gellish all kinds of things are named with terms in single, whereas collections are classified as a collection and in a separate expression it is specified to which class each of the elements belongs. This avoids that the concept ‘apples’ needs to be defined next to ‘apple’, etc. and thus avoids a complete doubling of the number of concepts in the Gellish dictionary.
Another expression that could be used to express the same fact is:

- Cs is classified as a collection of apples.

This classifies the whole collection, instead of each of its elements, whereas it refers to the concept ‘collection of apples’ that implies that each of the elements of such a collection is classified as an apple. This shows that the concept ‘apples’ is nearly synonymous with ‘collection of apples’. But semantically the above two expressions express the same fact. This demonstrates that subtyping the concept ‘collection’ by ‘collection of apples’, etc. is also superfluous. If this was not the case then the Gellish dictionary would have to be duplicated again. So, in Gellish there are hardly subtypes of collection applied, apart from some general subtypes, such as:

- collection
    - collection of items
    - collection of pairs
    - collection of aspects

There are also concepts that are single kinds of things that are composed of two or more distinct kinds of things. For example, the concept ‘nut or bolt’ is a particular kind of thing. Each of its elements is either a nut or a bolt. Because these two kinds of things are mutually exclusive, there is not a thing that

is a nut as well as a bolt. However, this is not a generally applicable constraint, as overlapping kinds can also be combined in a composed kind of thing. The relation between a composing kind and a composed kind is a special kind of specialization relation, called a partial subtype relation.
For example:

- bolt   is a partial subtype of   nut or bolt

These combinations of kinds of things are often used in some well-known classification systems, such as the UNSPSC classification system. However, such combined concepts are vague classes, because classifying a particular individual thing as a 'nut or bolt' leaves the classification rather vague, because neither the aspects of nut, nor the aspects of bolt can be allocated to the individual. Although they are valid kinds of things, it is recommended not to use them. For example, it is recommended to avoid expressions such as:

- b-1   is classified as a   nut or bolt

But if the individual thing would be classified by a single class, such as:

- b-1   is classified as a   bolt,

then it is clear what the thing is. Furthermore, it implies that b-1 by definition also belongs to the composed class 'nut or bolt', because of the fact that bolt is a partial subtype of nut and bolt.

A plural kind of thing or collection of kinds of things (concepts or classes), is also referred to in terms in plural and they also have a number of elements that can vary between 0 and infinity. For example, 'ASTM classes' is a collection of classes that is defined by the ASTM organization (the American Standards bureau for Testing Materials). The elements of these collections are kinds of things. For example:

- ASTM A670   is an element of   ASTM classes

Application of plural kinds of things typically occurs in situations where constraints apply on the selection from a limited collection of kinds. For example, an RGB signal is a signal that is composed from signals that represent red, green and blue, but no other colours. Therefore, the collection of colours red, green and blue is a plural kind of thing that is used in the definition of an RGB signal. Also 'picklists' that present multiple options for selections in a computer program are usually collections of (qualified) kinds of things, that are composed of a number of discrete kinds or options from which a user may choose. They form the 'allowed values' for concepts in that particular context.

Figure 57 illustrates how single and plural things are composed and how they relate.

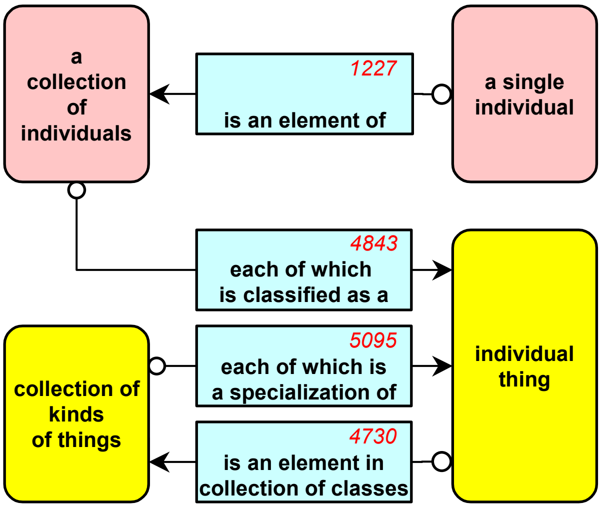


Figure 57, Single and plural things

For a correct interpretation of Figure 57 it is important to be aware that the concept called ‘individual thing’ is a kind and not an individual thing. The figure illustrates the definition of four relation types that are relevant for plural things. The upper relation type ‘is an element of’ is a relation between individual things. Every relation that is classified as a ‘is an element of’ relation specifies that an individual thing belongs to a collection of individual things. The second relation type is used to classify relations between an individual thing and a kind of thing. The third and the fourth relation types are relation types that can be used to classify relations between kinds of things.

3.6.4 Implied plural facts

Facts about plural things are facts that express directly or indirectly something about each of the elements of the plural things. This implies that for semantic reasons a relation type that expresses a fact about an individual thing is not suitable to be used to express implied plural facts about the elements in a collection. Therefore, the Gellish ontology contains separate kinds of relations to express implied plural facts. An implied plural relation is a single relation between a plural thing and a single thing, that implies a relation between each of the elements of that plural thing to be related to the single thing according to a particular kind of relation. The Gellish phrase in the expression of an implied plural fact therefore always begins with the term ‘each’. So, in stead of ‘is a part of’, the implied plural relation type becomes ‘each of which is a part of’, etc. Examples that conform to Figure 57 are:

- collection S	each of which elements is classified as a	sheep
- collection C	each of which elements is a specialization of	animal

The first relation specifies that the collection S consists of sheep, such as sheep-1, sheep-2, etc. The second one expresses that the collection C consists of kinds of animals (subtypes of ‘animal’), such as cow, bird, fish, etc.

3.6.5 Lists, tables and matrices

In natural languages it is possible to express that something is the case for each element of a collection. However, in natural language it is difficult to express verbally that something is the case for elements in a list dependent on their position in the list. For that purpose it is possible in textual expressions to define a table which expresses that something is the case for every element in a column in the table, in other words for every element in a particular position on the horizontal axis.

Before we can define the content of a table we first need to define the concepts list, matrix and table. These concepts are included in Gellish by the following definitions:

list	is a specialization of	collection of elements	that are arranged in a particular sequence.
matrix	is a specialization of	list	of lists, which lists have a corresponding arrangement of elements.
table	is a specialization of	list	of lists that is a matrix that form the table body, plus a table header and a row header.

A table header consists of up to four lists:

1. A list of kinds of aspects that classify the nature of the elements in the columns of the matrix.
2. A list of kinds of roles that classify the roles of the elements in the columns of the matrix.
3. A list of units of measure that classify the scales for the quantification of the elements in the columns of the matrix.
4. A list of relation types that classify the relations between the things that have aspects and the qualitative aspects that are the elements in the columns of the matrix.

A row header consists of:

5. A list of the things that have aspects that are qualified by the elements in the rows of the matrix. It forms the part of the first table column under the header.

The table body consists of:

6. A matrix (list of rows) that forms the parts of the table rows under the table header.

The allocations of elements to their positions in a list can be done as follows. First all elements are specified to be an element of the list (collection). Then the first element is allocated to the first position in the list. The any next element is specified to be a next adjacent element after its preceding element. A closed list can be defined by explicitly specifying which element is the last one in the list.

For such a specification the following relation types are required:

Left hand object id	Left hand object name	Fact id	Relation type id	Relation type name	Right hand object id	Right hand object name
101	A-1	201	5331	is the first element in	100	List A
102	A-2	202	5332	is the next element after	101	A-1
103	A-2	202	5332	is the next element after	101	A-1
...				…		
100+n	A-n	200+n	5338	is the last element after	100+n-1	A-n-1

A list that is defined as above can be presented on a single row, for example as a line in a table. An example of a table is an equipment table, which usually consist of a list of aspects about a number of equipment items of a particular kind. For example, a list of pumps, as is illustrated in the following table.

Left hand object name	Role	Capacity	Differential head	Design pressure
		dm3/s	**mlc**	**bar**
P-6501	Booster pump	15.0	25	5.0
P-6502	Circulation pump	5.0	20	4.0

Such a kind of table is defined for a particular kind of thing and is intended to be filled with data about either members or subtypes of that particular kind of thing. A particular pump table of such a kind can be used for data about members of the kind 'pump' or for subtypes of the kind 'pump'.

For example, in general it is a fact that:

- pump table is a subtype of equipment table
- pump can have aspects that can be presented as a pump table

A particular (empty) table with an explicitly defined content is a qualified subtype of such a pump table. For example, the above (empty) table, which we will call pump table type A, with its specific definition, can be specified as:

- pump table type A is a qualification of pump table

A table type definition can be regarded as a *template* to present or to enter data.

A *definition of a table* template can be provided in Gellish as described below. For example, the above pump table type A is defined as follows:

pump table type A	has columns that are defined by	list of kinds of aspects-A			
pump table type A	has columns roles that are defined by	list of kinds of roles-A			
pump table type A	has column relations that is a list of	list of kinds of relations A			
pump table type A	has rows that are defined by a	collection of pumps			
list of kinds of relations A	is a list of	has a role as a	has as aspect	has as aspect	has as aspect
list of kinds of aspects-A	is a list of	role	capacity	differential head	design pressure
list of kinds of roles-A	is a list of	pump role	pump capacity	pump differential head	pump design pressure
list of UoM's-A	is a list of	-	dm3/s	mlc	bar

Human beings can interpret a table quite easily, but the above table illustrates that a definition of its semantics that is sufficient for unambiguous computer interpretation requires the definition of the following seven components:

- A **list of kinds of aspects**, that has a role as **table header row.**
  Each kind of aspect in the list qualifies the qualitative aspects in the cells in the same column as where the kind of aspect appears. For example, the list of aspects A contains role, capacity, differential head and design pressure. The kind of aspect 'capacity' in the above table is qualified by the qualitative aspects 15.0 and 5.0 in the same column.
- A **list of kinds of roles**, that has a role as **(second) table header row.**
  Each kind of role in the list qualifies the role of the qualitative aspects in the cells in the same column (this row is not shown in the example pump table). For example, the kind of role

‘pump capacity’ is a role of a capacity that is quantified by the role of the qualitative aspects 15.0 and 5.0 in the same column as being pump capacities. These roles are required to be explicit in cases where constraints on the allowed values for the qualitative aspects are specified for a kind of thing. For example, there may be constraints defined for pump capacities (values for a capacity when it is possessed by a pump) that are not constraining the values of capacity in general.

- A **list of kinds of relations**, which is a **(third) table header row.**
  Each kind of relation in such a list qualifies the relations between the row items and the qualitative aspects on the same rows (this list is also not shown in the above example table).
- A **list of unit of measures (scales)**, which is an optional **(fourth) table header row**.
  Each unit of measure in such a list is used to quantify the magnitude of the quantitative aspects of which the value is presented in the same column. If there are no scales defined for that table, then the cells in the table contain qualitative aspects, but not numbers, except for numeric quantities.
- A **collection of items (or kinds of items)**, that define the **rows in the table**.
  If the collection is a list, then the sequence of the items in the list defines their presentation sequence in the table. The items define the left hand object columns of the table. Each item in the list has a relation to the qualitative aspects on the same row.
- A **body aspect matrix**.
  This is a matrix of qualitative aspects for the items (or kinds of items). The matrix in the above example pump table is the part in normal font. An aspect matrix is a **list of lists** of qualitative aspects. Examples of qualitative aspects are 15.0 dm3/s, or just 15.0.
- **Lists of qualitative aspects**, that are the **rows of the body matrix of the table**.
  The sequence of the qualitative aspects in the lists conforms to the sequence of the aspects, roles, relations and units of measure in the header rows.

An inverse table is defined in a similar same way, but with columns and rows interchanged. An example of an inverse table is a material balance table, where a list of streams define the columns and stream properties and composition (concentration of components) define the rows.

Note that neither the above tables, nor the example table below does contain UID’s, but a full Gellish Table will have them, according to the Gellish rule that each thing shall have a unique identifier. This means that each left hand object shall be preceded by a left hand object ID, the relation type shall be preceded by a fact ID and a relation type ID and each right hand objects in the list shall be preceded by a fact ID and a right hand object ID. It is optional whether or not these identifiers are displayed in a user interface.

U-6500 pump table	has rows that is a list of	P-6501 aspects	P-6502 aspects		
U-6500 pumps	is a list of	P-6501	P-6502		
P-6501 aspects	is a list of	Booster pump	15.0 dm3/s	25 mlc	5.0 bar
P-6502 aspects	is a list of	Circulation pump	5.0 dm3/s	20 mlc	4.0 bar

The table type definition implies that for the specific table, the following additional expressions are required:

- Each list in the above table has to be classified as a list or matrix or table.
- The definition of the lines that define the above tables is not dependent on their relative position in the list of lines. This means that the lines can be presented in any sequence.

This means that a ‘row’ does not need to appear visually in a table. This conclusion also implies that a table can be defined once for all, and can be used as a template many times.

For example, it is possible to define that

- P-6503	is an element of	U-6500 pumps

and to define or to determine that

- P-6503 aspects	is an element of	U-6500 pump table
- P-6503 aspects	is a list of	Feed pump, 10 dm3/s, 30 mlc, 6.5 bar

without specifying the position of the pump relative to the other pumps. This is sufficient for the interpretation of the P-6503 aspects and for the presentation of all the U-6500 pumps in a pump table according to the definition of pump table type A.

Because of the above rigorous table definition it becomes possible to automatically generate a table from the source Gellish data with atomic facts and vice versa: it is possible to generate atomic Gellish expressions from a table that is semantically completely defined in the above way.

3.7 Individuality aspect: an individual thing or a kind

Things appear to have an individuality aspect that makes apparent whether a thing has a unique independent identity or whether the identity of the thing is dependent on a mental conclusion that it indicates a commonality between other things. This means that the individuality aspect has two possible values: individual and kind.

This section discusses the relation between individual things and kinds of things, its subtype 'phenomenon' and also the further subtypes of the kind 'phenomenon', being the kinds 'totality' and 'aspect'. It also discusses the 'classification of individual' relation between them.

3.7.1 Classification of phenomena

Individual phenomena can be distinguished in (individual) totalities and aspects. There are commonalities between individual aspects that make human beings conclude that those aspects are of the same kind. For example, the qualitative concept 'red' is a kind of aspect that can be used to classify the colours of all red totalities. Even if all the aspects of a totality are classified, then nevertheless it appears that totalities themselves can also be classified. This might seem to be superfluous, because it seems possible to evaluate which conceptual aspect is used for classification of the totality and then use the various qualification (values) of those aspects to qualify the totality. However, the classification of the aspects is insufficient for the classification of the totality, because of at least the following reasons: First, the classification of a totality will generally be an essential classification that classifies the essence of the nature of a totality, without specifying which aspects determine that nature. For example, a totality can be classified by its intended role. Then it might remain unspecified which physical aspects make the totality suitable for that role. In some cases those physical aspects can even be derived from the intended role of the totality. Secondly, most kinds of totalities are characterized by a large number of aspects that are all implied in the definition of the kind. This includes all the aspects that are inherited from all the supertypes in the specialization hierarchy of those totalities. All those aspects together define a single 'nature' of the totality. Furthermore, totalities are more than the sum of their aspects. So, there are commonalities between individual totalities that make human beings conclude that those individual totalities are of the same kind. This means that a qualitative aspect is something else than the kind of totality that is qualified by that aspect. For example, the concept 'red' is another concept than the concept 'red totality'. In other words, instances of both, individual aspects as well as individual totalities, can be classified by their kind.

The classification of a totality is always done on the basis of one or more of its (usually intrinsic) aspects, whereas each totality has a virtually infinite number of aspects. This implies in theory that each totality can be classified in a nearly infinite number of kinds. Fortunately, in practice there are particular kinds of aspects that have a preference above others, probably because they determine other aspects, due to correlations and dependencies between aspects.

One of the aspects that has a preference is the physicality aspect. That aspect determines whether a totality is a physical object or not. There are even philosophers that are of the opinion that there are no other things than physical totalities that possess aspects. In other words, in their opinion there would be only one subtype of totality in the ontology, which would mean that the concept 'totality' would coincide with the concept 'physical object'. In other words totality and physical object would be synonym terms for the same concept. However, there are others who state that for example, animals and human beings are not primarily qualified as physical, but are primarily qualified as biotical and psychical phenomena. To limit the ontology on beforehand to essentially physically qualified totalities as the only totalities we can think of would mean an unwanted reduction of the expression capabilities of the Gellish language and ontology. This is strengthened by the observation that, although many people deny the 'existence' of metaphysical totalities, it should be at least possible to talk about such totalities. This makes it necessary to position such totalities in an integrated ontology. This may imply that there will be 'facts' within in the hierarchy that are qualified as 'options', because of differences in opinion on the details of the qualification of kinds of things. For example, something may be classified by one persons as real and by another as imaginary. Such differences of opinion do not obstruct a practical application of the ontology. This is the reason why physical object is (only) one of the subtypes of totality in the Gellish ontology. A discussion about the choices that are made on this subject in the current version is beyond the scope of this research.

3.7.2 Kinds of totalities and kinds of aspects

3.7.2.1 Distinction and relation between kinds of totalities and aspects

As discussed in section 3.2, the Gellish ontology makes a fundamental distinction between totalities and aspects, so that a totality is more than the sum of its aspects. This is also reflected in the specialization hierarchy, where there is a section with a hierarchy of kinds of totalities and another section with a hierarchy of kinds of aspects. Kinds of totalities are intended to classify individual totalities (for example, it may classify an individual totality as a car), whereas kinds of aspects are kinds that classify individual aspects (for example, it may classify an individual aspect as red). But an individual totality may in general not be classified by a kind of aspect. For example, it would be incorrect to classify an individual car as red, because the classified thing (the car) and the classifier (the colour) are of a different nature. The correct way is either to classify the car as a red car, or to classify its colour aspect as being red. So any of the following expressions is correct:

- C1	is classified as a	red car

and

- C1	has aspect	C1-colour
- C1-colour	is classified as	red

A classification of an individual totality as being of a particular kind of totality always means that that thing is classified by a subtype of 'totality' and on the basis of the qualification of a limited number of its aspects, which qualitative values determine the additional constraints of the subtype of the kind of totality. For example:

- car model S 40	is a specialization of	car
- car model S 40	shall have as possessed aspect a	colour of model S 40
- colour of model S 40	is by definition qualified as	red

whereas

- colour of model S 40	is a specialization of	car colour
- car colour	is a specialization of	possessed aspect
- car colour	can be a role of a	colour

Note, that the explicit definition of the concept 'car colour' enables to define constraints that are only valid for car colours, possibly only in a particular context (e.g. "a Ford car can be of any colour provided it is black"). The explicit subtype concept 'colour of model S 40' enables to define the constraint that the colour is by definition red, or only one from a collection of allowed colours (see further section 6.2).

All non-classifying aspects of the individual totality are left out of consideration for the classification of the totality. The totality has them, but they are irrelevant in the context of its classification. This also means that a kind of thing (a subtype) is generally only defined by just some *discriminating aspects* whose values (qualitative aspects) are constraining for the subtype. But the kind of thing has more aspects of which the values are not specified, because they do not constrain the kind. For example, assume that something is classified as a car. Then, from the definition of that kind it can be concluded that the thing is a totality (because car is a subtype of totality) and that totality must have all aspects that are required to be a car. However a car certainly has a colour, but that aspect is irrelevant for the classification as car. The totality will have also other aspects that are not included in the definition of a car. So the kind of totality does not exclude that its members generally have other aspects than those that define the kind. In other words, some aspects are necessary and sufficient to determine a concept. They are constraining for the concept. Other aspects are optional.

This differs from most information systems. In most systems a class is defined by a fixed number of aspects, of which some are constrained by ‘allowed values’. This implies that an instance cannot have more aspects than those defined for the class. It usually also means that the aspects that have no degrees of freedom, thus that are without options, do not appear as ‘attributes’ nor are they attribute values that are inherited from the supertype classes.

Classification of aspects is not done on the basis of their aspects, because aspects have no aspects. An individual aspect of a totality is in its nature an aspect that is abstracted from other aspects. A classification of an aspect therefore means that that aspect is classified on the basis of its own essence, in which classification other aspects don’t play a role, although other aspects may be correlated with the aspect. For example, if something is qualified as red, then the definition of red implies that the qualified thing is an aspect (because red is a subtype of aspect) and, because the thing as a whole in its essence is qualified as red, the qualified thing is a colour by its nature (because red is a qualification of a colour). The fact that it ***is*** an aspect implies that it does *not* **have** aspects. Aspects don’t possess (intrinsic) aspects. The only exception is that aspects have (play) (extrinsic) roles in relations. Examples of such relations are correlations between aspects. Those correlations are sometimes regarded as aspects of aspects. For example, a range ‘has a’ lower limit. But actually the range does not possess the limit; the totality that is the possessor of the range also possesses the limit, whereas the range and the limit are correlated, because the range is constrained by the limit. For example, a quantity of fluid mixture has a boiling range that begins at the (initial) boiling point and ends with the dew point (the final boiling point). In this example it would be incorrect to say that the boiling range possesses the boiling point and the dew point. Strictly speaking the fluid possesses a boiling point and a dew point as well as a boiling range, whereas there are correlations between the boiling point and the boiling range and between the dew point and the boiling range, which correlations express that the points are the limits of the range, as follows:

- Fluid1	has aspect	R1
- Fluid1	has aspect	BP
- Fluid1	has aspect	DP
- R1	has as lower limit	BP
- R1	has as upper limit	DP

In addition to that it is a fact that the correlation C1 is applicable to Fluid1. The relation between the correlation and the correlated aspects is further discussed in section 3.3.8.5.

3.7.2.2 Definition of the concept kind

From the previous paragraph it can be concluded that the definition of the concept *totality* (which has a role as kind of individual totalities) always is limited to the commonalities among only some aspects of the individual totalities that are categorized by means of classification or that are generated (designed) on the basis of earlier defined kinds. Similarly, the definition of *aspect* is limited to the commonalities in nature or intensity (or magnitude) of individual aspects. This also holds for the concept plural aspect.

Therefore, we can use as *definition of the concept 'kind'*:
A kind is a commonality or potential commonality of things[47]. This commonality can be in nature or magnitude of the things or in nature or magnitude of one or more aspects of the things. The commonality can be between real and imaginary things.

A commonality of kinds of totalities is a (complex) fact that can be expressed as relations that specify constraints on the aspects of the totalities. Those constraints can have the nature of ranges or of point values, possibly with implicit or explicit tolerances that indirectly specify ranges. Those constraints can be used in two ways: (1) as criteria to verify whether a thing is of the kind or (2) as boundary conditions for newly created things that are required to be of the kind. A designed thing is usually designed to be of a kind. For example, the design of a pump shall ensure that the result of the design is a pump. The same holds for manufactured things.

Note, that the above definition defines a kind as one particular concept, even if the concept has a plural nature. This concept is sometimes referred to as a 'class as one'.

The commonality of totalities can in principle be based on any of the aspects that totalities have. In other words any aspect can be used as component of the definition of a kind of totality and thus as criterion for classification (or categorization) or for the allocation of aspects to a member of the kind of totality. The commonality can also be in that fact that some aspect of a kind of totality is present or not, such as having a part or not, or it can also be in the question whether the qualification or quantification of aspects of a particular kind are within a particular bandwidth or not.

The commonality between aspects can be either in the nature of the aspects or in their magnitude (intensity).
A commonality between aspects in nature means that only the principle of the phenomena is the same, irrespective of a qualification of their magnitude, including size, intensity or extension.

3.7.2.3 Conceptual aspects and qualitative aspects

If the commonality between aspects is in the nature of the aspects, then the concept (kind) is called a *conceptual aspect,* because it is only defined as a concept. In other words, if an individual phenomenon is classified as a particular conceptual aspect then it is classified by a non-qualified or unquantified kind. For example, two things that both have a temperature, have in common that each of them possesses an aspect that can be classified by the concept temperature, irrespective of the fact that they may have very different temperatures (values).
When a commonality between aspects includes also a commonality in quality or quantity (magnitude), then they have a common *qualitative aspect* or common aspect value. For example, two things, each of which has a colour that is red, apparently have colours that both can be qualified by the same qualitative aspect, being red. And two things that have the same temperature at a particular moment in time, such as 37 °C, have temperatures over time that at that particular moment can both be qualified by the same common qualitative aspect (or quantitative temperature), being '37 °C'. Note that this qualitative temperature is a 'degree of hotness' irrespective of the question on what scale it is expressed. So it remains the same thing (has the same identity) when it is expressed on another scale, such as in °F.

This is illustrated in the table below.

P1	has as aspect	T1	
P2	has as aspect	T2	
T1	is classified as a	temperature	
T2	is classified as a	temperature	
T1	is equal to	37 °C	

[47] An example of a 'potential commonality' is the kind 'fusion reactor', as there are no fusion reactors fabricated yet. However, when they will be fabricated, they will be classified as fusion reactor. So, the concept 'fusion reactor' is a potential commonality of some possible future reactors.

T2	is equal to	37 °C	
37 °C	can be quantified on scale as equal to	37	°C

Note that P1 and P1 as well as T1 and T2 are individual things. Temperature is a conceptual aspect, 37 °C and 37 are a qualitative aspects and °C is a scale.

The commonality as being equal is determined by a *tolerance* around the point value of the quantitative aspect, because of the limited accuracy that can be achieved in practice. This tolerance is used as a rule for the definition whether a particular individual aspect may be qualified as having the commonality. In engineering practice there is a common rule that the tolerance is equal to 50% of the last decimal on the scale that is used to express the magnitude. For example, in the case of 37 °C the rule implies that temperatures have a commonality with other temperatures that are 37 °C if those temperatures have a numeric value on a Celsius scale that is greater than 36.5 °C and less than 37.5 °C. Therefore, in the Gellish language such ranges with tolerances around pivot values are defined and included as qualitative aspects.

Any kind of physical object (totality) at a high generalization level is defined mainly by means of conceptual aspects. Such kinds are therefore conceptual kinds of physical objects of which it is specified that they have particular conceptual aspects (or more precisely expressed: their members have aspects of a particular conceptual kind of aspect), without having specified what the magnitudes of the aspects are. For example, the concept pipe is defined by the fact that individuals of that kind have an internal diameter while there is no constraint on the magnitude of the internal diameter. Any internal diameter is acceptable. So conceptual physical objects have conceptual aspect (their members have aspects in concept).
A kind of physical object that is further specialized (qualified) by qualification of (some of) its aspects becomes a more qualitative physical object that has qualitative aspects. For example: ‘6 inch pipe’ is a qualification of pipe which has a nominal diameter that is 6 inch, whereas ‘6 inch’ is a qualitative aspect that is a qualification of distance. However, there seem to be no strict boundary between conceptual and qualified physical object, because always some aspects are qualified. For example, although the shape of the concept pipe may not be defined explicitly, that shape is nevertheless constrained in order to be a pipe.

3.7.3 Standard specifications and catalogue items

Various standardization organizations, such as ISO, DIN, API, etc. define standardized kinds of physical objects. Other organizations often define internal standards for the products they sell or for the products they buy. Such kinds of physical objects are qualitative physical objects that are described by a standard specification. A selling organization will have its internal standard specifications for product quality assurance and its external standard specifications that describe catalogue items from which buyers may select. On the other hand buying organizations may have their standard buyer specifications or ‘buying descriptions’ that contain the criteria for products of a kind that are possibly bought. Note that a standard specification as well as a catalogue item is a specification of a kind of thing.
One of the type of applications of the Gellish language includes that sellers as well as buyers express their specification in the Gellish language. This would enable that an automated system can assist in the verification whether seller items satisfy or nearly satisfy the buyer’s requirements.

These kinds of qualitative physical objects (with standard specifications) can also be used to judge whether realized artifacts are according to their specifications. They usually define criteria for judging whether a realized product is conform the standard specification or is of a particular type. This is especially done for serial or mass production products where the specification includes a guarantee that the delivered real items are within the ranges defined for the kind of thing. For example, a catalogue item or a manufacturer model or type defines a kind (category or type) of item, whereas the minimum quality criteria of a grade of gasoline define a kind of gasoline.

Figure 58 illustrates the kinds of relations that are required to specify qualified kinds of physical objects.

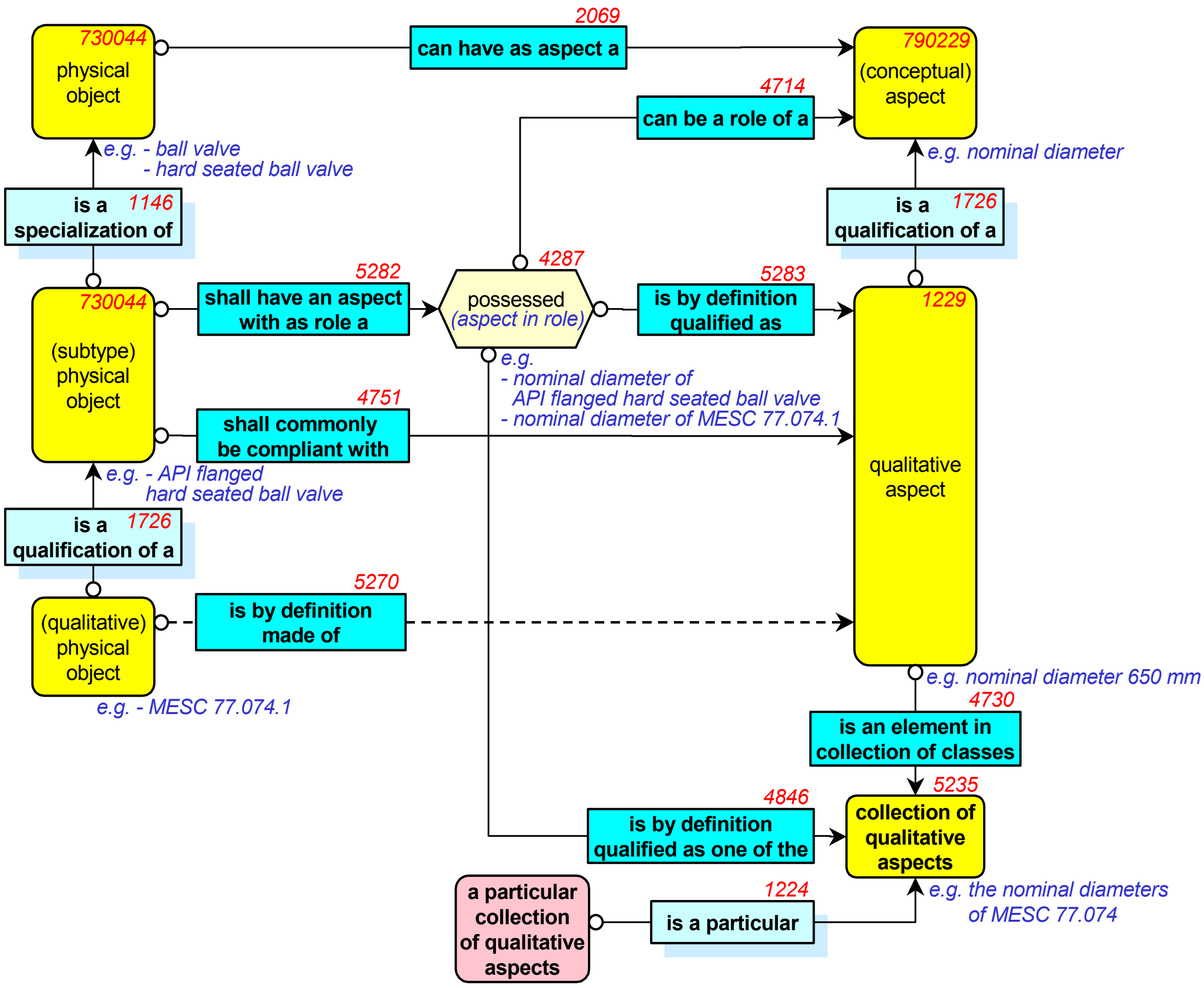


Figure 58, Relation types for the specification of standard items

The specification of qualified physical objects is usually done on the basis of a fill-in-the-blanks template or 'standard form' that is valid for a 'family' of products of a kind. Such a family specification is a specification of a (subtype) kind of physical object that is a supertype of the qualified physical objects that are created when the template is filled-in several times. Therefore, the family specification prescribes which aspects need to be specified when a qualified subtype is created and which options are available as 'allowed values' for the qualification of those aspects.
Figure 58 illustrates that a conceptual aspect is used in a role that is specific for the possessor kind of physical object. For the aspect in that role, there are particular constraints applicable (or options available). Therefore, a collection of qualified aspects can be created and it can be specified that a particular aspect in that role shall have a value from those 'allowed values'. The qualified kinds of physical objects (catalogue items) can be defined by making a selection from the allowed values and the consistency of various catalogue items can be verified on the basis of them.
The dotted line in the figure illustrates a 'short-cut relation' that can be used when the relation type implies that the qualified aspect is of a particular type. In this case the relation type 'is by definition made of' implies that the qualitative aspect is a material of construction. Therefore it is already implied that the conceptual aspect that has the role in this context is a material of construction. This makes that the short-cut relation is semantically correctly interpretable. The relation called 'shall commonly be compliant with' is not a relation with an aspect of the physical object that is specified, but it relates the object to some qualified information, typically the content of a standard specification document.

4 Information technology perspective

In information technology data models or schemes are developed either to define the structure of data in a database or to define the structure of data in an interface, for the exchange of data between computer systems. The scope of such a data model is usually limited to a particular 'universe of discourse' of which it describes the structure.

The Gellish language can be regarded as a very large and run time extensible data model that can be used as a definition of a database structure as well as an interface design. Its scope is mainly determined by the extension of its dictionary / taxonomy.

This section discusses first the essence of generic modelling as applied in Gellish together with a description of the difference between Gellish as a data model and a conventional data model. After that it discusses the role of classification and the development of a taxonomy as a hierarchy of a large variety of kinds of things. Finally it illustrates how it is possible to provide a data model that enables to describe such a large variety of things.

4.1 Issues in conventional data modelling

A conventional data model describes the structure of reality as entity types (or object types), with their attribute types and relations between attribute types of those entity types and possibly 'methods' as kinds of operations to which the objects can be subjected. Various methodologies are in use in information modelling, such as the EAR method[48], the NIAM method[49] and various object-oriented methods.

Each of those methods provides freedom of the criteria for the choice of the entity types, attribute types and relation types in a data model. Not only the choice of concepts, but also the choice of the names for those concepts is free. At first glance this seems to be an advantage, because it provides flexibility and does not put constraints on the modelling of reality. This is fine for stand-alone systems. However, differences in the use and naming of concepts has great disadvantages and should be regarded as one of the main causes of the difficulty and high costs for mappings for exchange of data between systems and for integration of data from different sources. In other words it is the main cause of lack of open data communication and data sharing in information technology.

In order to enable free data exchange and data integration by universal interpretation of data it is required that system developers share common concepts and use the same names and identifiers for common concepts. This can only be achieved through standardization of high quality concepts. This cannot be achieved through the use of the conventional methods alone, as they need additional guidance to find the answer on the following questions:

1. What is the level of generalization that is required for the choice of entity types?
   Generic entity types provide a wider scope for the instances that are supported by the entity, but less generic entity types makes them more specific and provides more precise semantics, expressed in appropriate attribute types and more precise common methods.
   The conventional methods for the design of data models have a pragmatic approach and only provide some guidance or rules. They do not provide generally valid rules or standard entity types and selection criteria for the choice of entity types. The method of 'normalization' does not provide criteria for the selection of the level of generalization.
2. How to deal with candidate entity types that appear to be roles? Should entity types that are dependent on the role that is played by an object be avoided or not?
   There are hardly generally applicable rules available in the current methods. As a result the choice of entity types is far from uniform. As a consequence, it often occurs that information in different

[48] The Entity-Attribute-Relationship method.

[49] The Natural language Information Analysis Method, also named the Nijssen Information Analysis Method, after its inventor Prof. Nijssen.

systems about the same thing is stored in different entities with different attributes. For example, dependent on the prime application focus, the choice of the entity type for storage of information about an organization might be ‘organization’, ‘supplier’, ‘customer’, ‘distributor’, etc. When the data model is intended for a system for sales support, then pragmatic considerations logically lead to the conclusion that the entity types and attribute types ‘customer’ and ‘customer number’ will be included in the data model. On the other hand, similar logic for the design of a data model for a system for support of a procurement department might lead to the situation that in that system the same organization is recorded in the entity ‘supplier’ with as attribute a ‘suppliers number’.

3. Which attribute types shall be added to the entity types and how should they be named? Should entity types and attribute types be ‘as much as possible’ be independent on the application context?
   Also for this question there is a lack of guidance. The database designer will often make a choice about this on the basis of pragmatic considerations about the information that has to be stored for the application and much less on the basis of the real product structure or on the basis of an integrating ontology about the structure of reality. For example, a data model for a procurement system with buyer’s specifications or for a system for a vendor catalogue will often reflect the perspective of the buyer or seller. A product catalogue is usually intended for products that are sold as a whole. From that perspective it is logical that all aspects of the products are recorded as direct attributes of the product as a whole. Therefore such a data model will assume that properties of parts of an assembly are recorded as properties of the assembly and no product structure can be stored. This is then usually in conflict with data models for design systems, where a product structure is essential and where properties of the parts are instances of attributes of those parts. Such differences are self-evidently reflected in the names of those attributes.

4. What should be the proper names for entity types and attribute types?
   For data models of stand-alone systems this question seems of low importance, especially because these names are usually not exchanged between systems nor are they intended for communication with the users of the database systems. Therefore, the terminology is usually chosen close to the applicable application area, although many system designers modify this terminology by using abbreviation and coding systems so that entity type names and attribute types names become artificial names.
   Problems occur when data models need to be mapped to each other because data between systems have to be exchanged. Such a mapping then often reveals that it is difficult to answer the question whether two entity types with different names and different attribute types are intended to store information about (exactly) the same kinds of things or not.
   Even during the creation of standard data models (such as those standardised by the ISO organization) is a wide spread practice to modify names of entities on purpose, in order to avoid discussions whether the object type about which information will be stored is actually the same as an existing entity in another standard data model. An example of this practice are the entity types of ISO 12006-3 that consequently added a prefix for every entity type name. Some justification of this practice is given by the fact that entity types usually do not precisely represent a true object type, but it represents a particular information collection about an object type. This makes it necessary to distinguish different collections of information.

5. When should something be modelled as an entity type, when as an attribute type and when as a relation (type)?
   This question holds especially for the modelling of occurrences (processes and activities) and for correlations between things.

6. What are the rules for proper high quality definitions of entity types? How should synonyms and homonyms for names of entity types and attribute types be modelled? How do we solve the confusion that is caused by the use of many different kinds of coding systems (still irrespective of the variety in constraints on field lengths for names)?

A pragmatic approach to the above questions has as short term advantage that it then is relatively easy to recognize that the applicable application area is reflected in the resulting data model, whereas the analyst neither need to acquire knowledge that is wider than that application area, nor need to be aware of all kind of supertypes and subtypes of the chosen entity types.

The disadvantage however is that the possible application of the resulting data model is inherently limited to the application area for which it is designed and is constrained with respect to the differences between subtypes of the entities at the chosen abstraction level. As a consequence many data models need to be extended as soon as the scope of the application area increases or as soon as additional subtypes need to be distinguished by differences in attributes or methods.

We can conclude that the current methodologies do not prescribe the choice of entity types, attribute types nor their naming conventions. Due to this lack of standardization data models that are made by different analysts generally have a large personal flavour and will be very different, even if they are made for the same application area.
In other words: *Data modellers do not use a common language.*
The consequence is that real integration of systems and data is difficult or not worth the effort and the exchange of data between systems is hampered by the complicated and costly transformations and translations that have to be made.

4.2 The Gellish language as a data model

The above-mentioned issues and the resulting constraints on data exchange and data integration has resulted in a development of a methodology to increase the applicability of data models. This methodology started with the work of Matthew West (1994) and Bruce Ottman and was called ‘generic data modelling’. The Gellish language can be regarded as a next stage of development of such a generic data model. The concepts of generic data modelling are incorporated in the definition of the Gellish language. Gellish is still grammatically generic, but is specific in its vocabulary, taxonomy and semantics.

The rules in the following paragraphs illustrate the similarity as well as the differences between the Gellish language and (generic) data models.

4.2.1 Explicit classification

Rule 1: Implicit classification through instantiation in conventional data models (or conceptual schemes) is replaced in Gellish by explicit classification relations between instances of ‘thing’.

In the earlier versions of generic data models the explicit classification was defined as a relation between instances of the entity ‘individual thing’ and instances of the entity ‘kind of thing’ (or ‘class’). This explicit classification required a ‘library’ of kinds of things, which resulted in the development of a taxonomy of kinds of things (or ‘classes’). Such a ‘class library’ allows for a larger number of classes and has a greater flexibility to add classes and to inherit aspects from supertype classes than a conventional data model with fixed entity types. The ‘class library’ therefore provides a large number of specific concepts that enable that the classification of things is more specific than often is possible with an ordinary data model.

The use of a generic data model in combination with a 'class library' is illustrated in Figure 59.

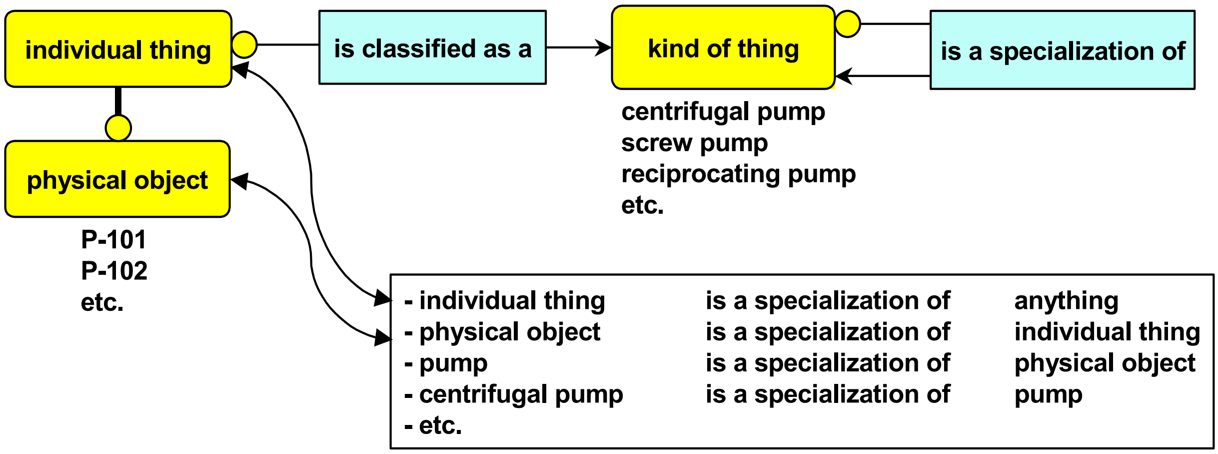


Figure 59, The relation between entity types and explicit classes (instances)

The upper part of Figure 59 illustrates a part of a generic data model in which the individual things are instances of the entity type 'individual thing', while the classes are instances of the entity type 'kind of thing'. The specialization relation enables to realize the rules that the classes shall be arranged in a subtype-supertype hierarchy.
The instances of the right hand part of the generic data model are illustrated by the instance table at the lower part of the figure. This table illustrates that the classes form a specialization hierarchy.

A further stage of development was caused by an analysis of the relation between the top of the class library and the corresponding entity types in the generic data model. Once the hierarchy of classes was extended with the higher level classes, the question arose: what is the relation between the entity types, such as 'physical object' and 'individual thing' in the data model and the instances such as 'physical object' and 'individual thing' in the hierarchy of instances? *The conclusion was that they are different representations of the same things.*

Another step was the change from a closed semantic model to an open semantic model.
A conventional data model has a closed semantic, as it fixes the relation types that can be instantiated. In first instance this was also the case with the generic data model. The rationale behind this was that an open semantic would imply that the users would get the freedom to create their own language and this would result in new lack of standardisation and consequently miscommunication. However, when it was discovered that the class library could provide the standard semantics also for the relation types, then it appeared that the semantic could become open, provided that the class library would also standardise the relation types, in the same way as the data model standardizes the object types. So, this change means that the relation types that are fixed in a conventional data model are extendable in the Gellish language.

This is illustrated in Figure 60.

- individual thing	**is a specialization of**	**anything**
- physical object	**is a specialization of**	**individual thing**
- pump	**is a specialization of**	**physical object**
- centrifugal pump	**is a specialization of**	**pump**
etc.		
- is related to	**is a specialization of**	**anything**
- is classified as a	**is a specialization of**	**is related to**
- is a specialization of	**is a specialization of**	**is related to**
etc.		
- is classified as a	**requires a role-1 as a**	**classified**
- is classified as a	**requires a role-2 as a**	**classifier**
- (an) individual thing	**can have a role as a**	**classified**
- individual thing	**can have a role as a**	**classifier**

Figure 60, Classes (entity types) and relation types included in the Gellish ontology

The upper part of Figure 60 shows a part of the hierarchy of kinds of things (classes, entity types, object types). The middle part shows a part of the hierarchy of kinds of relations and the lower part shows a part where the semantics of a kind of relation (the 'is classified as a' relation) is defined.

Note that the 'is classified as a' relation is defined as a relation between *a member of* the role-1 class and the class 'individual thing' *or one of its subtypes*. The fact that a subtype of individual thing can also play a role as classifier is defined by inheritance.

As a consequence of the conclusion that entity types are identical to classes (instances) in the hierarchy, all the entity types from the generic data model were included in the hierarchy of instances, including also the relation types. After that, the data model could be abandoned and replaced by the class hierarchy. The only data model that was left is the bootstrapping data model as is described in section 2.4.2 about the basic semantic concepts.

This means that the data model is incorporated in the ontology and the hierarchy is extended with additional concepts. In other words, the Gellish dictionary / taxonomy becomes a (very large) extendable data model.

4.2.2 Kind of roles are classes, but are not proper entity types

Kinds of roles are often used as entity types for physical objects in conventional data models. This has a severe disadvantage, because instances may get other roles, or may get several roles at the same time. This often necessitates a modification of the data model. Further investigation of the nature of the classification relation versus an implicit classification through instantiation therefore resulted in the following rule:

Rule 2: The classifying kinds of things shall be independent of the role of the classified things.

Kinds of things are intended to classify the individual things, just as entity types are intended to classify their instances. This classification (or instantiation) should classify the *nature* of the individual things, so it should classify what they *are*, independent of the role which they incidentally play or are intended to play. This general recommendation is an obligatory rule in Gellish, which means that things must be classified according to what they are and not according to the role they play. The relations between an individual thing (or instance) and its roles are not classification relations (or instantiations), but are 'has as role' or 'can have as role a' relations.

This is illustrated in Figure 61.

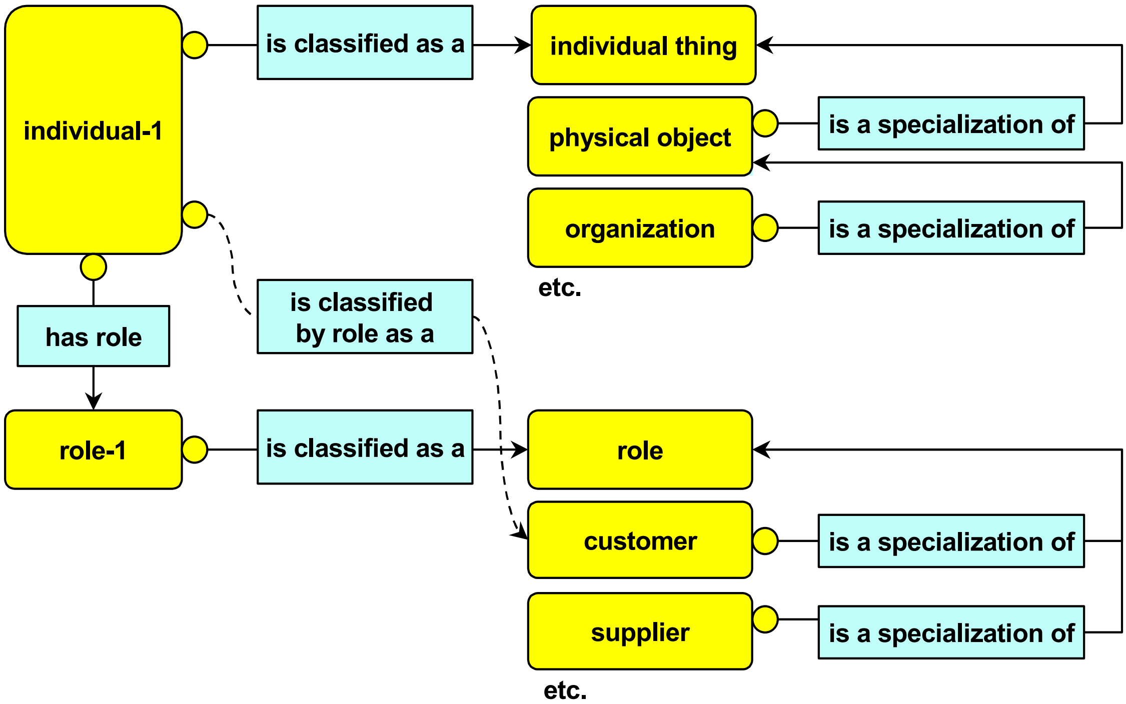


Figure 61, The nature of things and their roles

In the above figure the individual-1 has two relations: a classification relation and a (dotted) relation that classifies its role (e.g. as a customer). The latter is a 'short-cut' relation, because actually the individual-1 has a possession relation with an individual role-1, whereas that role-1 is classified by a kind of role. The figure also illustrates the application of the above rule to the above example of a customer. It can be concluded that a customer *is* not a customer by nature. It *is* an organization that only in particular circumstances has a role which role can be classified as customer. Therefore in conventional data models the entity type shall be 'organization' (or a generalization of that) and not 'customer' and in Gellish the individual company shall be classified as organization or as one of its subtypes.

4.2.3 Generic grammar

Rule 3: The grammar (relation types) shall be defined at the highest generic level of the role players at which the relation type is valid.

The grammar of the Gellish language (the structure of expressions) is determined by its standard relation types. This rule states that a relation type shall be defined as a relation between kinds of things (entity types or attribute types), which kinds shall be as generic as possible. This means that the related kinds of things that are used to define the semantics of a kind of relation are so generic that the relation type is just valid. The intention of this is that the grammar is defined on the level at which a natural language grammar is also defined. This is the level at which the relation type has the widest validity. For example, if we apply the above rule on the definition of a composition relation, then it appears that the most generalized level at which the semantics of the concept 'composition relation' is valid is when it is defined as a relation between 'individual things', because an individual thing can be composed of other individual things. This level is even more generic than a relation between physical objects. This illustrates that the semantics of a composition relation should not be defined on the level of, for example, a pump, even if an application system is about pumps only.

4.2.4 No distinction between entities and attributes

Attributes are things that have explicit relations with other things (this is equivalent to relations between attributes). The consequence of that is expressed in the following rule.
Rule 4: Relations between 'attributes' shall be made explicit, just as relations between 'entities'.

This is one of the reasons why the Gellish language does not make a distinction between entities and attributes. Each 'attribute' or 'attribute type' in a traditional data model is a thing on its own in Gellish. If a thing appears to be an aspect, then the aspect will have an explicit 'is an aspect of' relation with the thing that has a role as its possessor (this is equivalent to a relation between an attribute and a primary key of an entity). This eliminates and thus solves the issue whether something should be an entity or an attribute.
Furthermore, a conventional data model defines implicit and sometimes explicit relations between attribute types. These imply that there are relations between entities (represented by primary keys) and their attributes and also imply that there are relations between attributes. However, the semantics of a relation between kinds of things (classes or entity types) differ from the semantics of relations between individual things (being members of those classes). In other words, a relation between kinds of things expresses that a member of a kind 'can have a relation of a kind' with a member of another kind. On the other hand: a relation between individual things expresses that one individual thing has a relation of a kind with another individual thing. This difference in semantics between 'can have' and 'has' is usually ignored in conventional data modelling methods.

All the implied relations are represented in Gellish as explicit relations between things, each of which relations is explicitly classified by standard relation types of which the semantics is included in the definition of the Gellish language.

Most attribute types in a conventional data model are equivalent to subtypes of 'aspect' in the Gellish language. The Gellish language allows to add an unlimited number of subtypes of aspects in a dynamic way. Therefore, this dynamic addition of aspects is equivalent with a capability as if attributes could be added in a dynamic way to a conventional data model.
The addition of an aspect to a totality is done as follows:

1. The kind of aspect is defined by the specification of a specialization relation with its direct supertype aspect, extended with a textual description of the qualitative aspect in which it is distinguished from its 'brother' subtypes.
2. The individual aspect is created and it is defined that it is classified by the new kind of aspect.
3. The individual totality is related to the individual aspect by an 'has aspect' relation.

This illustrates that the Gellish language enables that definitions of object types can be extended dynamically and that product models can be created that consist of a network of relations that can be extended without constraints.

As said above, in Gellish a definition of a kind of thing (an entity type) does not imply a definition of a collection of attributes (other things that provide information about the thing). In other words, instead of entity types that are defined by their attribute types, in Gellish the kinds of things are only defined by their nature, whereas aspects are only related to those kinds of things as and when required, although without limitations. This implies that an individual object (an instance) does not represent a particular collection of aspect (attributes), but only represents the object itself with its unique identity. On the other hand, the definition of possible kinds of aspects for members of a kind of thing, does not constrain the aspects of those members to those kinds of aspects. Constraints only apply when it is explicitly specified that in a particular context members of a kind 'shall have' aspects of a particular kind. In other words, the attribute types are not pre-defined. For example, the definition of a pump in Gellish does not specify which aspects a pump may have; its number and kind of aspects in unconstrained (unlimited!). Nevertheless it is possible that it is expressed in Gellish that for example, a pump 'can have as aspect a' design pressure. This means that this is a recognized possibility. It does not specify that it shall have a design pressure, nor does it specify that it cannot have other aspects. Only when it is specified that (in a particular context) a pump shall have as aspect a design pressure, then it is obligatory that individual pumps must have design pressures. This still allows that in a particular file such design pressures are not available. The content of the file can still be correct Gellish, although it does not comply with those requirements

So, an individual thing in Gellish represents a 'Ding an sich', which identity is represented by the unique identifier. By relating this identifier to multiple 'names', it is possible to *refer* to the object through multiple names that are each others synonyms (possibly in different languages). The aspects are allocated to the individual thing through relations between the identifier of the individual thing and identifiers of aspects. The individual thing can have as many aspects as are required by all its applications, without being determined by its classification (or its instantiation relation).
This differs from conventional data models, where the entity represents a particular collection of attributes with information about the object and where the unique identifier (the primary key of the entity) actually is an identifier of a collection of attributes.

4.2.5 The nature of an identity.

Role 5: The nature of each identity shall be defined through an explicit classification or specialization relation.
The Gellish language makes a clear distinction between individual things and kinds of things (concepts), and requires that each individual thing is classified by a kind of thing, while the latter must be an existing kind that is already part of the Gellish language definition or of its explicit proprietary op 'open source' extension by having an explicit specialization relation with an existing supertype kind of thing.
For example, if in Gellish a particular thing with unique identifier 123456 is classified as a car, then this is not an instantiation relation between an instance and an entity type, but a relation between two instances, which relation is classified as a classification relation.
The treatment of individual things and kinds of things both as instances enables to add kinds of things as well as individual things to a Gellish database (a database with Gellish expressions). This enables to extend the language definition by the addition of classes (kinds of things) in a similar way as how individual things are defined and added to the vocabulary. It also enables to express knowledge by relating classes by relations, which relations are classified by kinds of relations in a similar way as individual things are related by relations that are classified by kinds of relation between individuals things. It also means that the individual things with their aspects shall comply with the knowledge that is modelled about the kind of thing, which implies among others that the values for the aspects are within the boundary values for the aspects of the kind.

Definition of instantiation.
The method of explicit classification, instead of implicit classification through instantiation in conventional methods, raises the question how this classification relation relates to the implicit instantiation relation between an instance and an entity type (or object type) in most conventional data modelling methodologies.
In those methodologies it is allowed that for the same kind of thing there may exist different entity types with different collections of attributes. This is necessary, because a particular entity type defines a particular collection of attributes for the kind of thing, whereas different applications require different collections of attributes for the same kind of thing. Therefore, the entity type definition combines a definition of the nature of the instances with a definition of a collection of attributes. Furthermore, different entity types for the same kind of thing will generally have different collections of instances, which collections may partly overlap each other. The conclusion of this is that the instantiation relation is a combination of a collection relation and a classification relation, whereas the classifying entity type constrains the attributes of the instances to the collection of kinds of attributes that are defined for the entity type.
The explicit classification relations in Gellish language do not imply a collection relation and do not constrain the aspects of the classified thing. If required, a collection can be defined and the elements can be included in the collection using a collection relation.
The Gellish language stimulates to define as many specialized subtypes as required, where conventional models may limit the complexity of a data model and create an attribute, often called 'type'. For example, in Gellish it would be recommended to create a number of kinds of stamps

arranged in a hierarchy, such as stamp, and its subtypes Dutch stamp, German stamp, etc., next to the definition of a kind of thing named ‘stamp collection’. Assume that a particular stamp, say S1, is classified as a Dutch stamp while it is declared that S1 is an element of C1 (whereas the latter is classified as a collection of stamps). In a conventional method there are various options. Possibly a database designer would define an entity type called stamp, with two attributes: a ‘type’ and a ‘collection’, whereas those attributes will either be pointers to indicate that they represent relations or they will be free text (which text may be constrained to the values in a pick list). But the designer could also define entity subtypes and instantiate S1 as one of those subtypes. This illustrates that the semantics can be defined in various ways, which requires conversion if two systems have chosen different solutions. On the other hand, there is only one unambiguous way in the Gellish language, being a classification of S1 according to the most specialized subtype that is required.

4.2.6 Kinds of things versus entity types.

In the Gellish language, everything is in principle defined by the semantics of the relations between things. For example, if something is related to something else by a classification relation (which is defined as a relation between an individual thing and a kind of thing, in which the kind classifies the individual), then the semantics of that relation determines that the classified thing is apparently an individual thing and the classifying thing is apparently a kind of thing. Furthermore, the classifying kind of thing will have a specialization relation with its supertype kind of thing, and a relation with a discriminating aspect (the ‘definition’). These two relations determine the meaning of the classifying kind of thing and thus indirectly they determine what the individual thing is. This means that if the kinds of the relations between things are inconsistent with respect to the definitions of the nature of the related things, then the relations form a semantically incorrect Gellish collection of expressions. In other words, a new thing that has a specialization relation with another thing and thus appears to be a kind of thing, defines a new concept, which can be used for classification of other things, without fixing the kind of information (the collection of attributes) about the members of the kind. By applying this method rigorously, the Gellish language became the equivalent of a data model with thousands of entity types, whereas it is extended continuously through its ‘open source’ dictionary / taxonomy.

The Gellish language and related methods have ‘data driven’ definitions of kinds of things (‘class definitions’), which makes them more flexible than ‘hard coded’ entity types and object types. Furthermore, in the Gellish language, the difference between the data model (or conceptual schema) and the data (the instances) disappeared. The interpretation rules for the semantics of the data is not contained in a data model anymore, but the relations between the things determine the context from which the semantics can be derived.

Depending on someone’s perspective the Gellish dictionary / taxonomy can be regarded to be:

- A normal dictionary, because it contains identifiers, names, synonyms and definitions of kinds of things.
- A taxonomy, because it contains subtype/supertype relations between kinds of things.
- A (large) data model, because it contains definitions of kinds of relations that specify possible relations between members of the kinds of things.

The advantage of the above described extendibility is that the data model can become large and widely applicable. As a result of that, the Gellish language is applicable for a wide variety of application area’s, such as machines and their components, airplanes, ships, roads, buildings, nuts and bolts, liquids and vapours, persons and organizations as well as their capabilities, roles and behaviours, including the occurrences and processes in which they are involved. This means that in many cases the Gellish data model is so widely applicable that no other data model is required anymore, or that only a dictionary extension is required to make it applicable for a particular application area. This general applicability is further discussed in section 4.4 about the question: ‘Are data models becoming superfluous?’

The main obstacle for wide application of the Gellish language will probably be the time it takes to develop generic algorithms for searching and for optimisation of the performance.

The Gellish language does not pretend to be complete, not does it pretend to be the only way to model reality, but it does pretend that it is at least complete enough for the description of the structure and behaviour of things that occur in physics and technology.

4.3 Transformation of conventional data models into Gellish

This section discusses the transformation of conventional data models (conceptual schemes) to the Gellish language.
In conventional data models an entity type (or object type or class) usually is defined by the definition of a number of attribute types. This means that the entity type with its attribute types actually forms a template for the recording of information about entities (instances of the entity type). When an instance of such a template entity type is created, it means that an entity is created that is defined to have room for a collection of attribute values of types that are on beforehand allocated to any instance of that entity type. The method to create conventional data models is therefore rightly called 'information modelling'.
An example of the definition of two entity types 'pump' and 'line', each with a number of attribute types is given in Figure 62.

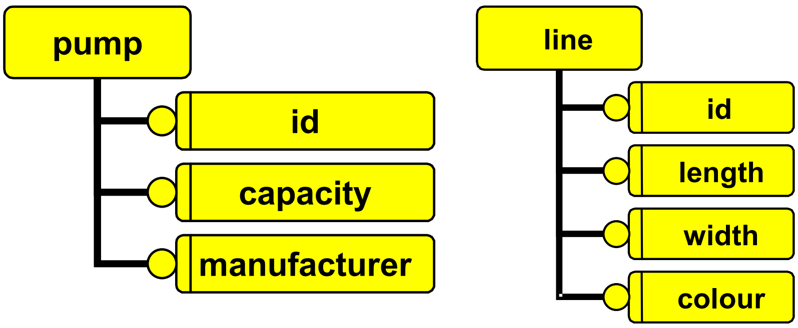


Figure 62, Examples of entity definitions: entity types with attribute types

It is a rule in Gellish that the definition of new concepts requires that they are added to the Gellish dictionary / taxonomy by defining them as subtypes of existing concepts to create one consistent specialization hierarchy of concepts. Assume therefore that the supertype concepts of pump, capacity and manufacturer (being rotating equipment item, mass flow rate and maker) exist already, but that those concepts themselves do not exist yet. Then we can transform the entity type and attribute types by adding pump, capacity and manufacturer with their UID's to the Gellish dictionary / taxonomy as follows:

Left hand UID	Left hand object name	Fact UID	Relation type UID	Relation type name	Right hand UID	Right hand object name
130206	pump	1	1146	is a specialization of	130227	rotating equipment item
551564	capacity	2	1146	is a specialization of	550021	mass flow rate
990104	manufacturer	3	1146	is a specialization of	5161	maker
990104	manufacturer	4	4714	can be a role of a	990001	company
130206	pump	5	2069	can have as aspect a	551564	capacity
130206	pump	6	5157	can be manufactured by a	990001	company

Note that the identifier (the id in Figure 62) is not included in the mapping because it actually represents the concept 'pump'. In Gellish such an id is treated as a name, just as anything in Gellish can have a name and/or identifier. Therefore, also the capacity, the manufacturer role and the company in the above example have an id or name.

The first three lines (being expressions of facts) in the above table specify an extension of the Gellish dictionary / taxonomy. They specify an addition of the concepts pump, capacity and manufacturer, assuming that they would not be present already and assuming that rotating equipment item, mass flow rate, maker and company were already defined before. The last three lines specify an extension of the Gellish *knowledge base* with knowledge about pumps. Note that none of these does require an extension of the Gellish grammar, because the concepts (relation types) 1146 ‘is a specialization of’, 4714 ‘can be a role of a’ etc. are already defined as part of the Gellish language.

In conventional data models the entity type definition does not specify the kind of relation between the entity type and the attribute types. *The transformation into Gellish makes the semantics of the relation types between an entity type and the attribute types explicit.* This is illustrated by the above example, where the relation types between pump and capacity and between pump and company are explicitly defined as 4714 ‘can have as aspect a’ and 5157 ‘can be manufactured by a’ relation types respectively.

Integration of the above definition with another (part of a) data model may include the integration of overlapping entity types. For example, assume that another model specifies basically the same entity type ‘pump’, but with some other attribute types. For example, it may add as attribute type ‘pump type’ (with allowed values: reciprocating pump, centrifugal pump or rotary pump) and ‘design pressure’. The integration in Gellish just means superposition of definitions and knowledge, as follows:
In the Gellish language it is not allowed to create a new unique identity (UID) for the same thing, even if other aspects of the thing are to be recorded, although it is allowed to allocate an additional synonym name for the thing and it is allowed to express that specific requirements only apply in a specific context. Therefore, in Gellish the attribute types of another entity are always only additions to the existing collection of relations, either as possession of aspect relations or as specialization relations or as other kinds of relations. For example, an entity type ‘pump’ with the additional attributes ‘pump type’ with the above three ‘allowed values’ imply specialization relations with the concept pump. This means that the attribute transforms into the following addition of concepts (ignoring their UID’s) to the Gellish dictionary / taxonomy (assuming that they were not yet present):

reciprocating pump	1146	is a specialization of	pump
centrifugal pump	1146	is a specialization of	pump
rotary pump	1146	is a specialization of	pump

Once the knowledge is added to the Gellish knowledge base, the knowledge can be used to derive *which* information can be specified about individual things, such as pumps, compressors and lines. It can also be used to derive *how* information can be specified, because it is defined in the Gellish dictionary / taxonomy that a ‘can have as aspect a’ relation can have as realization a ‘has aspect’ relation. The above expression, that a pump can have as aspect a capacity, means that an individual thing that is classified as a pump (or one of its subtypes) may have (or typically has) an aspect, say C1, that is classified as a capacity. For example, it defines that the following expressions are semantically valid:

P-6501	1225	is classified as a	pump
P-6501	1727	has as aspect	C1 of P-6501
C1 of P-6501	1225	is classified as a	capacity
C1 of P-6501	5020	is qualified as	5 kg/s

However, the Gellish language allows that information about individual things (in this case pumps) is specified without the necessity that the knowledge about the kind of thing is specified, provided that the concepts are available, including the grammatical concepts (such as the kinds of relations: 1225, 1727 and 5020 in the above example table)! *This illustrates that Gellish has similar capabilities as a natural language in which the availability of concepts is also sufficient for the expression of* *information. This is opposed to conventional data models, where an entity with attributes acts as a template for the creation of instances and where instances cannot be created without the availability of such templates.*

If for example, at another occasion, the design pressure of the pump is specified, then that information can be merged in a very simple way by just adding the new expressions to the existing expressions, provided that the same UID for the pump is used. This illustrates that it is important that various parties that work on the same subject should not create new identities (new UID's) for things that are already defined by somebody else and that agreements should be made to avoid overlapping ranges for the allocation of UID's to new things.
On the other hand there is no problem if the same thing (having the same UID) is named differently in different contexts (then those names are synonyms), as long as those contexts are explicit and as long as it is agreed which contexts are used as 'language communities', because to avoid confusion names shall be unique within a context.

The consolidation or data integration of different data models in conventional methods often means that yet another entity type is defined with the smallest common denominator of attribute types of its predecessors or by defining a number of new entity types in a specialization hierarchy.
In Gellish consolidation and data integration is relatively simple: new concepts are added by including them in the existing specialization hierarchy of concepts, while standardising their names and synonym names and by simple addition of relations to the existing ones. For example, a design pressure as attribute of a pump can be added by the following Gellish expression:

pump	can have as aspect a	design pressure

Note that the aspects that are possessed by pump are by definition inherited by all subtypes of pump.

A characteristic of conventional entity type definitions is that attribute types are only defined in the context of their possessor entity type. This means that an attribute type is not a context independent concept, but actually is a 'role of a concept'. This implies that attribute types of different entity types are defined independent of each other. They may have the same name, but that does not necessarily mean that they are the same kind of thing. Their entity type is their definition context. This also implies that attribute types are usually not arranged in a specialization hierarchy. In the Gellish language, an aspect concept is defined independent of its possessor, whereas the specification of a relation with a possessor defines a role of the aspect. For example the concept 'capacity' is defined above as a specialization of mass flow rate, irrespective whether it is a capacity of a pump or of a compressor or of something else. So, the fact that a compressor also can have a capacity, is expressed by a relation with same concept 'capacity' as follows:

compressor	can have as aspect a	capacity

Figure 63 presents a summary of the way in which the example entity type of a pump can be transformed to an extension of the Gellish dictionary and knowledge base.

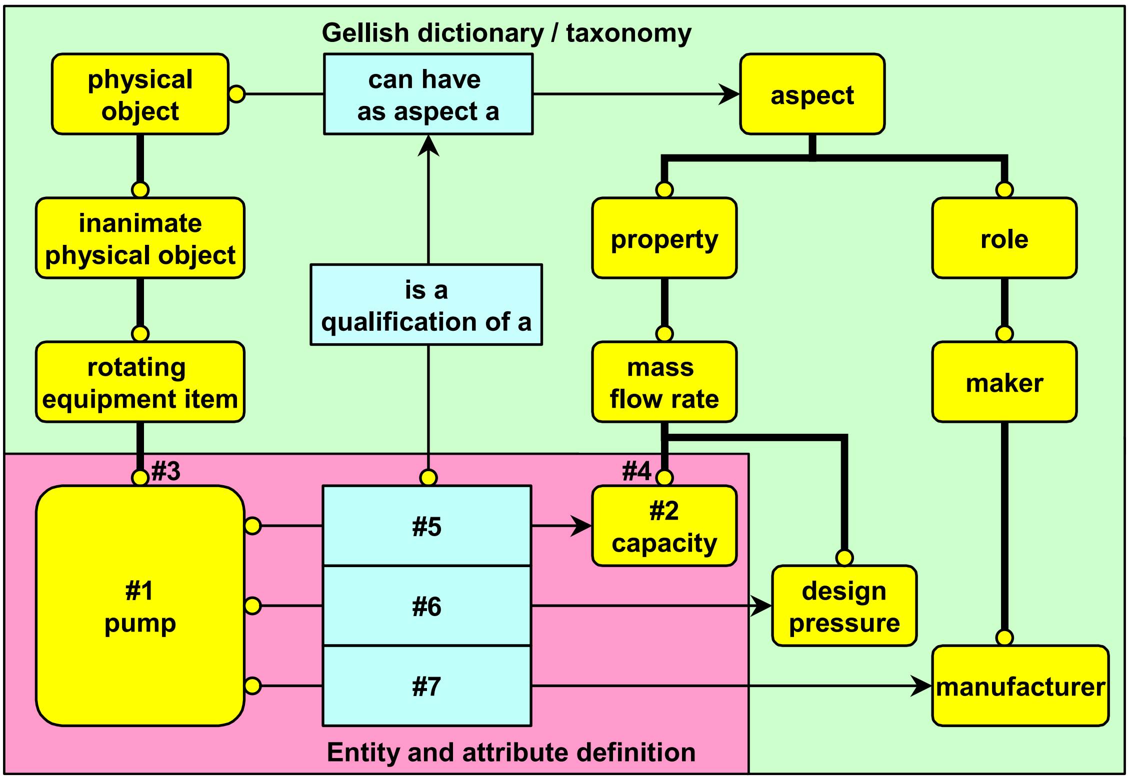


Figure 63, Result of a transformation of the entity type pump into Gellish

The shaded bottom left corner of Figure 63 contains the concepts and relations (#1, 2, 3 and 4) that are added to the Gellish Dictionary / Taxonomy and the relations 5, 6 and 7 that are added to the Gellish knowledge base.

Note that the above expressions make use of the existing concepts from the Gellish dictionary / taxonomy for kinds of relations as well as for kinds of things that are related.

The above conversion of the definition of a conventional entity with its attributes demonstrates that the semantics of an entity/attribute definition can be replaced completely by Gellish expressions and can thus be added to the Gellish dictionary / taxonomy and knowledge base. It also illustrates that once the concepts are available in Gellish, the addition of knowledge that is contained in entity definitions is not necessary, because without that it is nevertheless possible to express information about individual things (equivalent to instances of the entities).

4.4 Are data models becoming superfluous?

In this paragraph it is illustrated that the wide scope and flexibility of the Gellish language makes in principle conventional data models superfluous. This is mainly achieved by the elimination of the barrier between on one hand the definition of the semantics of application areas, as expressed in data models, and on the other hand the contents of databases, as expressed as data model instances.

4.4.1 Integration of entity types and instances

A common characteristic of conventional data modeling methodologies is that they make a strong and strict separation between the concepts defined in a data model which define an (empty) database structure (and which define the semantics, the meaning of the user data!) on one hand and the user data as stored in the database on the other hand. As a consequence it is typical for these methodologies that the concepts used to define the rules for interpretation of the data (as defined in the data model) are not accessible or extendable by normal users of database applications and are usually hidden for them by the “user interface”. This separation between meta data and data differs from human communication via natural languages, where it is possible to provide interpretation rules in the same language in which the exchanged data is expressed.

Note: This separation between the two worlds enables the common practice that entity types and attribute types have encoded names that are explicitly different from the names of object types in natural language and in normal business and engineering practice. These different conventions between IT models and user data are an obstacle for communication between those disciplines and for the integration of the two worlds.

An additional constraint of the conventional methodologies is the fact that each data model, and thus the semantics for the interpretation of the user data, is fixed once the (empty) database is defined. Any extension of this semantics requires a redefinition of the structure of the database and a transfer (conversion) of the data from the old to the new database structure.

A further constraint of the conventional methodologies is that either the scope of a data model is limited, or the data model becomes very big or very generic which causes that the model is difficult to manage and apply. On the other hand, generalization of data models may lead to a wide scope, but has as disadvantage that it leads to loss of accuracy of the semantics.

Finally, each conventional data model is different, so that exchange of data between different systems means that the data shall be converted from one data structure to the other and vice versa. In many cases the semantics of one model is richer than in the other, so that semantics is lost during conversion from the rich to the less rich model. These differences are caused by the fact that the conventional data modeling methodologies do not have a common, systematic and standardized approach to the reuse of elsewhere defined concepts.

The result of the current state of the art is that data storage is done in a Babylonian mix of data models with the consequence that exchange of data between systems is impossible, except where dedicated bilateral translators are created between each pair of data models.

The Gellish language implies a modeling methodology that does not have these constraints and does not have the distinction (and barrier) between the user data and the meta data in the data model. On the contrary, it consists of an extensible semantics, expressed in Gellish itself. This universal data structure is equivalent to a data model of over 20.000 entities and attributes selected from natural language concepts. Furthermore, the Gellish semantic concepts are standardized, so that not every user can arbitrarily modify the language.

The flexibility of the semantics is achieved by:

1. Enabling the storage of knowledge about classes (kinds of things) in addition to knowledge about individual things in the same data structure.
2. Eliminating the difference in treatment between attribute types and instances, by defining the attribute types as classes on their own in the ontology in the same way as the instances and replacing the instantiation by explicit classification relations between individual values (instances of attribute types) and the applicable class that classifies them.
3. Eliminating the difference between entity types and attribute types and replacing the implicit relation between entities and their attributes by explicit classified relations between things.

Figure 64 compares some essential concepts in conventional methodologies with concepts in the Gellish modeling methodology.

Conventional Data Model Concepts	Gellish Concepts
Instantiation - Implicit classification relations	**Explicit classification relations**
Entities have Attributes - Implicit relations between entity and attributes - Implicit roles of objects in relations	**Explicit Relations between objects** **Explicit roles of objects in relations**
Subtyping of entities (not of attributes) - Methodology does not require a consistent subtyping strategy - Usually a limited use of inheritance	**Specialization relations between classes** - Methodology requires that every class has at least one supertype, which results in one integrated specialization hierarchy - Full use of inheritance
Entity and attribute types are not instances (but are hard coded in a data model) - The data model is a hard coded knowledge model (meta model) outside the database	**Meta data (classes) are instances just as application data** - The data model is a flexible knowledge model stored as data in the database

Figure 64, Comparison of conventional data model concepts with Gellish concepts

The Gellish modeling methodology is supported by:

- A Guide on the extension of the Gellish language.
- The Gellish Application Manual, a Gellish user guide.
- The Gellish dictionary / taxonomy and knowledge base database, available as a set of Gellish Tables (e.g. in EXCEL).
- A Gellish Browser application.

4.4.2 Example of integration of data and semantics in Gellish

This paragraph illustrates how data and semantics are integrated in Gellish English.
We will use the example of the fact:

- a particular pump ('P-1') is pumping a particular stream ('S-1').

In a conventional data base it is required to declare some entity types and attribute types that define the semantics for the interpretation of such a fact in the form of a data model. In case of the example, the data model could for example consist of the entity types 'pump', 'process' and 'stream', each with some attributes.

In Gellish, the concepts 'pump', process' and 'stream' are concepts (without attributes) that are defined using expressions (instances of relations) that are included in the Gellish language, through expressions in one generally applicable Gellish Table. The table has a structure that supports the 'basic semantic structure' of Gellish, and the table contains the definition of a large number of concepts, such as 'pump', process' and 'stream', as is explained the previous sections.

4.4.2.1 Linking expression elements to Gellish concepts

In conventional database technology the semantic interpretation of an expression is done via the fact that any object is an 'instance' of an entity type (or object type) of which the semantics is predefined. For example, if P1 is an instance of an attribute 'name' of the entity type 'pump', then apparently P1 is the name of a pump, although there is no explicit (computer interpretable) relation defined that expresses that an instance of 'name' is a name of an instance of 'identifier'. It should be noted that such an instance also implies an implicit classification of the object P1 as being a 'pump'.

In Gellish all semantics is made explicit by the creation of explicit classification relations between the elements of the expression and the classes in the Gellish library of concepts, instead of instantiation relations with entity types. This is illustrated in Figure 65.

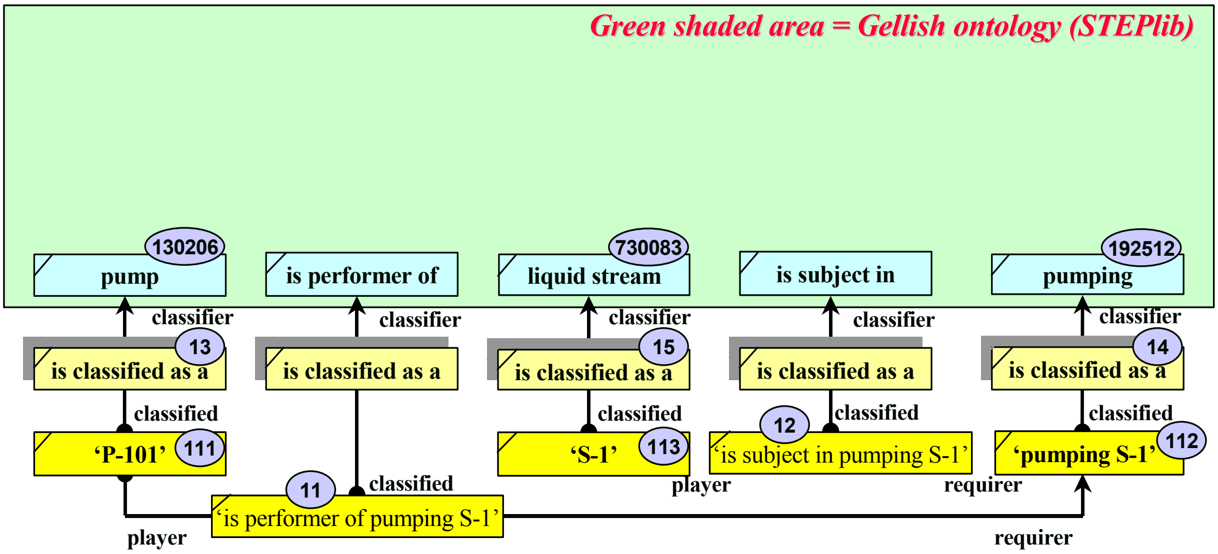


Figure 65, Links between an expression and Gellish concepts through classification

Figure 65 illustrates the expression that "P-101 is pumping S-1" (the bottom part). The 'pumping S-1' process is an interaction between the fluid S-1 and the pump P-101. The pump has a role as performer and the liquid has a role as subject in the pumping process. The boxes in the shaded area represent the Gellish concepts, being instances in the Gellish dictionary / taxonomy. The explicit classification relations with the concepts in those boxes provide the semantics for the interpretation of the expression.

Note that the shaded boxes all have the same name: "is classified as a". However, they are different individual classification relations. Each of those relations has a unique identifier. The name in the shaded box indicates that each of them is classified as a classification relation. In other words, each of them is a "is classified as a" relation.

The definition of the concepts in the Gellish dictionary / taxonomy is done via specialization relations as is illustrated in Figure 66.

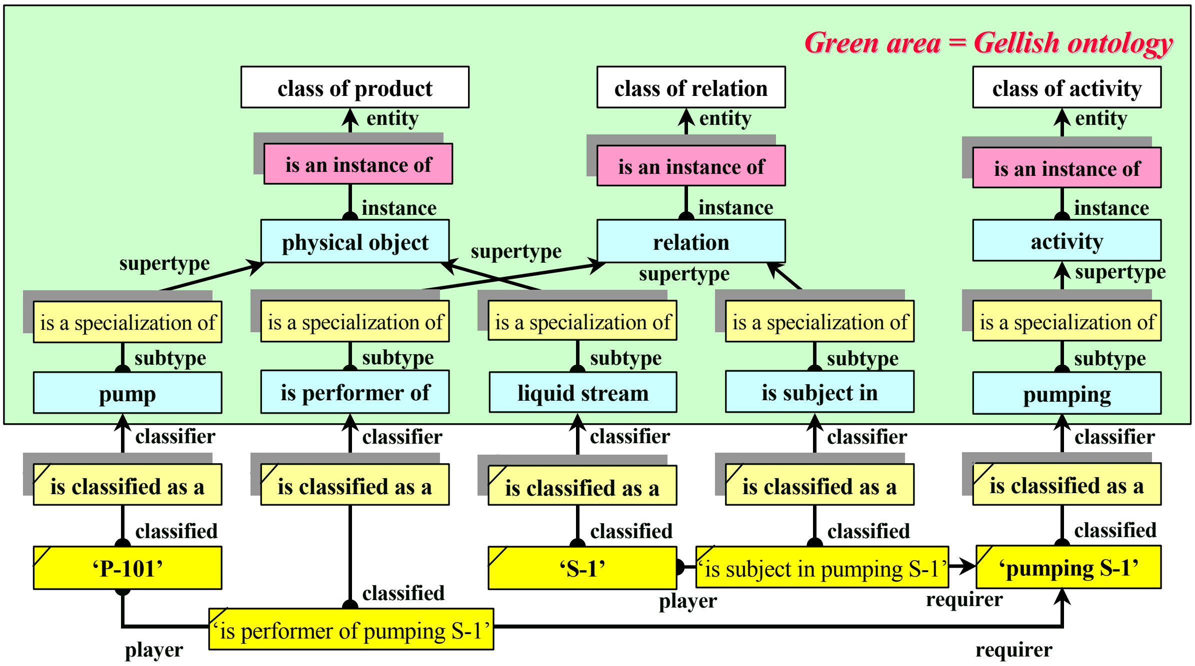


Figure 66, Definition of Gellish concepts in a specialization hierarchy

In practice there are several intermediate levels of specialization between e.g. 'pump' and 'physical object', etc.

This differs from the use of generic data models as defined for example in the ISO 10303-221 (AP221) and ISO 15926-2 standards, where a separation between the meta data in the data model and the instances is maintained. Actually those data models are equivalent to a selection of some of the higher level concepts to form entity types (which are given different names with non-natural, IT specific, naming conventions). This means that the use of those data models require instantiation relations between the concept in the library and the data model entities as is illustrated in Figure 67.

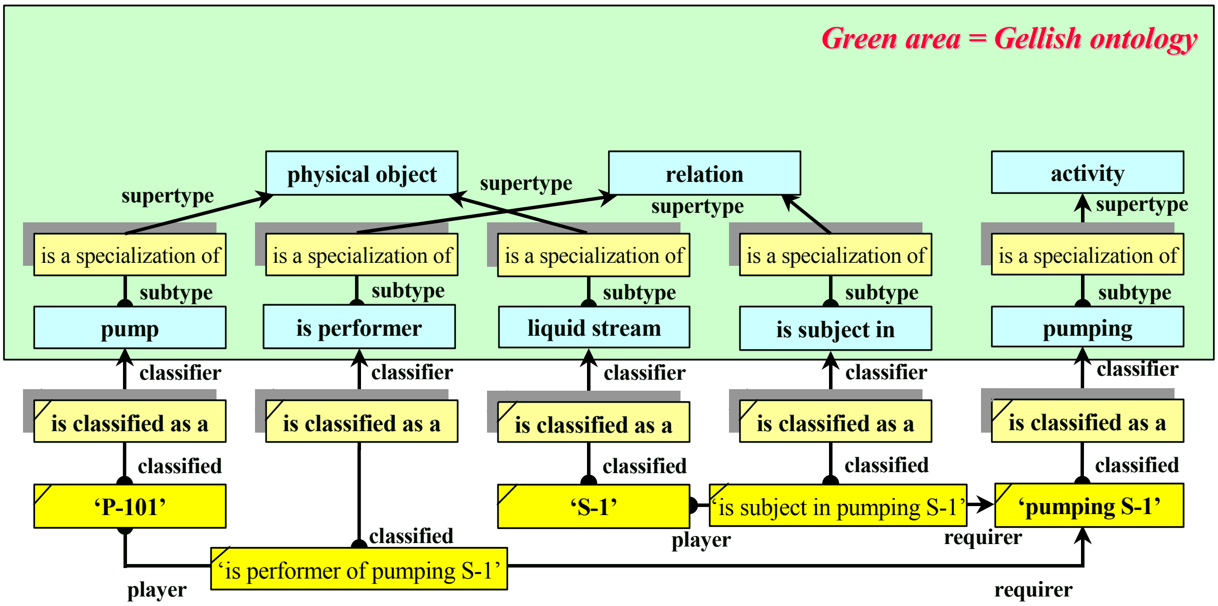


Figure 67, Relation of Gellish concepts to ISO 10303-211 or ISO 15926-2 data model entities

However, actually there is no need to use a data model at all, except for a single Gellish Table or equivalent structure that supports the 'basic semantic structure' that is described in section 2.4.2. Without the use of a data model it is still possible to interpret data expressed in Gellish, because the explicit classification relations provide interpretation rules for the expressions for which the relation types as well as the object types are defined in the Gellish dictionary itself. The only requirement is that all facts are expressed as instances of the 'basic semantic structure.

Figure 68 illustrates the complete definition of the concepts up to the concept called 'individual thing', which is defined as an element of the collection 'kinds of things'.

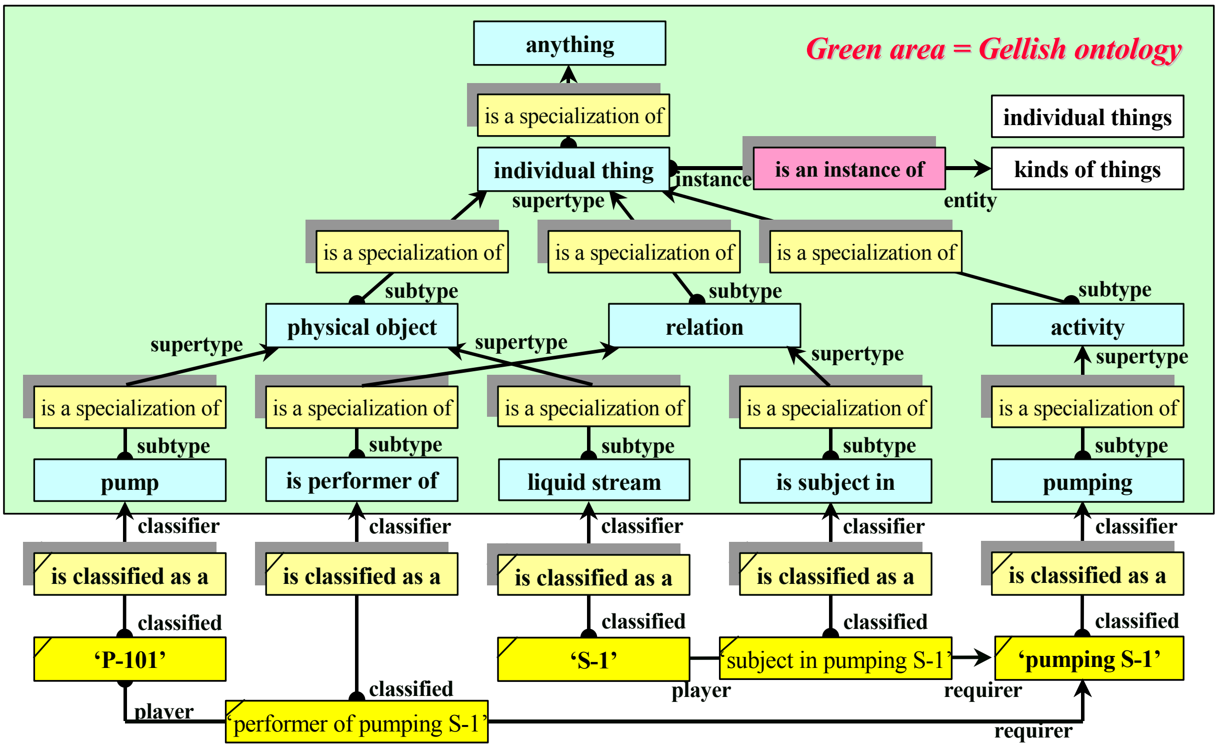


Figure 68, Instantiation in the 'basic semantic structure'

Figure 68 shows eight "is a specialization of" relations, each of which is a distinct relation between kinds of things. Similarly to what is described above about the "is classified as a" relation, this illustrates that the term 'is a specialization of' is not the name of each of those relations, but it is a name of the Gellish concept that qualifies those relations.

So, we distinguish between the various particular specialization relations and the 'is a specialization of' concept that is used to qualify those particular relations. Similarly we distinguish between the various particular classification (or 'conceptualization') relations for the classification of individual things and the 'is classified as a' relation concept that is used to classify those individual relations.

This more detailed definition of the semantics of a relation is illustrated in Figure 69.

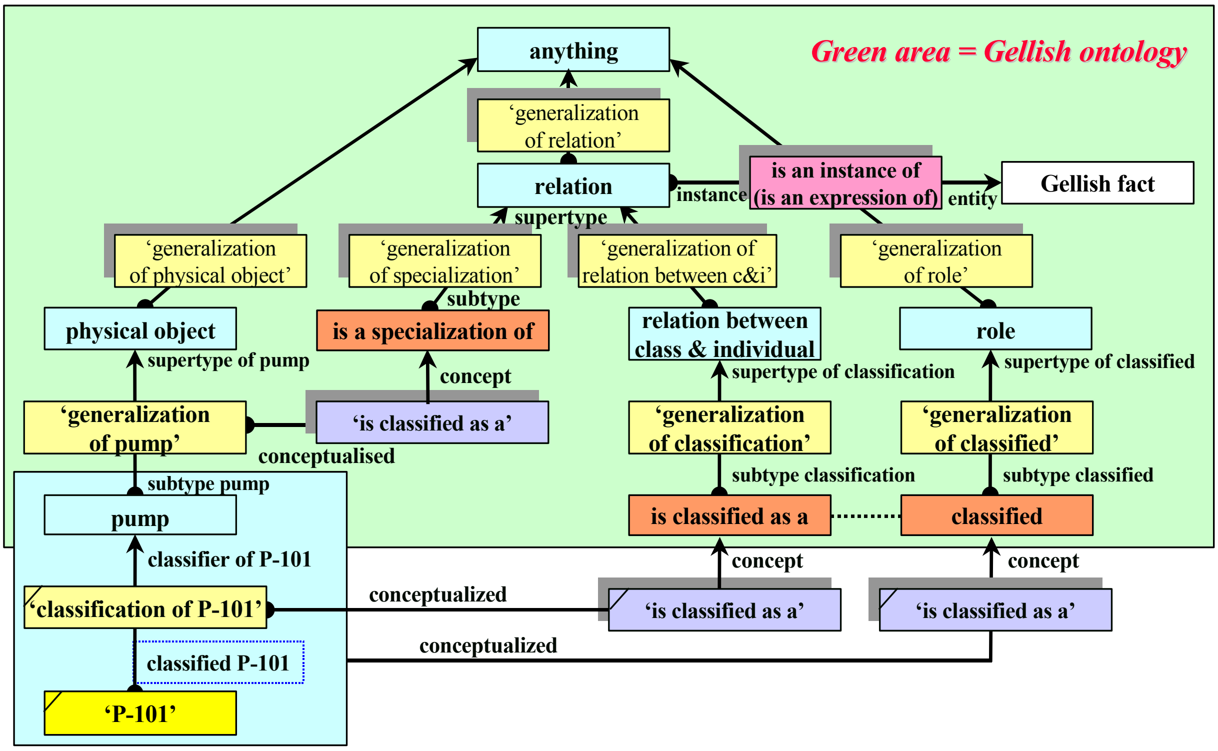


Figure 69, Particular conceptualization relations that classify relations and roles

Note that a conceptualization relation ('is conceptualized as a' relations) is a subtype of a classification relation, which is one of the basic semantic axioms. Those basic semantic axiom relations don't need to be conceptualized themselves, otherwise recursive definitions would occur.

4.4.2.2 Expression of knowledge in a Gellish knowledge base

Transformation of the expression of knowledge that is usually contained in data models to a Gellish database makes such knowledge accessible to end users and enables them to enhance that knowledge, whereas that knowledge is inherited via the specialization hierarchy and can be used in further Gellish expression. This is illustrated in Figure 70. The knowledge that a pump can be performer of a pumping activity is represented by a relation between the concepts 'pump' and 'pumping' ('to pump'). That relation is defined as a specialization relation of the 'can be a performer of a' relation. Etc.

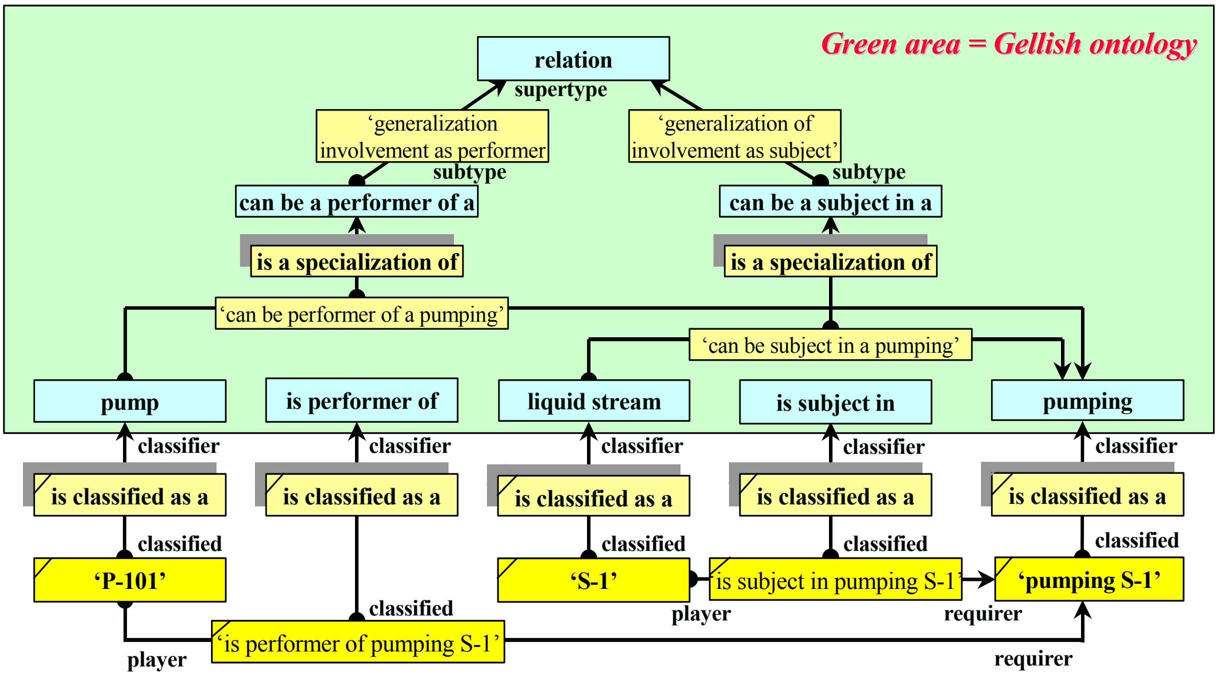


Figure 70, Modeling knowledge in a Gellish knowledge base

A significant number of such relations is included in the Gellish knowledge base. It is the intention that the Gellish knowledge base will be further extended with additional public domain 'open source' knowledge. It can also be extended privately with proprietary knowledge, including the extension with kinds of relations that extend the grammar of the Gellish language.

5 Implementation of Gellish in a single table

5.1 The Gellish Table

During implementations of the Gellish language, it appeared that information expressed in the Gellish language can be represented in one single generic table structure, which is called a Gellish Table. Therefore a standard Gellish Table is developed as a standard implementation method for information expressed in Gellish. This table defines the syntax or word ordering of the Gellish language. A single table has the advantage that the basic algorithms for searching and other data manipulation become very simple and no table joints are required. On the other hand new methods are required to express complicated queries, although the extensive number of kinds of relations enable to simplify many kinds of queries. A common implementation method has the advantage that data exchange files all have the same standard table structure, so that various software systems can write, read and interpret those files.

This section provides a definition of that Gellish Table and its subsets. The section describes the general table structure and the meaning of the columns and the relations between the columns, which comply with kinds of relations in the Gellish language. Data exchange files stored as a Gellish Table shall comply to one of the defined subsets. A database implementation may consist of one (or more) Gellish Tables, but may also deviate from that for example for performance reasons.

Implementation of Gellish either for storage of information in databases or for exchange of information in data exchange files or messages needs to be compliant to some database or file format, such as a physical data model (table definitions) or an .XLS or .DOC file format, etc., in order to allow software that recognizes the format to display the content of the database or file or message in a proper way. However, for the interpretation of the *meaning* of the content and for the automated processing of it, it is also required that the data (concepts, terms and structure) complies with the definition of the Gellish language as it not only defines the meaning of concepts and relations (equivalent to the definition of table columns and relations between them in databases), but also defines the meaning of terms and concepts that form the user data. Therefore Gellish includes natural language terminology and concepts in a formalized structure that allows for synonyms and homonyms. The grammar and syntax of natural languages, such as that of English, are not suitable for use by computers, because they are so flexible that computers cannot interpret them for the time being. Therefore, Gellish is an artificial language (actually only an artificial grammar and syntax or data structure, combined with natural language concepts and terminology) that is intended to cover a wide range of messages and that enables computers to unambiguously interpret the meaning of Gellish databases, files and messages.

Most other solutions to this requirement typically consist either of a dedicated ‘interface definition’ for a specific purpose communication between two or more systems or the communication is limited to a set of predefined message types, such as is the case for EDI messages.

There is one Gellish Table defined that has the capability to hold any expression of information in the Gellish language. On top of that a number of standard subset tables are defined with less table columns, intended for cases where a limited scope of data storage or exchange does not require the full table.

A Gellish Table can be implemented in several standard system independent *Gellish formats* that define the syntax for the presentation of data in the Gellish language:

1. The **Gellish Table format** (**GTF**).
   This is a Gellish Table implemented in MS-Excel format (XLS), or any SQL-based database table (e.g. MS-Access, Oracle or DB2), intended for computer-computer communication, but which is also human readable.

2. The **Gellish Tab delimited ASCII file format** (**GAS**),
   or the **Gellish Unicode file format** (**GUC**).
   These are character string representations of a Gellish Table, where fields are separated by a tab as a delimiter and the characters are encoded either in ASCII code or in Unicode. The table formats can be created by importing a Gellish Table in MS-Excel and saving the file in the appropriate format.
3. The **Gellish STEPfile format (G21).**
   This is an ISO 10303-21 implementation format of the Gellish Table.
4. The **Gellish XML format (GXL)**.
   This is an XML implementation of the Gellish Table according to ISO 10303-28.

This section describes the *Gellish Table* and the implementation format GTF. The other formats are straightforward derivations from this one.

Database systems can implement the use of a Gellish Table by import or export of data in a Gellish Table either to and from an internal Gellish Table or they can distribute imported data over various internal tables and collect data from various tables to export them in a single Gellish Table. During such distribution and collection it is usually required to convert terminology and data structures due to the difference between internal data structures and terminology and those used in Gellish. For example, an SQL Gellish Table can be added to an Oracle database, extended with some administrative columns to keep track of the synchronization and conversion rules between the Gellish Table and the other database tables.

In addition to the tabular form of the Gellish language there is an *experimental Gellish Natural Language (GNL)* format defined. This format is intended for human-computer communication. It is described in the document "The Gellish Language for the Semantic Web" (Andries van Renssen, 2002). Further research is required to validate its application in practice.

Gellish standardizes the content of messages and also standardizes the form of messages, but those two standards are strictly separated. The Gellish Language definition defines the semantics (the meaning) irrespective of the presentation form. The Gellish Table format is one of the at least four ways in which that semantics can be presented.
The above four formats are equivalent. The meaning of all four ways of expression is identical and defined unambiguously by the semantics of Gellish.
This separation between form and content definition gives a freedom to choose the most appropriate form (e.g. some people prefer an XML form). This enables computer software to interpret and process a message content automatically in whatever form it is.

The Gellish dictionary / taxonomy and knowledge base itself is also documented as a Gellish Table. It consists of virtually one large Gellish Table, from which various subsets can be derived.

The *Gellish Browser* (also called the STEPlib Browser) is an example of a Gellish enabled system that supports the creation, import and export, validation and browsing of any data in a Gellish Table. That software is an example of software that can process any knowledge or product data that is expressed in a standard Gellish neutral format. Its excellent response time proves the implementability of the Gellish language in commercial software. Similar software could act as agent software components in the Semantic Web. Such a Browser can be used to search for product data in a Gellish database as well as to browse the Gellish Language constructs and the dictionary and knowledge in the Gellish definition database (STEPlib) or its private extensions.

5.2 Basics of the Gellish Table

5.2.1 Expression of facts

Natural language terms such as 'person', 'car', 'colour', 'red', etc., etc., are defined in the Gellish dictionary. Those terms are names of concepts in Gellish English, as well as in natural English. In Gellish, each concept is identified by a Unique Identifier (the Gellish UID) and the names of those concepts are translated into various languages. This results in variants, such as Gellish English, Gellish

Nederlands (Dutch), etc., which all share the same UID for the concept.
An example of a part of a Gellish Table that defines the names of the concept of a pump (130206) in three languages is given in Figure 71 (the numbers 54, 2 and 101 refer to standard columns that are described later in this document):

54	2	101
Language	**Gellish UID**	**Concept name**
English	130206	pump
Deutsch	130206	Pumpe
Nederlands	130206	pomp

Figure 71, Gellish UID's and names in different languages

The table segment defines two facts on each line. These two facts are expressed by two implicit relations between the columns, as follows:

- A relation between column 2 and column 101, which expresses that an instance in column 2 is named by the string that is an instance of column 101 at the same line.
- A relation between column 54 and column 101, which expresses that the language mentioned in column 54 is the language in which the string in column 101 is expressed.

Each user defined object used in a Gellish Table shall also have a Gellish Unique Identifier (the User Object Gellish UID). Each User Object Gellish UID shall be unique in a range outside the range reserved for base Gellish (default User Object UID's: numbers above 10 billion). If no data integration will take place the receiving party does not need agree on the range of used UID's. If data integration is required with data available at the receiving party, then the range shall be agreed between the communicating parties and the sending party shall ensure that the UID's of existing things shall be used if those things are referred to.

To enable computer interpretation Gellish requires that the meaning of the User Objects is defined in Gellish by relating new User Objects to existing standard Gellish concepts, using particular kinds of relations, together with the optional provision a textual description. For these definitions the distinction between *concepts* and *individual things* is relevant. Note that the terms *concept* and *kind of thing* are used as synonyms in Gellish, whereas *class* is a subtype of that. Gellish requires that:

- Each new *individual* User Object shall be related to at least one Gellish concept or common value UID by a *classification relation* ('is classified as a').
- Each new user defined *concept* (or kind of thing) shall be related to at least one Gellish concept UID by a *specialization relation* or by a *qualification relation*.

A **classification relation** indicates that the User Object is an individual object that is classified by a Gellish concept.
A **specialization relation** indicates that the User Object is a concept that is a specialization of a Gellish concept.
A **qualification relation** indicates that the User Object is a qualified concept (also called a common value) that is a qualification or quantification of a Gellish concept.

For example, Figure 72 provides a part of a Gellish Table that defines my cycle-pump, called P-1, as a computer interpretable object known in Gellish, making use of the concept of a cycle-pump, which is not yet defined in Gellish. Therefore it is required that a second line is added that relates the new concept cycle-pump to an existing Gellish concept, in this case 'pump'.

54	2	101	1	60	3	15	201	4
Language	**Left hand object UID**	**Left hand object name**	**Fact UID**	**Relation type UID**	**Relation type name**	**Right hand object UID**	**Right hand object name**	**Full definition**
English	10.000.001	P-1	11.000.001	1225	is classified as a	10.000.002	cycle-pump	
English	10.000.002	cycle-pump	11.000.002	1146	is a specialization of	130206	pump	intended to inflate cycle tires.

Figure 72, Definition of User Objects in a Gellish Table

Each line in a Gellish Table expresses a main fact and a number of auxiliary facts. The semantics of the facts is defined by the kinds of relations between the columns in a Gellish table as defined later in this document. The main facts on the various lines are identified by the UID's in column 1. The relations between the UID's in columns 2 and 15 define the main facts and relations between other columns define a number of auxiliary facts.

For example, line 1 in Figure 72 expresses the following facts:

1. The relations between an instance in column 54 and the strings in columns 101, 3, 201 and 4 indicate that the strings in those columns are expressions in English; more precisely: they are expressions defined in Gellish English.
2. User object 10.000.001 is an individual object with the name "P-1".
   The fact that it is an individual object is inferred from the relation type "is classified as a", because such a classification relation always relates an individual to a class. This can be inferred from the Gellish database, because it contains two relations (with UID's 1.003.840 and 1.003.573) that express the facts that a classification relation requires two roles, a "classified individual" and a "classifier of individual". It also contains two other relations that express that the role of 'classified' can be played by an individual object and the role of 'classifier' can be played by a class (actually by the class 'individual thing' or by one of its subtypes).
3. User object 11.000.001 is a fact expressed as a relation between 10.000.001 and 10.000.002.

   The fact that object 10.000.001 is related to object 10.000.002 is the "main atomic fact" on this line of the Gellish Table.
4. Fact 11.000.001 is classified by 1225, being a standard Gellish classification relation concept. This defines the meaning of the main atomic fact, being in this case that fact 11.000.001 is qualified as a classification relation.
5. Relation type 1225 has the Gellish name "is classified as".
   Note that this is already defined in the Gellish database (STEPlib) and therefore, in principle, the relation type name is superfluous in the Gellish Table. However, the name is required in a Gellish Table, to support human readability.
6. User object 10.000.002 is a class of thing and has the name "cycle-pump".
   It is a general rule in Gellish that a name is formally allocated to an object only at a left hand side on a line where the object is defined by a classification, specialization or qualification relation or where the name is defined as a synonym or translation of an existing object name. On other lines and on the right hand sides a name is always only mentioned for human readability. Therefore, the fact that object 10.000.002 is called "cycle-pump" is formally defined on the next line in the above example Gellish Table and is referred to on this line only (a verification of consistency between the names on their various places is recommended).

Line 2 is required to ensure that the right hand term of line 1 is a defined object. Line 2 in Figure 72 defines similar facts. Note the following:

1. User Object 10.000.002 is a class with the name “cycle-pump”.
2. User Object 10.000.002 is a specialization of the existing Gellish object UID 130.206 with the name “pump”. From the Gellish database it can be inferred that the specialization relation is a relation with two roles: a subtype and a supertype, each of which is played by a class. So, both the left hand object and the right hand object is a class, which is consistent with the fact that the right hand object of line 1 is a class.
3. User Object 10.000.002 has an additional relation with the textual definition in column 4, which defines in what respect a cycle-pump distinguishes itself from the general concept of a pump and from its ‘brother’ types of pumps.

Similar fact can be described in Gellish by usage of other types of relations as defined in the Gellish database.

5.2.2 Atomic facts

Gellish principle 1: Every fact can be built up from elementary or atomic facts and each elementary or atomic fact is expressed as a relation between two things.

Because of this principle, each line in a Gellish Table contains one “main” atomic fact and various auxiliary atomic facts as is already illustrated by the facts that were derived from the content of Figure 72.
The expression of a main atomic fact only has the general structure as given in Figure 73:

101	**3**	**201**
Left hand object name	**Relation type name**	**Right hand object name**
thing-1	is related to	thing-2

Figure 73, General structure of the expression of an atomic fact

Gellish principle 2: A relation is in fact not a relation between object *names*, but between the objects themselves, whereas the Gellish UID’s represent the objects themselves.

Gellish principle 3: The meaning of a fact shall be indicated by a classification of the relation that represents the fact, with an earlier defined class. That earlier defined class shall be defined as a subtype of an earlier defined Gellish relation type (by a specialization relation).
In other words: each fact UID shall be classified by a subtype of ‘relation’ (UID 2850).

Because of principle 2, the main atomic fact is expressed in Gellish as the relation between the Gellish UID’s. Because of principle 3 column 60 is added to provide the class that classifies the main fact.

The result is a Gellish Table as presented in Figure 74.

2	**1**	**60**	**15**
Left hand object UID	**Fact UID**	**Relation type UID**	**Right hand object UID**
10.000.001	11.000.001	2850	10.000.002

Figure 74, Formal structure of the expression of a main fact

A Gellish Table is intended for computer-computer communication, but for debugging a human being should be able to understand it quickly. Therefore a Gellish Table combines the content of Figure 74 and Figure 73 into a table such as presented in Figure 72.

The content of a Gellish Table, such as the one illustrated in Figure 72, can be interpreted by a computer, because the meaning of ‘is classified as’ and ‘is a specialization of’ as well as the meaning of the concept ‘pump’ are predefined in Gellish. In other words, Gellish defines all the necessary semantics. Based on that, the new specialization relation can add a private extension of Gellish by defining the concept of a “cycle-pump” and with that extension it was possible to define the individual object “P-1” as a cycle-pump in such a way that a receiver system can interpret the facts.

5.2.3 Implied roles made explicit

Each object that is involved in a relation plays a particular role in that relation. That role remains implicit in most Gellish relations. However for the definition of the concepts in Gellish the roles are made explicit (in the Upper Ontology part of the Gellish database) in order to define the validity of the kind of things that may play the roles required by a kind of relation.

Also in some other occasions it is useful to make those roles explicit, especially for the modeling of functions and activities and for modeling the roles of objects in those functions and activities (occurrences).
Roles can be made explicit by decomposing atomic facts into two or more elementary facts. An elementary fact is expressed as a relation between an object and a role played by that object or a relation between a relation or occurrence and a role required by that relation or occurrence.

Each binary atomic relation between two objects requires two roles played by those objects.
Required roles and played roles can be made explicit by replacing one line in a Gellish Table by four lines: two lines describe the roles required by the relation and the other two describe which objects play those roles.

For example, the concept of classification of an individual by a class is defined in the Upper Ontology part of the Gellish database. That concept is one atomic fact that is defined by four elementary facts on four lines in the Gellish Table as follows:

2	**101**	**1**	**60**	**3**	**15**	**201**
Left hand object id	**Left hand object name**	**Fact UID**	**Relation type id**	**Relation type name**	**Right hand object id**	**Right hand object name**
730.067	individual object	1.001.423	4.714	can have a role as a	3.821	classified individual
1.225	is classified as a	1.003.840	4.731	requires as first role a	3.821	classified individual
1.225	is classified as a	1.003.573	4.733	requires as role-2 a	3.822	classifier for individual
730.067	individual object	1.001.215	4.714	can have a role as a	3.822	classifier for individual

Figure 75, Elementary facts about roles in relations

Note 1: The right hand objects in Figure 75 are classes of roles. Therefore, they are defined in the Gellish database as subtypes of the class ‘role’ and thus they are part of the overall specialization hierarchy of classes.

Note 2: The first line in Figure 75 defines that *a member of the class* ‘individual object’ can have a role as ‘classified individual’, whereas the last line defines that *the class* ‘individual object’ (*or one of its subtypes*) can have a role as a ‘classifier for individual’. This follows from the fact that the relation ‘is classified as’ is defined as being a subtype of ‘relation between an individual and a class’.

In some applications (e.g. for specification of standard requirements) it is required to make roles in Gellish relations explicit. This means that the individual roles are recorded as separate objects and that additional lines are required for the classification of those roles[50].

[50] The recording of explicit roles could possibly be simplified by the definition of an Extended Gellish Table version that has six additional columns: a Role ID, a Class of Role ID and a Class of Role Name for the left hand object and the same for the right hand object in atomic facts. In practice it appears that there are only a few occasions where these roles are explicitly used, whereas even then the Gellish Table without those addition appears to be suitable to record the required information.

5.2.4 Modeling Products, Requirements and Knowledge

The same Gellish Table can be used to describe facts about individual things or occurrences, requirements for things or knowledge about things in general. The only differences mean that other standard relation types are used to classify the relations and that it is indicated in which context a fact is valid. Typically different categories of facts use phrases that starts with different words as follows:

- A fact about an individual thing is expressed by a relation type that starts with “is” or “has”.
- A requirement starts with “shall” and must indicate a validity context (in column 18).
- A fact that describes knowledge typically starts with “can have” or “can be”.

This is illustrated in Figure 76 below.

101	18	1	3	201
Left hand object name	**Validity context for main fact**	**Fact UID**	**Relation type name**	**Right hand object name**
I-1		1	is a part of	P-1
impeller	handover to operations	2	shall have as aspect a	diameter
centrifugal pump		3	can have as part a	pump impeller

Figure 76, Example of Product data, a Requirement and Knowledge

The example above illustrates three main facts. The first one states that a particular impeller is part of a particular pump. The second fact states that the information about an impeller that is handed over to operations shall include a diameter. The third fact describes the general knowledge that a centrifugal pump shall have at least one impeller. The minimum and maximum number of simultaneous instances are indicated by the cardinalities, but those columns are not shown in the above figure.

5.2.5 Questions and answers in Gellish

Gellish can distinguish questions, from answers and confirmations or denials in a dialogue by modeling the communication activities as separate occurrences. However, also without modeling the dialogue itself it is possible to model a question or query. Some of the possible kinds of questions are discussed below.

A first kind of question is:

Which one(s)?

The are questions such as: “what are the object(s) that have a particular type of relation with another object?” For example, the question expressed in Figure 77:

101	3	201
Left hand object name	**Relation type name**	**Right hand object name**
what	is classified as a	pump

Figure 77, A Gellish query

This question asks for the object(s) that have a relation of type “is classified as” with the object “pump”. This can be interpreted by a computer as a question, because in Gellish it is defined that the term “what” expresses a question to identify the object(s) that satisfy the relation type. Furthermore it is common logic that the inheritance rules define that the question: “what is classified as pump?” implies: “what is classified as pump or as one of its subtypes?”

Intelligent agent software should be able to automatically generate the answer as a list of pumps. Such a list could for example consist of one cycle-pump and two centrifugal pumps, all three being subtypes of "pump":

- P-1 is classified as a cycle-pump
- P-101 is classified as a centrifugal pump
- P-102 is classified as a centrifugal pump

Another kind of question expresses:

Is it the case?

Gellish can be used to express what is the case, but it can also be used to express a communicative intent. This includes communications that contains questions, answers, confirmations, denials, etc. The question whether or to what extent something is the case according to the author of a proposition can be expressed in Gellish by adding an "intention" to the expression. This indicates the extent to which the relation expresses what is the case or the status of the process to become the case. This is called the "intention" of the proposition. The intention is a quality of an expression that can express not only that a proposition is a *question*, but it can also express that it is a *confirmation*, a *probability*, etc. For example:

101	43	3	201
Left hand object name	**Intention**	**Relation type name**	**Right hand object name**
P-1	question	is classified as	pump
P-1	confirmation	is classified as	pump

Figure 78, Intentions of a proposition

This means that basically the same proposition can be used for different purposes in a communication. This implies that the above two lines express two different opinions about the same fact.
Other 'allowed values' for the intention can be found in the Gellish database as qualitative classes that are a qualification of 'intention' or one of its subtypes.

5.2.6 Properties and inheritance

Other questions are similar as the question:

Does P-1 have a mass?

Such questions can be answered using the knowledge that is contained in the Gellish database. For example, it contains a relation that expresses that:

101	3	201
Left hand object name	**Relation type name**	**Right hand object name**
material	can have as aspect a	mass

Figure 79, Example of an inherited fact

Note, that in Gellish the relation type '*can have as aspect a*' means 'conceptually has a', which emphasize that in practice quantified data values about properties are not always allocated to an object, even if the object has the property. For example a material always has a mass, but the phrase 'can have …' indicates that the numeric value may be present or not in a data set about a material.

All the subtypes of material, that have a specialization relation with 'material', inherit this fact! (Software that implements Gellish shall ensure that this is the case). In other words all classes that are element of the specialization hierarchy of "material" inherit that they conceptually have a mass. This fact does not imply the availability of a numeric value according to a mass scale. It does imply, however, that individual objects, such as "P-1" which is classified as pump, which is a specialization (subtype) of material, conceptually also has a mass.

There are two options to make use of this:

1. The property can be *allocated* to the item when the individual is created and classified as such (as a proposal by the software).
2. An individual object which has properties can be verified against the properties defined and inherited from the specialization hierarchy of classes.

5.3 Gellish Table subsets

The simplest option is to use the full Gellish Table with all the columns present and filled in. However, depending on the application, users may decide to use only one of the predefined standard subsets of the set of Gellish Table columns.
In the description below it is indicated which columns are optional.

So a Gellish Table can be compliant with:

- Subset Nomenclature
- Subset Dictionary
- Subset Taxonomy
- Subset Product Model
- Subset Business Model
- Subset Extended Table

These standard subsets are defined in the following paragraphs.

Strictly speaking the sequence of the columns in a Gellish table is irrelevant, as long as the columns are indicated by their column identifier. Furthermore, the optional columns can be ignored from the tables when agreed between exchange parties, thus defining ad hoc Gellish Table subsets, called *subsets Free Table*.
In subsets Free Table, the selection of columns as well as the sequence of the columns is free. However, to avoid misunderstanding, the formal standard table subsets require the presence of all columns for the chosen subset in the indicated sequence.

5.3.1 Subset: Nomenclature

The *Nomenclature* table is intended as common terminology, synonyms and translations. The subset contains a list of particular names of things or terms (typically names of concepts, but also names of individual objects such as countries and other standard geographical objects) and their unique identifier in the Gellish language, together with the name of the language in which the names are expressed.

54	2	101
Language	**Gellish UID**	**Name of thing**
English	130206	pump
Deutsch	130206	Pumpe
Nederlands	130206	pomp

Figure 80, Nomenclature subset core example

A full Gellish *Nomenclature* table subset, including columns for status and administration, consists of the following columns in the indicated sequence:
0, 54, 16, 2, 101, 1, 8, 67, 9, 10, 12, 13.

Typically, the language in which the name of thing is expressed is the same language in which the language is expressed. The language name is itself a name of a particular thing which has a unique identifier in Gellish and which can be expressed in various languages.

Implicitly this table defines a naming relation between the UID and the name. This main fact (with a UID in column 1) is of the type 'is called' (or 'is referenced as'). There is also an implicit auxiliary fact, which defines the language context in which the naming is done. This fact is of the type 'is presented in'. The subset also allows defining the discipline or language community (sub-culture) in which a name is defined (column 16).

Misspellings and a pointer to the correct spelling can also be recorded in the nomenclature table. Misspellings are indicated by a status (column 8) 'replaced' and the 'identifier of successor of main fact' (column 67) indicates the fact id that defines the correct spelling.

Preferred terms to be used in particular contexts can be indicated by the 'validity context of main fact' (column 18).
The following table is an example of the main columns in a nomenclature table.

54	16	2	101	1	8	67
Language	**Discipline**	**Gellish UID**	**Name of thing**	**Fact id**	**Status**	**ID of successor of main fact**
English	mechanical engineering	130206	pump	1	accepted	
Deutsch	Maschinenbau	130206	Pumpe	2	vorgestellt	
Nederlands	werktuigbouwkunde	130206	pompe	3	vervangen	4
Nederlands	werktuigbouwkunde	130206	pomp	4	geaccepteerd	

Figure 81, Nomenclature subset example

In addition to those main columns, the following table columns provide information about timing and origin of the main fact. These columns also appear in the other subsets.

9	10	12	13	68
Date of start of life	**Date of la test change**	**Originator of la test change**	**Reference**	**Subset**
21 March 2005	21 March 2005	Andries van Renssen	rotating equipment peers	facts about rotating equipment
21 March 2005	21 March 2005	Andries van Renssen	Europump	facts about rotating equipment

Figure 82, Timing and origin of main facts

5.3.2 Subset: Dictionary

The *Dictionary* table is intended to provide textual definitions of things, especially of concepts, as an addition to the taxonomy subset. This implies a relation between the thing and the text that defines the thing.

To support the readability of a Gellish table, the name of the thing that is defined and the discipline are repeated in the Dictionary table.

The following is an example of the main columns in a dictionary table.

54	2	101	1	4	8
Language	**Gellish UID**	**Name of thing**	**Fact id**	**Textual definition**	**Status**
English	130206	pump	5	is a rotating equipment item intended to increase pressure in a liquid.	accepted
Nederlands	130206	pomp	6	is een apparaat met roterende delen dat bedoeld is om de druk in een vloeistof te verhogen.	geaccep-teerd

Figure 83, Dictionary subset example

A full Gellish *Dictionary* table subset, including columns for status and administration, consists of the following columns in the indicated sequence:
0, 54, 16, 2, 101, 1, **4**, 14, 8, 67, 9, 10, 12, 13.

*Note, that the **bold** number(s) specify the extension relative to the previous subset.*

The above example illustrates that definitions for the same concept can be given in different languages.

Verbal (spoken) or pictorial definitions require a relation to a sound or picture (or combination of them). However the textual definition (column 4) is meant for a string in ASCII or Unicode only. Therefore, such other definitions are represented in a Gellish Table in subset 'Product model', as described below.

5.3.3 Subset: Taxonomy

The "*Taxonomy* table" is intended to provide a specialization hierarchy of concepts, also called a subtyping hierarchy (sometimes erroneously called a classification hierarchy). This implies that there are subtype-supertype relations between the concepts. A subtype concept is a specialization of a supertype concept. The inverse of that relation expresses the same fact in another way, namely that a supertype concept is a generalization of a subtype concept.

The following example illustrates the main columns in a taxonomy table.

54	16	2	101	1	15	201	8
Language	**Discipline**	**Left hand UID**	**Left hand object name**	**Fact id**	**Right hand UID**	**Right hand object name**	**Status**
English	Engineering	130206	pump	7	130227	rotating equipment item	accepted
Nederlands	Engineering	130206	pomp	7	130227	apparaat met roterende delen	duplikaat

Figure 84, Taxonomy subset example

A specialization relation implies that the subtype concept inherits all the aspects that are intrinsic to the supertype concept.

Note that the left hand object name and the right hand object name, as well as the language, are strictly speaking superfluous, but they are added to support the readability of the table. If they are ignored it becomes clear that the two lines in the above example define the same fact, which is the reason why the fact id's are identical and the status of the latter one is set at 'duplicate'.

A full Gellish "*Taxonomy* table" subset, including columns for status and administration, consists of the following columns in the indicated sequence:
0, 54, 16, 2, 101, 1, **15**, **201**, 14, 8, 67, 9, 10, 12, 13.

5.3.4 Subset: Product Model

The *Product Model* table is intended for use in practice of data exchange to describe individual objects (including occurrences) during their lifecycle as well as knowledge about classes of objects. This subset consists of the following columns in the indicated sequence:
0, 54, **71**, 16, 2, **44**, 101, 1, **60**, **3**, 15, **45**, 201, **65**, **4**, **66**, **7**, 14, 8, 67, 9, 10, 12, 13, **68**.

5.3.5 Subset: Business Model

The *Business Model* table is intended for use in practice of data exchange to describe propositions. This includes business communication about both designs (imaginary objects) as well as real world objects (observed individual objects) during their lifecycle and about enquiries, answers, orders, confirmations, etc. This table is a superset (indicated in **bold**) of the product model table, so it can also be used for knowledge about classes of objects.

This subset consists of the following columns in the indicated sequence:
0, 54, 71, 16, **39**, 2, 44, 101, **43**, **18**, 1, 60, 3, **42**, 15, 45, 201, 65, 4, 66, 7, 14, 8, 67, 9, 10, 12, 13, 68.

5.3.6 Subset: Extended Table

The *Extended Table* consists of all columns defined in the table in the next chapter. It is meant for use of the full capabilities of the Gellish language, including scientific applications.
This set consists of the following columns in the indicated sequence:
0, **69**, 54, 71, 16, **17**, **50**, **38**, 39, 2, **56**, 44, 101, 43, 18, 1, 60, 3, 42, **52**, 15, 45, **55**, 201, 65, 4, 66, 7, **70**, **20**, 14, 8, 67, 9, 10, 12, 13, **53**, 68.

5.4 The Gellish Table definition

5.4.1 Implementation considerations

The Gellish Table is defined by the definition of the meanings of its columns.
A Gellish Table can be expressed in the structure of some other format. For example a Gellish Table can be implemented as a spreadsheet table (.xls) or as an MS-Access database table (.mdb), an SQL table such as an Oracle table, a DB2 table or any other database table. It can also be implemented as an ASCII text file (.txt), provided that commas separate the fields and that all text in a field is enclosed by quotes (‘).

So, in all those cases the Gellish Table will be sent to another party as a file in a proprietary format! Except for the .txt file format. For example, it may be sent as an .mdb file, which is suitable for the MS-Access database software, but that requires that the receiving party possesses MS-Access software.
However, the Gellish Table File format (*GTF-format*) is a neutral (software independent) tabular file format. It is therefore recommended to implement it either as an ASCII or Unicode text file or as a spreadsheet, such as .XLS, which can be read by most other spreadsheet software as well or in ISO 10303-21 (STEPfile) or 10303-28 (XML).

5.4.2 The Gellish Table column definitions

5.4.2.1 The Gellish table definition (header)

Each Gellish Table File has in principle a table header as given in Figure 72, extended with additional columns as described in this chapter.
A Gellish table can consist either of a complete set of columns or of one of the pre-defined subsets of columns as described above.
Each column has a column ID and a column name and has a meaning as defined below.
Note that the presence of a value in a column field implies one or more relations with values in other columns as described below. *Those relations define the facts about the objects!*

Note: If the table is implemented in a spreadsheet or ASCII or Unicode file, then the table starts with a header of three lines, as follows:

- The first line is a free text header line.
- The second line contains the column ID’s which consists of standard numbers, although arbitrarily chosen. They allow the columns to be presented in a different sequence without loss of meaning (the numbers below correspond to those column ID’s).
- The third line contains human readable text in every column field with a short name of the column. This name is free text.

5.4.2.2 The Gellish table body column definitions.

The lines in a Gellish Table are independent of each other and thus the lines may be sorted in any sequence, without loss of semantics (meaning).

Each line (row) in the body of a Gellish Table (which in a spreadsheet starts on the fourth line) expresses a group of facts, which consists of a *main fact* and a number of *auxiliary facts*.

Main fact.
A main fact is expressed by a combination of the following three objects in the columns:

- A left hand object id (2), a fact id (1) and a right hand object id (15).

Prime auxiliary facts.
The prime auxiliary facts are expressed by the following pairs of objects (the third object that identifies the fact is left implicit, but should be made explicit in a database):

- The relation between the left hand object id (2) and the left hand object name (101).
- The relation between the right hand object id (15) and the right hand object name (201).

- The relation between the fact id (1) and the relation type id (60).
- The relation between the relation type id (60) and its name (3).

Secondary auxiliary facts.
The secondary auxiliary facts are expressed by the pairs of objects that form the context for the validity of the id’s and names for objects identified by their id’s:

- The relation between the main fact (1) and its validity context (18).
- The relation between the left hand object id (2) and its uniqueness context (17).
- The relation between the right hand object id (15) and its uniqueness context (52).
- The relation between the uniqueness context for the left hand name (16) and the relation between left hand object id and left hand object name (2, 101).
- The relation between the uniqueness context for the right hand name (55) and the relation between right hand object id and right hand object name (15, 201).

Ternary auxiliary facts.
Some ternary auxiliary facts as described in the table below.

Dependent on the type of main fact (the main relation and its relation type) slightly different auxiliary facts can be distinguished and thus slightly different conventions are used to fill in the fields on the line as indicated in the table below.

The table columns in a Gellish Table are defined as follows (the numbers correspond with the column ID’s):

0	**Presentation sequence key**	string. A presentation sequence key indicates a relative position in a list of lines. It is meant to support *sorting* the content of a Gellish table. It has no contribution to the meaning of the facts represented on the line. The presentation sequence does not effect the meaning of the lines. This column can be arbitrarily filled-in for use in a specific context.
69	**Unique language identifier**	integer. The unique identifier of the language in which the name of the left hand object (see column 101) and the name of the relation type (see column 3) and the status (see column 8) is spelled and, if present, in which the definition (see columns 63 and 4) is spelled. The language is a context for the validity of the referencing relation between the UID and the string that is the name.
54	**Name of language of left hand object name**	string. The name of the language of the left hand object name indicates the name of the language for which a UID is given in column 69 and that is a context for the name of the left hand object (see column 101) and the name of the relation type (see column 3). If the relation type name is not available in that language, it may be given in English. The allowed values for ‘language name’ are the names defined in STEPlib (or your private extension). Currently there are names of natural languages and of (artificial) programming languages. For example in STEPlib: - natural language is a conceptualization of English, French (francais), German (Deutsch), etc.
17	**Uniqueness context for left hand object id.**	string (optional). The uniqueness context for left hand object id provides the context within which the left hand object id, given in column 2, is a unique reference to something. The default context is 'Gellish'.

2	**Unique left hand object identifier** (UID-2)	integer. A unique left hand object identifier is the identifier of the main object about which the line defines a fact. That main fact is an association between two objects mentioned in column 2 and 15. The external identifier (name) of the object in column 2 can be given in column 56 with its text attribute in column 101 'name of left hand object'. A *UID* is an artificial sequence number, provided it is unique in a managed context. For example, the UID 4724 is a reference number of a telephone extension in the context of my company in The Hague. An identical number may refer to a different object in a different context, such as the extension with UID 4724 in the context of your company. The uniqueness context is given in column 16 (subject area). Such a context itself is defined on a separate line in a Gellish table. Note, that a fact represented by an association or relationship is also an object.
71	**Uniqueness context identifier for left hand object name** (UID-7)	integer (optional). The uniqueness context identifier for left hand object name provides the context within which the left hand object name in column 101 is a unique reference to the object id in column 2, in addition to the language context (see column 65 and 54). The context is superfluous (and is for human clarification only) on all lines other than lines with a specialization, a qualification or a classification relation, because only there the left hand objects, identified by their UID, are *defined* to have a name. If no context is given on a definition line, then the name for the left hand object is unique in the whole Gellish language (and no homonyms are possible).
16	**Uniqueness context name for left hand object name (subject area)**	string (optional). The uniqueness context name for left hand object name is the name for the uniqueness context of which the identifier is given in column 71. The name is optional (and is for human clarification only) because the context UID in column 71 shall be a reference to a context that is defined on another line, where its UID and name appears in columns 2 and 101respectivily.
50	**Unique plural fact identifier** (UID-4) - see figure 3.	integer. A unique plural fact id is a unique *identifier* of a *set of facts* as identified in column 1. This is intended to indicate a collection of facts that are identified by the above mentioned local unique fact identifiers (UID-1). A plural fact identifier is typically used as an *identifier* of a *(sub) template* or *view*. When a plural fact identifier is filled-in, it implies the existence of an inclusion relation (.. is an element of ..) between the main fact on this line identified in column number 1 and the set of facts identified in column number 50.
38	**Left hand object type name**	string (optional). An object type of the left hand object (with the UID in column 2) indicates the name of the entity type of the left hand object in a particular data model about which the line defines the main fact. This column is superfluous as it can be inferred via inheritance from the mapping of the appropriate object or its classifying class in the Gellish specialization hierarchy to the appropriate data model.
39	**Reality**	string (optional). The reality is a classification of the left hand object, being either *imaginary* or *materialized* (= real). This indicates that the object is either a product of the mind or an object whose existence is based in the physical world, either as natural or as artificial object. If not specified, then the reality shall be interpreted from the context or from a explicit classification fact. For example, during design a pump will be an imaginary (although realistic) object, when fabricated a pump will be a materialized object. Note that an object cannot be imaginary and materialized. An installation relation relates an imaginary object to a materialized object. Classes are always imaginary.

56 **Identifier of left hand term** (UID-6)
integer (optional).
The identifier of left hand term is the unique identifier *of the name* in column 101, which is a name of the object identified in column 2. It is the UID of the encoded information for which the text in column 101 forms the attribute. Typically, his column is left blank by people, but can be filled in by a computer. Note: in fact the string in column 101 itself can be used as its own identifier.

44 **Left hand object cardinalities**
integers or 'n' (optional).
For common associations between classes this column contains the *simultaneous cardinalities for the left hand object class*. This means that it indicates the minimum and maximum number of members of the class that can be associated with a member of the right hand object class at the same time.
The cardinalities may be specified by:

- a comma separated list of two integers that indicate the lower and upper limit cardinalities.

The upper limit may be the character 'n' to indicate that the upper limit is unlimited.

101 **Left hand object name**
string.
A 'name' of the object identified in column 2 and associated with it via an "is referenced as" association in a context referred to in column 54. For example, a tag name or some other code.
It is the attribute of the encoded information identified in column 56. When there is no ID filled-in in column 56, then the text is only present for easy human reference to an object. It facilitates readability when the lines are sorted in a different sequence later.
Nameless objects can exist, which implies that there is no instance in columns 56 and 101 for an object in column 2.

43 **Intention**
string (optional).
An intention indicates the extent to which the main fact is the case or is the case according to the author of a proposition. An intention includes also a level of truth. If a line expresses a proposition or communication fact, then the intention qualifies the proposition. If a line expresses a fact, then the intention indicates whether the relation of the type is *true* or *false*.
For example, the intention may indicate that a proposition is an affirmative *request (question), confirmation, promise, declination, statement, denial, probability or acceptance*.
Default = '*true*', which means a qualification by stating: this fact "is the case".

18 **Validity context for main fact**
string (optional).
The validity context for main fact provides the context within which the fact id, given in column 1, represents a valid fact. If not given, the fact is valid in all contexts.

1 **Unique identifier of main fact** (UID-1)
integer.
A unique main fact identifier is an identifier of the *main* fact that is represented on the line (such as an association or possession relationship). This main fact is of the type as indicated in column 3 'relation type name'.

60 **Relation type ID**
integer.
A relation type ID is unique ID for the class that qualifies the fact in column 1, whereas a name of the type of relation is given in Gellish in column 3.

3 **Relation type name (Gellish)**
string.
A *relation type name* (or *fact type name*) is a name of one of the subtypes of relation or class of relation expressed in Gellish English.

52	**Context name for right hand object id**	string (optional). A context name for right hand object id is a name that indicates where the object referenced by a UID in column 15 is defined. This can be an external source such as a reference data library. For example, Gellish is a standard context for all objects defined in STEPlib. Other contexts can be, for example, a company database system, an XML namespace, an identifier of a template, or '*interface*' which indicates that the right hand object id is a dummy object identifier, that needs to be replaced by (matched with) an id of another object by an application.
15	**Right hand unique object identifier** (UID-3)	integer. A right hand unique object identifier is the UID of the object associated with the object in column 2. The name of this right hand object can (optionally) be given as right hand term in column 201. The name of an object that has a name is defined only on a line where the fact type indicates a referencing association to the object. On other lines a filled in name is only meant to support human readability.
45	**Right hand object cardinalities**	integers or 'n' (optional). For common associations between classes this column contains the *simultaneous cardinalities for the right hand object class*. This means that it indicates the minimum and maximum number of members of the class that can be associated with a member of the left hand object class at the same time. The cardinalities may be specified in the same way as for the left hand object.
55	**Uniqueness context for right hand object name**	string (optional). A uniqueness context for reference by right hand object is a context within which the right hand object id in column 15 with the name in column 201 forms a unique reference to the object in column 2. N.B. Only applicable when the association type in column 3 indicates a referencing association ('is referenced as') in the context of this uniqueness context. In other cases this column need not be filled in. Typically this context points to a library or language which contains the same object, but with a different UID or name as the left hand object.
42	**Description of main fact (template text)**	string (optional). A description of the main fact (column 1) is meant to be presented to a user. The text is intended as an aid for interpretation of the meaning of the main fact in its context and may imply an instruction to a user for what should be filled in as a value for the right hand term or what should be selected from a pick list in order to finalize a fact or group of facts. The text might appear on a user interface (e.g. a fill-in-the-blanks form or data sheet) and supports human understanding of the meaning of the fact(s) and the intention of the object in column 15 and 201 and optionally the UoM in column 7. For example: the text 'temperature of the fluid at inlet' suggests that a value and a unit of measure should be supplied.
201	**Right hand object name**	string. A right hand object name is a string or value which is a textual name of the object identified in column 15, and which is associated with the object in column 2 with a name in column 101. For example, a tag name or code, numeric value, class name or free text description.
65	**Partial description**	string (optional). A partial description is a description that together with the relation type name (column 3) and the right hand object name (column 201) forms a full definition as presented in column 4.

4 **Full definition**

string (optional).
A full definition is a textual description of the characteristics that identify the left hand object or members of the left hand object class. Typically this is a concatenation of the term “is a(n)”, the right hand object name and the text in column 63 (partial description).

66 **Unit of measure identifier**

integer (optional).
The unit of measure identifier identifies the scale used for interpretation of the numeric value of a property in column 201. In case column 201 contains a concept of property name, the indicated UoM UID in column 66 indicates the default.

7 **Unit of measure name (UoM)**

string (optional).
The unit of measure name is the name of the scale used for interpretation of the numeric value of a property in column 201. In case column 201 contains a concept of property name, the indicated UoM in column 7 is a name of the default.

70 **Picklist UID**

integer (optional).
The unique identifier for the collection of objects from which values for instances of the right hand term may be selected in the context of an instance of the left hand term.
Note, this column (together with column 20) is meant as a short-cut for subtyping a (right hand) aspect type in the context of the left hand object and adding an additional line which defines that the value for a subtype “shall be one of the” picklist collection of aspect values.
For example, model X shall have a colour from the list of “model X colours”
is a short cut for:

model X	shall have a	model X colour
model X colour	is a specialization of	colour

and

model X colour	shall be one of the	model X colours.

20 **Picklist name**

string (optional).
The name of a picklist or domain identified by the Picklist UID in column 70. The name of the picklist shall be unique in the same context as the context for the right hand term (column 201) as defined in column 16 on the line where the right hand term is defined and occurs as a left hand term.

14 **Remarks**

string (optional).
A remarks field is intended for comments related to the fact or the existence of the left hand object, its definition or status.
The remark is implicitly associated with the fact via: an ‘is described by’ relation with a string of text that has a role as a *remark*.

8 **Approval status of main fact**

string.
An approval status indicates the status of the main fact. The status of the other facts on a line can be derived from the status of the main fact. A status can be any of the qualifications of ‘approval status’ in STEPlib. For example: proposed, issue, deleted, proposed to be deleted, ignore, agreed, accepted, accepted association (= only the main fact is accepted), or replaced (see also the ‘Gellish Extension manual’ or Guide on STEPlib’). The status ‘replaced’ indicates that the main fact is deleted and that a successing fact (see column 64) exists. The reason of the status may be clarified in the remarks column (see column 14).

67	**UID of successing fact**	integer (optional). The UID of the fact by which this line, and especially the main fact which UID is given in column 1, is replaced when the status in column 8 is "replaced". It indicates that there exists a succession relation between the two facts. Note: If the relation type is the last classification relation or specialization relation for the left hand object, then the life of the left hand object is terminated and replaced by the left hand object of the successing relation.
9	**Date of start of validity**	date/time, stored as a real value in the '1900 data system'. A date of start of life is the moment of the begin of the validity of the main fact. It is implicitly associated with the main fact via a "valid since" association.
10	**Date of latest change (end of validity)**	date/time, stored as a real value in the '1900 data system'. A date of latest change indicates the latest change of one of the auxiliary facts. If the status in column 8 is "deleted", "replaced" or "history", then the data of latest change indicates the moment of the end of the validity of the main fact. Then it is assumed to be related to the main fact by a "valid until" relation.
12	**Author of latest change**	string. The person who is the originator of the proposition or of the expression of the fact and who has (limited) responsibility for the content of the line; especially its latest change.
13	**Reference or Source**	string. The organization or position in an organization or the (part of) document that acts as the source or point of reference for the main fact.
53	**Line identifier** (UID-5)	integer (optional). A line id (UID-5) is intended indicate the collection of facts (or 'cloud' of facts) in which all the facts on one line in a Gellish Table are included.
68	**Subset name**	string (optional). The subset name indicates a collection of lines in a Gellish Table that are managed together and of which the line is an element. Typically it indicates an area of responsibility of a peer group. It may indicate a separate table or spreadsheet.

5.5 Gellish Table Format implementations

5.5.1 A Gellish Table Format (GTF)

A Gellish Table can be implemented directly in any tabular format.
For example it can be implemented in a spreadsheet or an SQL based database table, such as in XLS of MS-Excel, in MDB of MS-ACCESS or in an Oracle or DB2 database table.

And as such it can be exchanged.

5.5.2 The Gellish STEPfile format

The Gellish STEPfile format is a way to express the content of a Gellish table.

The Gellish STEPfile format defines a Gellish table in a form that is compliant with the STEP physical file standard (ISO 10303-21), also called a "part 21" file format. A file in this format is indicated by file extension '.G21'.

ISO 10303-21 requires that the entities that are instantiated in a STEP compliant file are defined in a Gellish data model, written in EXPRESS (ISO 10303-11).

5.5.2.1 Gellish subset Product Model data model

The **Gellish data model for subset 'Product Model'** as defined in EXPRESS is presented in the third column of Figure 85.

		SCHEMA Gellish_Table_Format_subset_Product_Model;
		ENTITY gellish_fact;
0	Sequence	presentation_sequence_key: OPTIONAL string;
54	Language	language_name: string;
71	LHContextUID	context_UID_for_left_hand_object_name: OPTIONAL string;
16	LHContextName	context_name_for_left_hand_name: OPTIONAL string;
2	LHObjectUID	left_hand_object_UID: integer;
44	LHCardinalities	left_hand_cardinalities: OPTIONAL LIST(2) of integer;
101	LHObjectName	left_hand_object_name: string;
1	FactUID	fact_UID: integer;
60	RelTypeUID	relation_type_UID: integer;
3	RelTypeName	relation_type_name: string;
15	RHObjectUID	right_hand_object_UID: integer;
45	RHCardinalities	right_hand_cardinalities: OPTIONAL LIST(2) of integer;
201	RHObjectName	right_hand_object_name: string;
65	PartialDefinition	definition: OPTIONAL string;
4	FullDefinition	full_definition: OPTIONAL string;
66	UoMUID	uom_UID: OPTIONAL integer;
7	UoMName	uom_name: OPTIONAL string;
14	Remarks	remarks: OPTIONAL string;
8	ApprovalStatus	status: string;
67	SuccessorUID	successor_of_fact_UID: OPTIONAL integer;
9	EffectiveFrom	date_of_creation: real;
10	LatestUpdate	date_of_latest_change: real;
12	Author	originator_of_change: string;
13	Reference	source: string;
68	Subset	subset_table: OPTIONAL string;
		UNIQUE ur1: fact_UID; ur2: left_hand_object_name, right_hand_object_name, relation_type_name;
		END_ENTITY;
		END_SCHEMA;

Figure 85, The Gellish subset Product_Model data model in EXPRESS

The first column in the figure refers to the column number in a Gellish table.

The second column provides standard column names for database implementations.

A row in a Gellish table corresponds directly with an instance of this "gelish_fact" entity.

The following example is an illustration of the body of a G21 file in ISO standard format for subset Product Model. The fact expresses that P-101 is classified as a centrifugal pump.

#1 gellish_fact(,'english',,'project A',10000001,,,'P-101',11000001,'is classified as',
130058,,,'centrifugal pump',,,,,,'accepted',,20Feb2003,20Feb2003,'AvR','AvR',)

When this is represented in a Gellish Table, not showing the empty columns and the last four columns, it becomes:

54	**16**	**2**	**101**	**1**	**3**	**15**	**201**	**8**
Language	**Context**	**Left hand UID**	**Left hand object**	**Fact UID**	**Relation type name**	**Right hand UID**	**Right hand object**	**Status**
English	project A	100000001	P-101	11000001	is classified as a	130058	centrifugal pump	accepted

5.5.2.2 Gellish subset Business Model data model

The **Gellish data model for subset Business Model** as defined in EXPRESS is presented in the second column of Figure 86.

		SCHEMA Gellish_Table_Format_subset_Business_Model;
		ENTITY gellish_fact;
0	Sequence	presentation_sequence_key: OPTIONAL string;
54	Language	language_name: string;
71	LHContextUID	context_UID_for_left_hand_object_name: OPTIONAL string;
16	LHContextName	context_name_for_left_hand_name: OPTIONAL string;
39	LHReality	reality_of_left_hand_object: OPTIONAL string;
2	LHObjectUID	left_hand_object_UID: integer;
44	LHCardinalities	left_hand_cardinalities: OPTIONAL LIST(2) of integer;
101	LHObjectName	left_hand_object_name: string;
43	Intention	intention: OPTIONAL string;
18	ValidityContext	validity_context_name: string;
1	FactUID	fact_UID: integer;
60	RelTypeUID	relation_type_UID: integer;
3	RelTypeName	relation_type_name: string;
42	FactDescription	description_of_main_fact: OPTIONAL string;
15	RHObjectUID	right_hand_object_UID: integer;
45	RHCardinalities	right_hand_cardinalities: OPTIONAL LIST(2) of integer;
201	RHObjectName	right_hand_object_name: string;
65	PartialDefinition	definition: OPTIONAL string;
4	FullDefinition	full_definition: OPTIONAL string;
66	UoMUID	uom_UID: OPTIONAL integer;
7	UoMName	uom_name: OPTIONAL string;
14	Remarks	remarks: OPTIONAL string;
8	ApprovalStatus	status: string;
67	SuccessorUID	successor_of_fact_UID: OPTIONAL integer;
9	EffectiveFrom	date_of_creation: real;
10	LatestUpdate	date_of_latest_change: real;
12	Author	originator_of_change: string;
13	Reference	source: string;
68	Subset	subset_table: OPTIONAL string;
		UNIQUE
		ur1: fact_UID;
		ur2: left_hand_object_name, right_hand_object_name, relation_type_name, intention, originator_of_change;
		END_ENTITY;
		END_SCHEMA;

Figure 86, The Gellish subset Business Model data model in EXPRESS

The second column provides standard column names for database implementations.

The second uniqueness rule expresses that a person can express a proposition only once. If somebody expresses the same proposition twice (e.g. on different days), then these expressions are considered to be the same proposition (with the same fact_UID).

The above example expressed as a STEP Physical File, compliant with GTF subset Business Model and ISO 10303-21 is as follows:

```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION((),'2;1');
FILE_NAME('gellish_table_format_subset_business_model','2003-05-02T23:18:26',('B.J.H. de Boer'),('TLO
Holland Controls b.v.'),'EXPRESS Data Manager version 20020107',$,$);
FILE_SCHEMA(('GELLISH_TABLE_FORMAT_SUBSET_BUSINESS_MODEL'));
ENDSEC;

DATA;
#1= GELLISH_FACT($,'english',$,'project A',$,10000001,$,'P-101',$,11000001,'is classified
as',$,130058,$,'centrifugal pump',$,$,$,$,$,'accepted',$,300000.,300000.,'AvR','AvR',$);
ENDSEC;

END-ISO-10303-21;
```

Figure 87, Header of a STEP physical file

5.5.2.3 Extended Gellish table data model

The **Gellish data model for subset Extended Model** is presented in Figure 88.

		SCHEMA Gellish_Table_Format_subset_Extended_Model;
		ENTITY extended_gellish_fact;
0	Sequence	presentation_sequence_key: OPTIONAL string;
69	LanguageUID	language_UID: OPTIONAL integer;
54	Language	language_name: string;
71	LHContextUID	context_UID_for_left_hand_object_name: OPTIONAL string;
16	LHContextName	context_name_for_left_hand_name: OPTIONAL string;
17	LHUniqueContext	uniqueness_context_left_UID: OPTIONAL string;
50	PluralFactUID	plural_fact_UID: OPTIONAL integer;
38	LHObjectType	left_hand_object_type: OPTIONAL string;
39	LHReality	reality_of_left_hand_object: OPTIONAL string;
2	LHObjectUID	left_hand_object_UID: integer;
56	LHTermUID	left_hand_term_UID: OPTIONAL integer;
44	LHCardinalities	left_hand_cardinalities: OPTIONAL LIST(2) of integer;
101	LHObjectName	left_hand_object_name: string;
43	Intention	intention: string;
18	ValidityContext	validity_context_name: string;
1	FactUID	fact_UID: integer;
60	RelTypeUID	relation_type_UID: OPTIONAL integer;
3	RelTypeName	relation_type_name: string;
52	RHUniqueContext	uniqueness_context_right_UID: OPTIONAL string;
15	RHObjectUID	right_hand_object_UID: integer;
45	RHCardinalities	right_hand_cardinalities: OPTIONAL LIST(2) of integer;
55	RHUnContextName	uniqueness_context_right_name: OPTIONAL string;
42	FactDescription	description_of_main_fact: OPTIONAL string;
201	RHObjectName	right_hand_object_name: string;
65	PartialDefinition	definition: OPTIONAL string;
4	FullDefinition	full_definition: OPTIONAL string;
66	UoMUID	uom_UID: OPTIONAL integer;
7	UoMName	uom_name: OPTIONAL string;

70	DomainUID	domain_UID: OPTIONAL integer;
20	DomainName	domain_name: OPTIONAL string;
14	Remarks	remarks: OPTIONAL string;
8	ApprovalStatus	status: string;
67	SuccessorUID	successor_of_fact_UID: OPTIONAL integer;
9	EffectiveFrom	date_of_creation: real;
10	LatestUpdate	date_of_latest_change: real;
12	Author	originator_of_change: string;
13	Reference	source: string;
53	LineUID	line_UID: OPTIONAL integer;
68	Subset	subset_table: OPTIONAL string; UNIQUE ur1: fact_UID; ur2: left_hand_object_name, right_hand_object_name, relation_type_name, intention, originator_of_change, date_of_creation; END_ENTITY; END_SCHEMA;

Figure 88, The Gellish subset Extended Model data model in EXPRESS

The second column provides standard column names for database implementations.

The second uniqueness rule expresses that a person can express a proposition more than once. If somebody expresses the same proposition twice (e.g. on different moments), then these expressions are considered to be different propositions (with different fact_UID).

5.5.3 The Gellish XML format (GXL)

To ensure a good quality, an XML representation of a Gellish Table shall be conform ISO 10303-28. This means that it is defined as a representation of a Gellish data model in EXPRESS as defined in the previous paragraph, although presented in XML, compliant with the conversion rules defined in ISO 10303 part 28.
The XSD specification for the presentation of a Gellish Table subset Business Model in XML was created as a proof of concept, as often people expect an advantage of an XML representation. However, it is questionable whether the expression of a Gellish Table in XML adds value.

The availability of an XSD schema means that an automated conversion procedure can convert a Gellish Table implemented for example as an Excel spreadsheet table (XLS) into an XML file. This can be implemented for example as an Excel Macro (written in Visual Basic).

5.6 A generic user interface for the Gellish language

Information systems usually present information in a complete different way as in natural languages. Usually they present information as values that are filled-in in predefined templates, often called windows, which are comparable with standard forms. A number of related windows with navigation capabilities between them are called a 'Graphical User Interface' (GUI) of the system. The GUI of an information system can be seen as the 'predefined' grammatical and syntactical structures of its 'language'. These structures or templates can be used to create a number of kinds of expressions, by varying the contents of the fields that can be filled-in. Those dedicated user interface windows have as advantage over a presentation in natural languages that they present dedicated cohesion and enable to quickly find information on a fixed location in the context of a large quantity of other relevant information. But the templates have as disadvantage that they fix the grammar and syntax of the 'language' in a limited number of structures. Furthermore, each information system has its own GUI, which means that each information system has its own limited number of kinds of expressions.

Because the Gellish language has a wide expression capability, it requires a generic user interface. A generic user interface based on a straight forward implementation of the atomic facts can provide such a generic view, in which anything can act as an 'object in focus', whereas all the things that are directly related to the object in focus are displayed. An example of such an atomic view is given in Figure 89 and Figure 90.

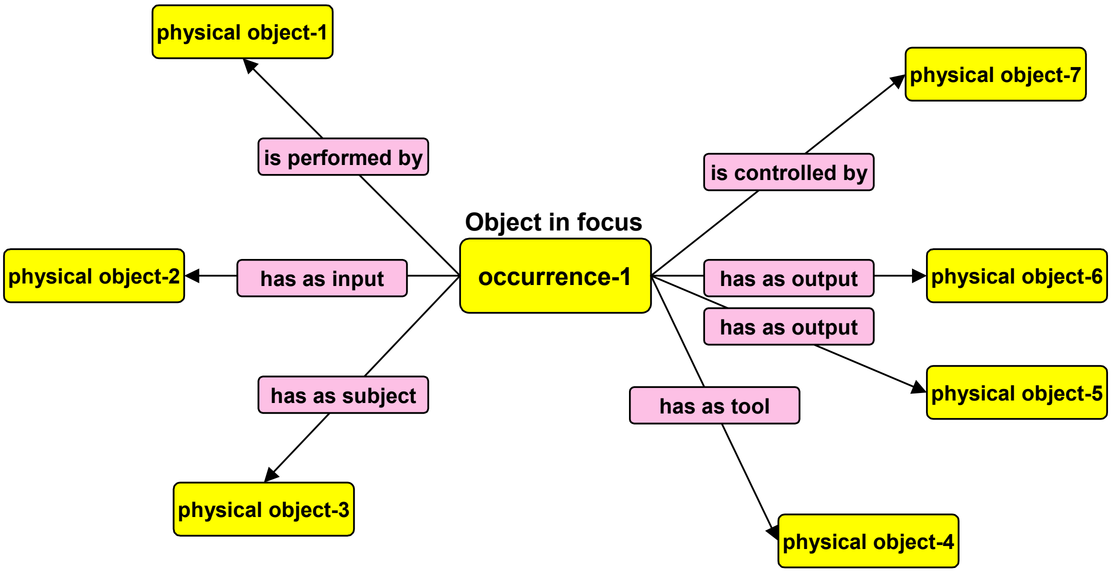


Figure 89, Example of a generic star user interface

A generic 'star' type presentation is sometimes suitable for knowledge presentation and navigation. To be a 'generic' user interface it is required that the pattern is generated from the source data by a tool and that it is simple to change the object in focus, for example by clicking on a displayed thing. However, an automated generation of the lay-out has the disadvantage that the patterns change when information is added or ignored and that the patterns are difficult to optimise for use interpretation. A presentation of expressions in Gellish in such a form only adds that the kinds of relations as well as the categories of knowledge are standardised. A disadvantage of a star presentation is that the amount of information that can be presented is limited and the user looses overview when the amount of information or the variety of relation types becomes high. This disadvantage is less apparent in a tabular presentation of the same content as is illustrated in Figure 90. This presentation allows that large quantities of information can be scrolled and a large variety of relation types can still be presented. If the user is able to select a view with only a subset of the relation types, a reasonable overview can be maintained.

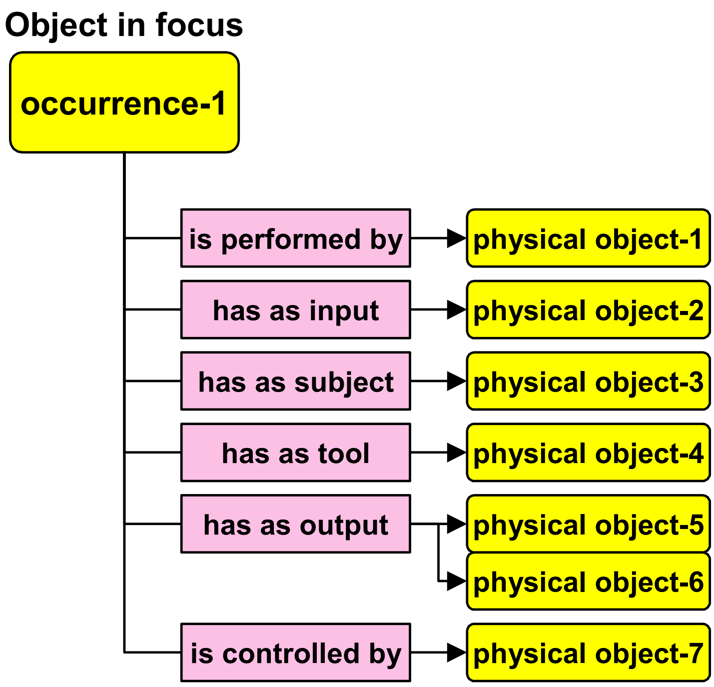


Figure 90, Example of a generic tabular atomic user interface

However, the atomic nature of these two atomic views requires navigation to related facts in order to see the things that are indirectly related to the object in focus to provide context for the thing in focus. Therefore, it would be beneficial if a generic user interface could be designed for the Gellish language that would have the same advantage of overview and accessibility of information that is provided by the dedicated user interfaces of existing information systems. Such a generic and yet practical user interface would significantly reduce the effort and costs that is currently required for development, training and maintenance of all the dedicated user interfaces of information systems.

This dilemma of generalization versus dedication might be solved by identification of a number of typical application scenario's. Candidate typical GUI components that will be generally applicable are for example:

- A product structure with properties.
- A matrix presentation with lists of aspects of a number of things, such as:
  - A hierarchy of subtypes of a kind of thing in focus.
  - A collection of elements with the aspects of each element in the collection.
  - A list of the individuals that are classified by a kind of thing in focus, with their aspects.
- A sequence (or network) of occurrences and intermediate 'streams' (inputs and outputs) and other involved objects.

An example of a typical 'product structure with properties' could include the following:

- Present the decomposition of the product in focus to a degree of decomposition to be specified by the user.
- Present all the aspects of the assembly and of each of its parts.
- Present all the aspects of the parts of the parts.
- Accompany the quantitative values by their unit of measure.

For example, a three levels product structure would presents the assembly, its parts and the parts of those parts, including their aspects could be presented in a generic user interface as illustrated in Figure 91.

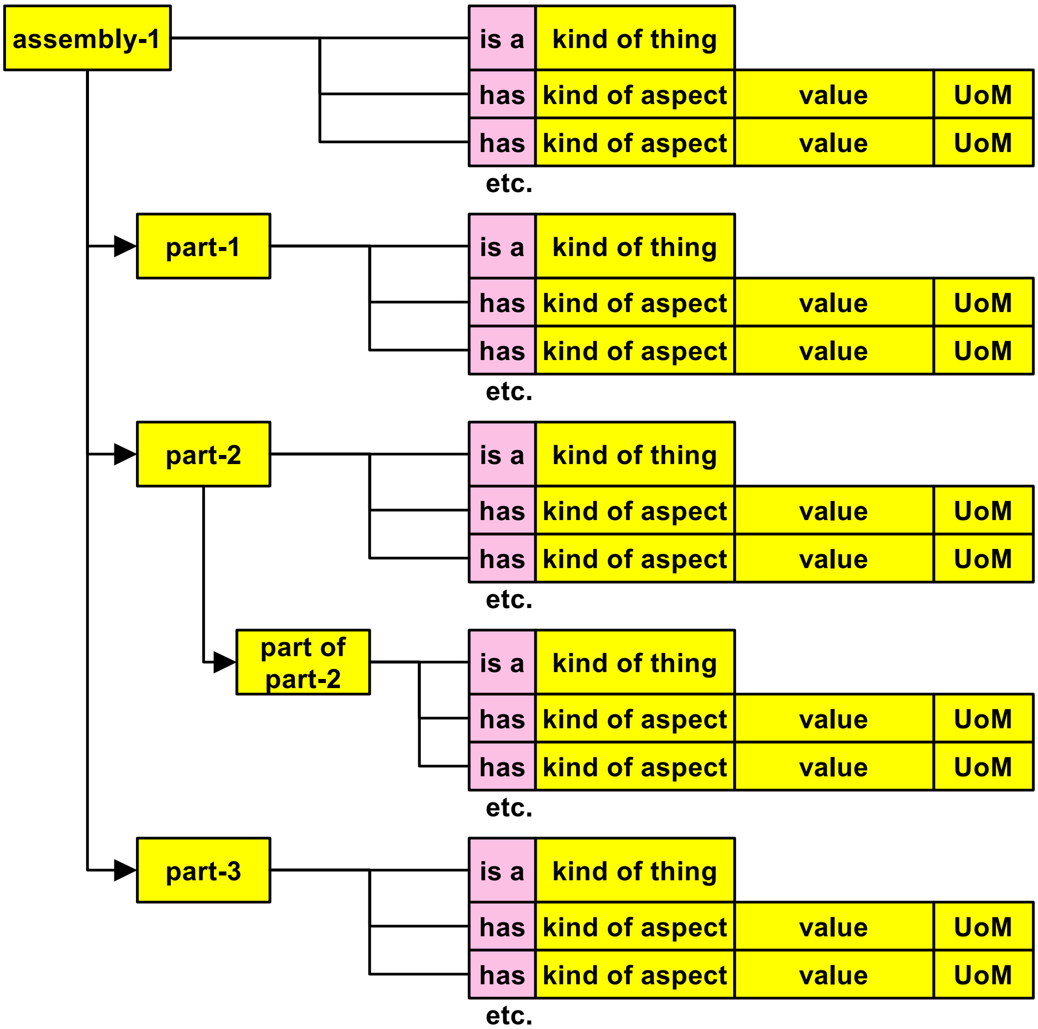


Figure 91, Generic user interface component for a product structure with its aspects

An application of this typical GUI for a product structure with aspects for a simple example could have a result as illustrated in Figure 92.

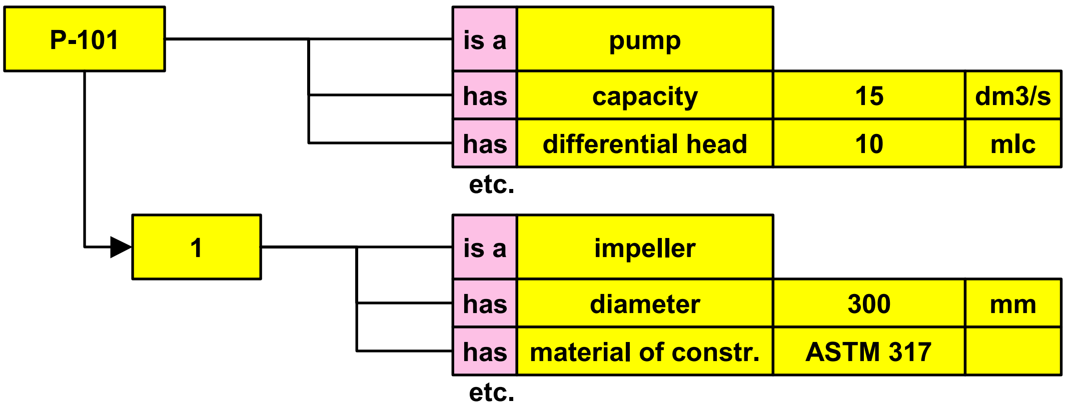


Figure 92, Example usage of a generic GUI component for a product structure

The above generic user interface component can allow for a varying number of decomposition levels as well as a varying number of aspects per part.

An example of a matrix type of generic user interface component is a subtype hierarchy of the types of things in a product type catalogue that are subtypes of a common supertype, as is illustrated in Figure 93.

ball valve	name	subtype	capacity	design pressure	body material
			dm3/s	bar	
	MESC 31.17.00.1	flanged ball valve	35	35	ASTM 316
	MESC 31.17.00.2	three way ball valve	10	10	ASTM 304
	MESC 31.17.00.3	eccentric ball valve	15	15	SS

Figure 93, Tabular generic GUI component used for catalogue items

The above tabular user interface component may seem trivial, but it should be noted that the presentation should be generated automatically from the source data. This means that the semantics of the column headers in relation to the rows and the cells in the matrix has to be interpreted from the semantics of the original data, as is explained in the earlier section 3.6.5 about lists and matrices.

A tabular generic GUI component can also be used to present aspects of a list of individual things that are classified by a common kind of thing in focus, such as a number of pumps as is illustrated in Figure 94.

pump	name	subtype	capacity	differential head	design pressure	material of construction
			dm3/s	mlc	bar	
	P-101	centrifugal pump	35	12	35	ASTM 316
	P-102	screw pump	10	5	10	ASTM 304
	P-103	reciprocating pump	15	40	15	SS

Figure 94, Tabular GUI component used to present individual objects of a kind

The above tabular GUI component content can be derived from the atomic facts the are expressed in Gellish (as a Gellish Table) by a relatively simple algorithm, because a direct transformation can be defined between an aspect of something and a cell in the above matrix. This transformation is based on the definition of a list as a collection of elements that are arranged in a particular sequence and the definition of a matrix as a list of lists, also arranged in a particular sequence. By specifying the components of those lists, and of the list of units of measure used for the quantification of the aspects, the semantics of the mapping is defined.

A similar tabular GUI component can be used for the presentation of a subtype hierarchy of types of things, for example document types, with information about each kind of thing, for example, their required format, their subject, their author, etc.

Further research on a generic user interface, possibly on the basis of this kind of typical GUI components, would be very beneficial, as it would greatly improve the acceptance of generic systems based on a generic language such as Gellish English. Otherwise the burden of transformations to dedicated user interfaces remains.

6 Application of the Gellish language

6.1 Knowledge models: An ontology of kinds of things

After the creation of the upper ontology of concepts and kinds of relations that express the knowledge required for the Gellish language definition, the ontology is extended with subtypes of the generic concepts, to form a full dictionary/taxonomy of concepts. Once those are available further knowledge about those concepts is added to the ontology, thus defining a Gellish knowledge base. The extension of the ontology is represented as a collection of kinds of facts and occurrences, kinds of totalities, kinds of aspects and kinds of roles that are required by the facts and occurrences and that can be played by the totalities and aspects.
This results in a knowledge base, which consists of a coherent hierarchical network of related things in which knowledge about the structure of the reality is expressed.

6.1.1 Product models of kinds of things

This section describes the usage of the Gellish language for the expression of knowledge about kinds of products, also called product models for kinds of things.

The definition of a concept or kind of thing in the previous chapter did not specify *how* each particular definition should be expressed. Each definition in an ordinary dictionary is provided as a string of text in a natural language. The structure of that text is only subject to the rules of the natural language and possibly by the rule that a definition should not be circular, although many definitions in ordinary dictionaries are nevertheless circular. Such textual definition explicitly specifies the obligatory aspects of things that belong to the kind and via the definition of the concepts whose names are used in the text it specifies implicitly other obligatory aspects. However, it is possible to structure the definition (the obligatory aspects things that are of the kind) as well as the optional aspects of things in the form of models of the kinds. Such a model records generally valid knowledge about the kind. Therefore, we call such a model a *knowledge model*. When such a model regards a model of a totality, then such a model is called a *product model of a kind*. An example of a knowledge model is a model of the concept 'centrifugal pump'. Such a knowledge model of a concept consists usually of a subset of all facts that are known about the concept. A knowledge model contains more than only the strict definition of the concept; it may contains also optional or typical aspects and aspect values, such as parts, shapes and sizes and thus allows for variations within the strict definition. There are kinds of aspects of which it is explicitly specified that members of the concept may have aspects of those kinds, but whether they do occur or not does not determine whether they are members of the concept.

A product model of a kind of thing consists of the collection of facts that essentially define the kind as well as the facts that are normally the case or are optionally the case, while leaving freedom for the unspecified facts.

An example of a defining fact about the kind 'car' could be the fact that a well-formed car can (and must) have three or more wheels. If the required minimum number of simultaneous facts of the kind (= the minimum simultaneous cardinality) is zero means that the fact is optional for the members of the kind. Thus that kind of fact does not belong to the definition of the kind. If the required minimum number of simultaneous facts of the kind is greater than zero, then it means that the fact is obligatory for a well-formed totality of the kind. It also means that the minimum number of facts of that kind belong to the definition of well-formedness for the kind. In the example of the definition of a car the minimum is three, so, it is a condition for that kind that a well-formed thing must have at least three wheels in order to possibly be a member of the kind. The fact that this only holds for well-formed things follows from the observation that a car which wheels are dismounted is still a car, although not a well-formed one. The definition of the concept well-formed therefore includes at least that things of the kind must be complete according to the constraints of the simultaneous number of required facts. A thing that is not well-formed must have only the absolute minimum number of facts that are required to be a member of the kind. In practice these minimum requirements are expressed in the textual definition of the kind. The facts that are modeled as generally valid facts therefore define the well-formed things and the constraints on their options. Those facts are also the most useful ones for practical applications.

An example of a part of a knowledge model of a lubrication oil system is presented in the following part of a Gellish Table.

2	101	1	60	3	15	201
Left hand object id	**Left hand object name**	**Fact id**	**Relation type id**	**Relation type name**	**Right hand object id**	**Right hand object name**
130,206	pump	11,000,223	1,191	can be a part of a	130,600	pump system
100,081	heat exchanger	1,000,003	1,191	can be a part of a	100,379	cooling system
10,132	strainer	11,000,013	1,191	can be a part of a	130,600	pump system
10,431	piping system	11,000,016	1,191	can be a part of a	100,086	heat transfer system
10,431	piping system	11,000,018	1,191	can be a part of a	130,600	pump system
10,671	change-over system	11,000,022	1,146	is a specialization of	10,431	piping system
71,305	flow control system	11,000,031	1,146	is a specialization of	70,760	control system
71,306	level control system	11,000,033	1,146	is a specialization of	70,760	control system
71,307	pressure control system	11,000,036	1,146	is a specialization of	70,760	control system
71,308	speed control system	11,000,038	1,146	is a specialization of	70,760	control system
71,309	temperature control system	11,000,040	1,146	is a specialization of	70,760	control system
71,314	pump control system	11,000,044	1,191	can be a part of a	130,600	pump system
130,102	driver	11,000,056	1,301	can be a role of a part of a	130,600	pump system
340,026	coupling	11,000,065	1,191	can be a part of a	130,600	pump system
521,016	collection vessel	11,000,068	1,146	is a specialization of	640,020	performer
521,016	collection vessel	11,000,069	4,714	can be a role of a	520,243	vessel
521,017	accumulator system	11,000,072	1,146	is a specialization of	370,534	storage system
130,600	pump system	1,000,001	1,191	can be a part of a	370,335	lubricating oil system
100,086	heat transfer system	1,000,002	1,191	can be a part of a	370,335	lubricating oil system
130,709	oil conditioner	1,000,004	1,191	can be a part of a	131,935	conditioner system
131,935	conditioner system	1,000,005	1,146	is a specialization of	730,014	system
370,190	filtration system	1,000,006	1,191	can be a part of a	370,335	lubricating oil system
131,935	conditioner system	1,000,007	1,191	can be a part of a	370,335	lubricating oil system
370,534	storage system	1,000,008	1,191	can be a part of a	370,335	lubricating oil system
10,036	filter	11,000,012	1,191	can be a part of a	370,190	filtration system
10,431	piping system	11,000,014	1,191	can be a part of a	40,045	drain system
10,431	piping system	11,000,015	1,191	can be a part of a	370,190	filtration system
10,431	piping system	11,000,017	1,191	can be a part of a	370,335	lubricating oil system
10,431	piping system	11,000,019	1,191	can be a part of a	370,534	storage system
10,671	change-over system	11,000,020	1,191	can be a part of a	370,190	filtration system
10,671	change-over system	11,000,021	1,191	can be a part of a	100,086	heat transfer system
40,045	drain system	11,000,023	1,191	can be a part of a	370,335	lubricating oil system
70,202	instrument	11,000,024	1,191	can be a part of a	70,760	control system
70,760	control system	11,000,029	1,191	can be a part of a	370,335	lubricating oil system
71,305	flow control system	11,000,030	1,191	can be a part of a	70,760	control system
71,306	level control system	11,000,032	1,191	can be a part of a	70,760	control system
71,306	level control system	11,000,034	1,191	can be a part of a	370,534	storage system
71,307	pressure control system	11,000,035	1,191	can be a part of a	70,760	control system
71,308	speed control system	11,000,037	1,191	can be a part of a	70,760	control system

71,309	temperature control system	11,000,039	1,191	can be a part of a	70,760	control system
71,311	breather cap	11,000,041	1,191	can be a part of a	370,534	storage system
71,312	filtration control system	11,000,042	1,191	can be a part of a	370,190	filtration system
71,313	heat transfer control system	11,000,043	1,191	can be a part of a	100,086	heat transfer system
100,081	heat exchanger	11,000,047	1,191	can be a part of a	100,086	heat transfer system
100,471	heating system	11,000,052	1,191	can be a part of a	370,534	storage system
130,218	reservoir	11,000,058	1,191	can be a part of a	370,534	storage system
130,486	connecting rod	11,000,060	1,191	can be a part of a	10,671	change-over system
370,023	auxiliary system	11,000,066	1,191	can be a part of a	370,335	lubricating oil system
521,016	collection vessel	11,000,067	1,301	can be a role of a part of a	40,045	drain system
521,017	accumulator system	11,000,070	1,191	can be a part of a	70,760	control system
521,017	accumulator system	11,000,071	1,191	can be a part of a	370,534	storage system
521,021	fill opening	11,000,073	1,191	can be a part of a	130,218	reservoir
820,020	control valve	11,000,074	1,191	can be a part of a	70,760	control system
820,905	three way valve	11,000,081	1,191	can be a part of a	10,671	change-over system

The above table illustrates the composition of a lubrication oil system and the composition of one of its sub-systems, a pump system. It also illustrates that there can be new subtype kinds of things defined ‘on the fly’ by the insertion of specialization relations that define new concepts as a specialization of an existing concept in the Gellish dictionary / taxonomy. Those new concepts can then immediately be used for the expression of knowledge (and constraints) about those subtypes. The cardinalities are not shown in this part of the table, but they specify for example that there may be 1 to 5 pumps in a pump system. These cardinality constraints are defined to be only valid in a particular context. Outside that context the general knowledge about possible compositions and roles of things is valid, often without cardinality constraints.

Note that the expression of such knowledge mainly relates concepts that are already present in the Gellish dictionary / taxonomy. Such knowledge can be used as a guide to create individual things of these kinds, or standard specifications (which are subtypes, because they also relate kinds of things, see the next paragraph). However, it is not a prerequisite that the knowledge models exist before standard specifications or individual product models can be defined. For example, it is allowed to express in the Gellish language that the individual pump P-107A is part of Pump system-107, even when it would not have been defined that a pump can be part of a pump system. A knowledge model actually is primarily an aid for a designer or a verification to identify exceptions on normal practice or best practice knowledge, just as ‘text book knowledge’ is not intended to constrain designers either.

6.2 Standard specification models – standard requirements or standard offerings

The Gellish language also enables the expression of standard requirements or standard offerings that either expresses definitions of kinds of things (specialized types), or that express *constraints on kinds of things which constraints are only valid within a particular context*. For example: when a particular standard specification is applicable to company X project handovers, then ‘project handover to company X’ (X handover) is the context for validity of the expressions that express the standard specifications. For the expression of such requirements, the Gellish relation type phrases begin with term ‘shall’. For example: in the context of X handover a compressor shall have as part a lubrication system and a value for its capacity.

Or in a Gellish Table:

- X handover compressor shall have as part a lubrication oil system

- X handover compressor shall have as aspect a capacity

Collections of these kinds of general requirements are called ‘standard specification models’. Examples of such models are standard buying descriptions, product catalogues and specifications expressed in best practice guides.

An example of a part of a standard specification model of a lubrication oil system is given in the following part of a Gellish Table.

2	101	1	60	3	15	201	66	7
Left hand object id	**Left hand object name**	**Fact id**	**Relation type id**	**Relation type name**	**Right hand object id**	**Right hand object name**	**UID of UoM**	**UoM**
70,405	temperature gauge	2,000,025	4,956	shall have as aspect a	553,027	dial size	570,423	mm
70,405	temperature gauge	2,000,026	4,956	shall have as aspect a	553,030	stem diameter	570,423	mm
70,405	temperature gauge	2,000,027	4,956	shall have as aspect a	553,029	stem insertion length	570,423	mm
70,405	temperature gauge	2,000,028	4,956	shall have as aspect a	550,273	temperature range	570,073	deg C
100,081	heat exchanger	2,000,045	5,017	shall be subject in a	192,505	manufacturing		
100,081	heat exchanger	2,000,046	4,956	shall have as aspect a	550,603	heat flow rate	570,418	kW
100,081	heat exchanger	2,000,048	4,994	shall be classified by a	492,032	model		
100,081	heat exchanger	2,000,049	4,995	shall be made of a	553,031	tube material		
100,081	heat exchanger	2,000,050	4,956	shall have as aspect a	553,034	tube side inlet diameter	570,411	inch
100,081	heat exchanger	2,000,051	4,956	shall have as aspect a	553,035	tube side outlet diameter	570,411	inch
130,030	bearing	2,000,053	4,994	shall be classified by a	3,818	subtype		
130,030	bearing	2,000,054	4,989	shall be a part of a	130,031	bearing assembly		
130,031	bearing assembly	2,000,055	4,989	shall be a part of a	130,652	triple screw pump		
130,206	pump	2,000,057	4,956	shall have as aspect a	551,564	capacity (mass flow rate)	570,043	kg/s
130,237	seal	2,000,059	4,989	shall be a part of a	130,652	triple screw pump		
130,575	oil reservoir	2,000,061	4,956	shall have as aspect a	553,037	free surface area	570,097	m2
130,575	oil reservoir	2,000,062	4,995	shall be made of a	552,856	material of construction		
130,575	oil reservoir	2,000,063	4,956	shall have as aspect a	551,021	retention time	570,422	min
130,575	oil reservoir	2,000,064	4,956	shall have as aspect a	553,036	volumetric charge capacity	570,403	dm3
820,020	control valve	2,000,075	4,956	shall have as aspect a	553,032	inlet diameter	570,411	inch
820,020	control valve	2,000,076	4,956	shall have as aspect a	552,743	liquid flow coefficient		
820,020	control valve	2,000,077	4,956	shall have as aspect a	553,033	outlet diameter	570,411	inch
820,020	control valve	2,000,078	4,956	shall have as aspect a	910,260	pressure rating	570,428	psi
820,020	control valve	2,000,079	4,956	shall have as aspect a	550,254	set pressure	570,056	bar
820,020	control valve	2,000,080	4,956	shall have as aspect a	551,034	size	570,411	inch
990,001	company	2,000,218	5,019	shall be performer of a	192,505	manufacturing		

The above table illustrates that in a particular context (of handover of data about delivered equipment between specified parties) it is specified that, for example, any temperature gauge shall have the listed aspects (with values). It also specifies for example, that a pump shall have as aspect a capacity (mass flow rate). This requirement is inherited by all the subtypes of pump, such as by triple screw pump, as well as by a horizontal triple screw pump and by a manufacturers model that is a subtype of that (such as an Allweiler - SN 210-46 model) as defined in the following paragraph. So, if for example individual thing P-107A is classified as an Allweiler - SN 210-46, then that individual P-107A shall have as aspect that is classified as a capacity (mass flow rate).

This is specified as follows:

- P-107A has as aspect C-1
- C1 is classified as a capacity (mass flow rate)
- C1 is quantified as equal to 5 kg/s

The Gellish language defines the semantics that a 'has as aspect' relation can be a realization of a 'can have as aspect a' or a 'shall have as aspect' relation. This enables that any individual product model (see the next paragraph) for which these specifications are applicable can now be verified against these criteria by software that can interpret both sets of Gellish expressions.

Note that the table also specifies in which unit of measure the values for the quantifiable aspects shall be expressed.

6.3 Individual product models or requirement models - using knowledge

This section describes the use of the Gellish language and the use of knowledge that is expressed in Gellish for the description of individual products or designs, their testing, operation and maintenance.

An example of a part of a product model of an individual lubrication oil system is given in the following part of a Gellish Table (to improve readability the fact id's are not shown).

2	101	60	3	15	201	66	7
Left hand object id	**Left hand object name**	**Relation type id**	**Relation type name**	**Right hand object id**	**Right hand object name**	**UID of UoM**	**UoM**
1,000,001	P-107A	1,190	is part of	1,000,002	Pump system 107		
1,000,001	P-107A	1,727	has aspect	1,000,007	absorbed power at rated duty of P-107A		
1,000,001	P-107A	1,225	is classified as a	1,000,041	Allweiler - SN 210-46		
1,000,001	P-107A	1,727	has aspect	1,000,009	capacity of P-107A		
1,000,001	P-107A	1,727	has aspect	1,000,013	discharge pressure of P-107A		
1,000,001	P-107A	1,225	is classified as a	150,147	horizontal triple screw pump		
1,000,001	P-107A	4,761	is performer of	1,000,051	pumping process 107		
1,000,002	Pump system 107	1,190	is part of	1,000,003	LO-100		
1,000,002	Pump system 107	1,225	is classified as a	130,600	pump system		
1,000,003	LO-100	1,225	is classified as a	370,335	lubricating oil system		
1,000,003	LO-100	1,369	is referred in	1,000,024	LO-100 outline drawing T-12346		
1,000,003	LO-100	4,753	shall be compliant with	491,939	API Std 614, fourth edition, April 1999		
1,000,003	LO-100	1,190	is part of	1,000,111	K-1301 compressor system		
1,000,003	LO-100	1,369	is referred in	1,000,062	LO-100 picture & P&ID		
1,000,003	LO-100	4,753	shall be compliant with	5,490,406	DEP 31.29.60.32-Gen.		
1,000,006	P-107A symbol	1,225	is classified as a	610,342	pump symbol		
1,000,006	P-107A symbol	1,713	refers to	1,000,001	P-107A		
1,000,006	P-107A symbol	1,269	is derived from	1,000,059	library pump symbol-1		
1,000,007	absorbed power at rated duty of P-107A	2,044	is quantified by	920,594	6.5	570,418	kW
1,000,007	absorbed power at rated	1,225	is classified as a	550,307	absorbed power at rated		

	duty of P-107A				duty		
1,000,008	Allweiler	1,225	is classified as a	990,001	company		
1,000,008	Allweiler	5,152	is manufacturer of	1,000,001	P-107A		
1,000,009	capacity of P-107A	2,044	is quantified by	921,473	242	570,554	dm3/ min
1,000,009	capacity of P-107A	1,225	is classified as a	550,430	rated capacity (volume flow rate)		
1,000,010	charge capacity of Res-112	2,044	is quantified by	922,240	1850	570,403	dm3
1,000,010	charge capacity of Res-112	1,225	is classified as a	553,036	volumetric charge capacity		
1,000,011	dial plate of TI-110	1,225	is classified as a	71,310	dial plate		
1,000,011	dial plate of TI-110	1,727	has aspect	1,000,057	diameter of dial plate of TI-110		
1,000,012	diameter of stem of TI-119	2,044	is quantified by	920,108	8	570,423	mm
1,000,012	diameter of stem of TI-119	1,225	is classified as a	550,188	diameter		
1,000,013	discharge pressure of P-107A	2,044	is quantified by	920,588	5.8	570,056	bar
1,000,013	discharge pressure of P-107A	1,225	is classified as a	550,580	discharge pressure		
1,000,014	free surface of Res-112	2,044	is quantified by	920,103	3	570,097	m2
1,000,014	free surface of Res-112	1,225	is classified as a	553,037	free surface area		
1,000,015	H-108	1,727	has aspect	1,000,016	heat exchange capacity of H-108	570,418	kW
1,000,015	H-108	1,190	is part of	1,000,052	Heat transfer system 108		
1,000,015	H-108	1,225	is classified as a	1,000,042	OKG25/305		
1,000,015	H-108	1,225	is classified as a	100,137	shell and tube heat exchanger		
1,000,015	H-108	1,190	is whole of	1,000,028	shell of H-108		
1,000,015	H-108	1,190	is whole of	1,000,038	tube of H-108		
1,000,015	H-108	1,190	is whole of	1,000,048	tube side inlet connection of H-108		
1,000,015	H-108	1,190	is whole of	1,000,049	tube side outlet connection of H-108		
1,000,015	H-108	1,190	is whole of	1,000,039	tubesheet of H-108		
1,000,015	H-108	1,727	has aspect	1,000,040	volume flow rate at shell side of H-108		
1,000,016	heat exchange capacity of H-108	2,044	is quantified by	920,303	50	570,418	kW
1,000,016	heat exchange capacity of H-108	1,225	is classified as a	550,603	heat flow rate		
1,000,017	insertion length of stem of TI-110	2,044	is quantified by	920,416	160	570,423	mm
1,000,017	insertion length of stem of TI-110	1,225	is classified as a	553,028	insertion length		
1,000,018	liquid flow coefficient of PCV-109	1,225	is classified as a	552,743	liquid flow coefficient		
1,000,018	liquid flow coefficient of PCV-109	2,044	is quantified by	920,540	0.5	570,068	-
1,000,019	material of construction of Res-112	5,020	is qualified as	370,124	316L stainless steel		
1,000,019	material of construction of Res-112	1,225	is classified as a	552,856	material of construction		
1,000,020	nominal diameter of inlet connection of H-108	2,044	is quantified by	920,103	3	570,411	inch

1,000,020	nominal diameter of inlet connection of H-108	1,225	is classified as a	551,563	nominal diameter		
1,000,021	nominal diameter of outlet connection of H-108	2,044	is quantified by	920,103	3	570,411	inch
1,000,021	nominal diameter of outlet connection of H-108	1,225	is classified as a	551,563	nominal diameter		
1,000,022	normal volume flow rate of S2	2,044	is quantified by	790,665	230	570,554	dm3/ min
1,000,022	normal volume flow rate of S2	1,225	is classified as a	551,233	normal operating volume flow rate		
1,000,023	Oeltechnik	1,225	is classified as a	990,001	company		
1,000,023	Oeltechnik	5,152	is manufacturer of	1,000,015	H-108		
1,000,024	LO-100 outline drawing T-12346	1,225	is classified as a	490,115	outline drawing		
1,000,024	LO-100 outline drawing T-12346	4,996	is presented on	1,000,035	T-1_Oil_mist_seperator-API_614-MAN_Turbo.pdf		
1,000,025	PCV-109	1,190	is whole of	1,000,043	body of PCV-109		
1,000,025	PCV-109	1,225	is classified as a	820,020	control valve		
1,000,025	PCV-109	1,727	has aspect	1,000,054	inlet diameter of PCV-109		
1,000,025	PCV-109	1,727	has aspect	1,000,018	liquid flow coefficient of PCV-109		
1,000,025	PCV-109	1,190	is part of	1,000,003	LO-100		
1,000,025	PCV-109	1,727	has aspect	1,000,055	outlet diameter of PCV-109		
1,000,025	PCV-109	1,727	has aspect	1,000,056	pressure rating of PCV-109		
1,000,025	PCV-109	1,727	has aspect	1,000,053	set pressure of PCV-109		
1,000,025	PCV-109	1,190	is whole of	1,000,047	trim of PCV-109		
1,000,026	Res-112	1,727	has aspect	1,000,010	charge capacity of Res-112		
1,000,026	Res-112	1,727	has aspect	1,000,014	free surface of Res-112		
1,000,026	Res-112	1,190	is part of	1,000,003	LO-100		
1,000,026	Res-112	4,794	is made of	1,000,019	material of construction of Res-112		
1,000,026	Res-112	1,225	is classified as a	130,575	oil reservoir		
1,000,026	Res-112	1,727	has aspect	1,000,027	retention time in Res-112		
1,000,027	retention time in Res-112	2,044	is quantified by	920,108	8	570,422	min
1,000,027	retention time in Res-112	1,225	is classified as a	551,021	retention time		
1,000,028	shell of H-108	4,794	is made of	280,043	carbon steel		
1,000,028	shell of H-108	1,225	is classified as a	520,204	shell		
1,000,029	size of body of PCV-109	2,044	is quantified by	920,103	3	570,411	inch
1,000,029	size of body of PCV-109	1,225	is classified as a	551,034	size		
1,000,030	size of trim of PCV-109	2,044	is quantified by	924,844	1 5/16	570,411	inch
1,000,030	size of trim of PCV-109	1,225	is classified as a	551,034	size		
1,000,031	splitting of S2	1,225	is classified as a	191,873	stream splitting		
1,000,032	stem of TI-110	1,727	has aspect	1,000,012	diameter of stem of TI-119		
1,000,032	stem of TI-110	1,727	has aspect	1,000,017	insertion length of stem of TI-110		
1,000,032	stem of TI-110	1,225	is classified as a	820,138	stem		
1,000,033	T-12345	1,225	is classified as a	490,183	P&ID		

1,000,033	T-12345	4,720	contains information about	1,000,001	P-107A		
1,000,033	T-12345	4,720	contains information about	1,000,003	LO-100		
1,000,033	T-12345	4,720	contains information about	1,000,015	H-108		
1,000,033	T-12345	4,720	contains information about	1,000,037	TI-110		
1,000,033	T-12345	4,720	contains information about	1,000,025	PCV-109		
1,000,033	T-12345	4,720	contains information about	1,000,002	Pump system 107		
1,000,034	simplified P&ID of LO-100.dwg	1,225	is classified as a	490,533	electronic data file		
1,000,034	simplified P&ID of LO-100.dwg	2,071	represents	1,000,033	T-12345		
1,000,034	simplified P&ID of LO-100.dwg	1,227	is element of	1,000,065	C:\BEYING		
1,000,035	T-1_Oil_mist_seperator-API_614-MAN_Turbo.pdf	1,225	is classified as a	490,533	electronic data file		
1,000,035	T-1_Oil_mist_seperator-API_614-MAN_Turbo.pdf	1,227	is element of	1,000,065	C:\BEYING		
1,000,036	temperature range of TI-110	2,044	is quantified by	920,053	100	570,073	deg C
1,000,036	temperature range of TI-110	1,225	is classified as a	5,520,273	temperature range		
1,000,037	TI-110	1,190	is whole of	1,000,011	dial plate of TI-110		
1,000,037	TI-110	1,190	is part of	1,000,003	LO-100		
1,000,037	TI-110	1,190	has as part	1,000,032	stem of TI-110		
1,000,037	TI-110	1,225	is classified as a	70,405	temperature gauge		
1,000,037	TI-110	1,727	has aspect	1,000,036	temperature range of TI-110		
1,000,038	tube of H-108	4,794	is made of	580,912	inhibited admiralty brass		
1,000,038	tube of H-108	1,225	is classified as a	100,167	tube		
1,000,039	tubesheet of H-108	4,794	is made of	580,913	naval brass		
1,000,039	tubesheet of H-108	1,225	is classified as a	100,168	tube sheet		
1,000,040	volume flow rate at shell side of H-108	2,044	is quantified by	920,426	170	570,554	dm3/min
1,000,040	volume flow rate at shell side of H-108	1,225	is classified as a	550,318	capacity (volume flow rate)		
1,000,041	Allweiler - SN 210-46	1,146	is a specialization of	150,147	horizontal triple screw pump		
1,000,042	OKG25/305	1,146	is a specialization of	100,081	heat exchanger		
1,000,043	body of PCV-109	1,225	is classified as a	520,281	body		
1,000,043	body of PCV-109	1,727	has aspect	1,000,029	size of body of PCV-109		
1,000,044	oil downstream of P-107 (S2)	1,727	has aspect	1,000,013	discharge pressure of P-107A		
1,000,044	oil downstream of P-107 (S2)	1,225	is classified as a	730,083	liquid stream		
1,000,044	oil downstream of P-107 (S2)	1,727	has aspect	1,000,022	normal volume flow rate of S2		
1,000,044	oil downstream of P-107 (S2)	4,786	is output of	1,000,051	pumping process 107		
1,000,044	oil downstream of P-107 (S2)	4,785	is input of	1,000,031	splitting of S2		

1,000,045	oil to H-108 (S3)	1,225	is classified as a	730,083	liquid stream		
1,000,045	oil to H-108 (S3)	4,786	is output of	1,000,031	splitting of S2		
1,000,046	oil upstream of P-107 (S1)	1,225	is classified as a	730,083	liquid stream		
1,000,046	oil upstream of P-107 (S1)	4,785	is input of	1,000,051	pumping process 107		
1,000,047	trim of PCV-109	1,727	has aspect	1,000,030	size of trim of PCV-109		
1,000,047	trim of PCV-109	1,225	is classified as a	71,315	trim		
1,000,048	tube side inlet connection of H-108	1,225	is classified as a	340,051	mechanical connection		
1,000,048	tube side inlet connection of H-108	1,727	has aspect	1,000,020	nominal diameter of inlet connection of H-108		
1,000,049	tube side outlet connection of H-108	1,225	is classified as a	340,051	mechanical connection		
1,000,049	tube side outlet connection of H-108	1,727	has aspect	1,000,021	nominal diameter of outlet connection of H-108		
1,000,050	K-1301 compressor	1,225	is classified as a	130,069	compressor		
1,000,050	K-1301 compressor	1,190	is part of	1,000,111	K-1301 compressor system		
1,000,051	pumping process 107	1,225	is classified as a	192,512	to pump		
1,000,052	Heat transfer system 108	1,225	is classified as a	100,086	heat transfer system		
1,000,052	Heat transfer system 108	1,190	is part of	1,000,003	LO-100		
1,000,053	set pressure of PCV-109	1,225	is classified as a	550,254	set pressure		
1,000,053	set pressure of PCV-109	2,044	is quantified by	920,577	4.6	570,393	barg
1,000,054	inlet diameter of PCV-109	1,225	is classified as a	553,032	inlet diameter		
1,000,054	inlet diameter of PCV-109	2,044	is quantified by	920,103	3	570,411	inch
1,000,055	outlet diameter of PCV-109	1,225	is classified as a	553,033	outlet diameter		
1,000,055	outlet diameter of PCV-109	2,044	is quantified by	920,103	3	570,411	inch
1,000,056	pressure rating of PCV-109	1,225	is classified as a	910,260	pressure rating		
1,000,056	pressure rating of PCV-109	2,044	is quantified by	920,406	150	570,428	psi
1,000,057	diameter of dial plate of TI-110	2,044	is quantified by	920,053	100	570,423	mm
1,000,057	diameter of dial plate of TI-110	1,225	is classified as a	550,188	diameter		
1,000,058	oil to PCV-109 (S4)	1,225	is classified as a	730,083	liquid stream		
1,000,058	oil to PCV-109 (S4)	4,786	is output of	1,000,031	splitting of S2		
1,000,059	library pump symbol-1	1,225	is classified as a	610,342	pump symbol		
1,000,062	LO-100 picture & P&ID	1,225	is classified as a	490,211	document		
1,000,062	LO-100 picture & P&ID	4,996	is presented on	1,000,063	Lub-oil-sheets-07-05-03.ppt		
1,000,062	LO-100 picture & P&ID	4,996	is presented on	1,000,067	PFS and P&ID Lub Oil System API 614		
1,000,063	Lub-oil-sheets-07-05-03.ppt	1,225	is classified as a	490,533	electronic data file		
1,000,063	Lub-oil-sheets-07-05-03.ppt	1,227	is element of	1,000,065	C:\BEYING		
1,000,065	C:\BEYING	1,225	is classified as a	492,017	directory		
1,000,066	P-107B	1,190	is part of	1,000,002	Pump system 107		
1,000,066	P-107B	1,225	is classified as a	1,000,041	Allweiler - SN 210-46		
1,000,069	PFS and P&ID Lub Oil System API 614	1,225	is classified as a	490,533	electronic data file		
5,490,406	DEP 31.29.60.32-Gen.	2,071	is represented by	5,499,188	31296032.zip		
5,499,188	31296032.zip	1,227	is element of	1,000,065	C:\BEYING		

The above table illustrates among others that all individual things in a product model are classified by a concept from the Gellish dictionary or by a subtype of such a concept.
For example, the third line defines the P-107A is classified as an Allweiler type SN 210-46 pump. Both the individual thing P-107A as well as the concept Allweiler SN 210-46 are new and not yet known in the Gellish dictionary / taxonomy. P-107A is added via this classification line, and fact id 2,000,184 adds the concept through the expression of the fact that Allweiler SN 210-46 is a specialization of horizontal triple screw pump, while that latter is an existing concept in the Gellish dictionary. The right hand side concepts are also either concepts from the Gellish dictionary (numbers below 1.000.000) or they are defined by classification relations elsewhere in the table.

If an individual thing is classified as being a member of a kind of thing, then the individual thing is supposed to have aspects that are obligatory for the kind and it may have aspects that are optional for the kind. It may even have aspects that are not specified for the kind, because a definition of a kind only specifies constraints and does not constrain things that are not specified.

> Note that the opposite is the case in many information systems, where each entity type or object type is defined by its attributes (and its methods), whereas the attributes define the degree of freedom and do not define the constraints on an existing freedom. This has as consequence for those systems that an instance of such an entity type cannot have attribute values for attributes that are not predefined as attribute of the entity type. This constraint is not applicable in Gellish.

Strictly speaking the individual thing *does not inherit* its aspects from its classifying kind of thing for two reasons: (1) individual things exist and have aspects independent of the definition of kinds of things and (2) the definition of aspect values for a kind of thing specifies typically an allowed range for those values which is wider than the aspect values of the individual thing. Therefore, we should say that the aspects of the kind *are applicable* for the individual thing. This means that we can *allocate* the constraints to the individual thing, but not its actual values, apart from the cases where the kind of thing specifies a non-optional point value for a kind of aspect. This allocation includes the constraining aspects that the kind of thing inherits from all the supertypes of the kind of thing. Therefore, the concept 'classification of an individual thing' is a relation type that implies that there is an unambiguously specification of the scope of the aspects that are inherited by the classifying kind of thing from the supertypes of the kind of thing and thus are allocated to the classified individual things.

Individual things that are classified by a kind has (or shall have) aspects that are within the constraints of the classifying kind of thing, including also all the constraining aspects that are inherited from all their supertypes. In addition to that it should be recognized that the degrees of freedom that are left by the applicable kinds of relations are also degrees of freedom that the classified individual things have.

6.3.1 Verification of a design

The knowledge that is contained in the definition of a kind of thing can be used to verify the information that is expressed (by another party) about an individual thing. This is the case, because the individual aspects of the individual things can be compared with the definition of the kind and by its optional aspects as well as with the standard specifications that are valid in a particular context.

This mechanism has many practical applications in the industry, especially when information about design is handed over together with delivery of artifacts.

This is illustrated by the comparison of the above three tables (in section 6.1.1, 6.2 and 6.3). In the case of a handover in this context (the BEYING project) the product model of the third table must be compliant with the standard specifications of the second table and must be within the constraints expressed in the knowledge model of the first table. This illustrates that, if the three models are all expressed in the Gellish language, then a computer can create a report on the verification of the product model.

This capability has many practical applications and is potentially of great value for the industry, especially to verify delivered products with their specifications and standards.

6.3.2 Usage of knowledge via classification

The concept 'kind of thing' can be used in two ways: as a conclusion and as a decision. These two ways of usage relate to two directions in which the 'classification of individual' relation can be used:

- **classification by conclusion**, which begins with the observation of the aspects of an existing individual thing and which concludes that it is one of a particular kind (or class)
- **classification by decision**, which begins with the existence of a particular kind of thing and decides that an individual thing is created that is (shall be or will be) of a that particular kind. After that the aspects of the individual thing are created within the constraints defined by the kind of thing.

This implies that the classification relation between an individual thing and a kind of thing can be created for two reasons, each of which gives a different perspective on the classification relation.

6.3.2.1 Classification by decision

Classification by decision is often used during the design of technical artifacts. In the beginning of the design process the imaginary individual artifact that is being designed is defined only by a limited number of aspects, or even only by the single fact that it has a role as performer of a particular process. On the other hand kinds of thing may be defined before individual things are created. For example suppliers who produce products before they sell them, as production for stock and selling from stock, define product types (kinds of things) from which designers can select. By *selecting* a *kind* of artifact and by *deciding* that the imaginary individual artifact shall be one of that kind, one declares (decides) implicitly that the imaginary individual artifact will have aspects that are characterizing the kind of thing.

This is illustrated by the example in Figure 95.

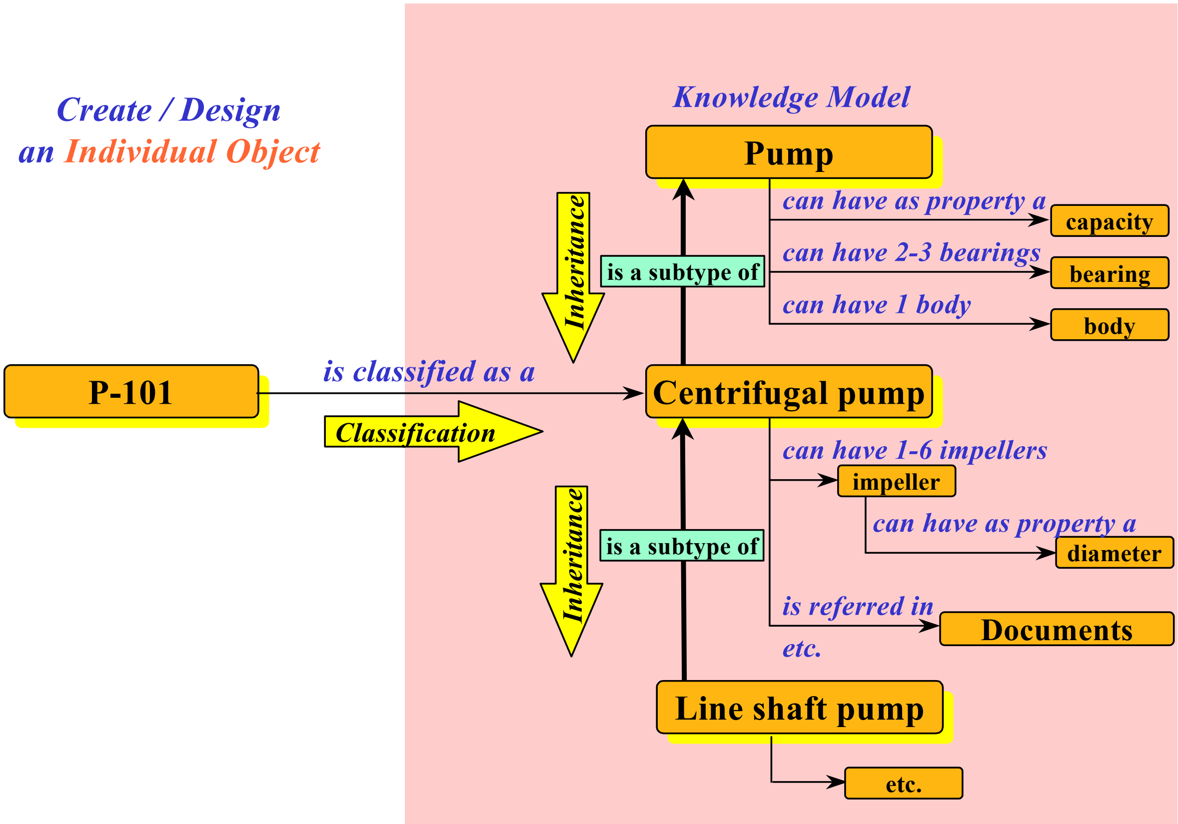


Figure 95, Aspects of a kind of thing are allocated to an individual thing

The left hand object P-101 is classified by a design decision to be a centrifugal pump. For that particular kind of thing the knowledge model (partly shown on the right hand side of the figure) contains a decomposition hierarchy that specifies that a well-formed centrifugal pump can consist of a

number of parts of particular kinds, such as impellers, shafts, etc. It also specifies which properties are conceptually possessed by an individual assembly of that kind and by those parts.
This classification therefore means that P-101 has (or will have) aspects that characterize a centrifugal pump, such as the fact that it can have 1 to 6 impellers and that an (each) impeller can have a diameter. The specialization hierarchy of the ontology defines that the concept centrifugal pump is a subtype of the concept pump. Therefore, the concept centrifugal pump inherits aspects from pump and from all its supertypes. For example it inherits from pump that it can have a capacity and two or three bearings and a body and it inherits from higher level supertypes for example that it is a solid item with a mass, which obeys the laws of physics, etc.
In the ontology there may also exist further subtypes of centrifugal pump, such as line shaft pump with its additional constraints. By classifying P-101 as a centrifugal pump it means that it is unsure whether P-101 will have the aspects of a line shaft pump or of one of the other subtypes.
So, the *decision* that the imaginary thing is classified as being of a particular kind implies that its aspects are within the limits of the ranges for the aspects of that kind and its supertypes, including the normal composition of that kind and including the normal aspects of its parts (and the parts of its parts, etc.). In other words, the classification by decision implies that the aspects of the kind of thing are *allocated* to the individual thing. This does not necessarily imply that its aspects and parts are within the constraints of further subtypes of that particular kind.
If later a real individual artifact is produced which has to comply with the imaginary design, then the aspects of the real individual artifact shall also have aspects that are within the same ranges. Informally, the allocation of aspects of a kind of thing to the individual thing is often called inheritance, which would means that the individual thing would inherit aspects of the kind of thing. However this allocation of aspects has to be clearly distinguished form the earlier described inheritance of aspects between kinds of things (from supertypes to subtypes), Between kinds of things that are each other sub- and supertype there is a true inheritance of all aspects. But an individual thing only has aspects that are within the ranges of the aspects that define the kind of thing. The kind will still have degrees of freedom, so that the individual things that are of the kind can and will differ from each other.

The creation of detailed knowledge models as part of an ontology with a specialization hierarchy enables software to assist by allocating the fixed part of the knowledge to the artifact that is being designed, while presenting the degrees of freedom (the variable part of the definition of the kind) to the user for selection of the options for the individual thing and for further detailed classification by a subtype of the kind. This methodology has advantages for a design process, because it speeds up the design and nevertheless leaves particular options open. Application of this methodology therefore can increase the efficiency of a design process.

This process is illustrated in Figure 96.

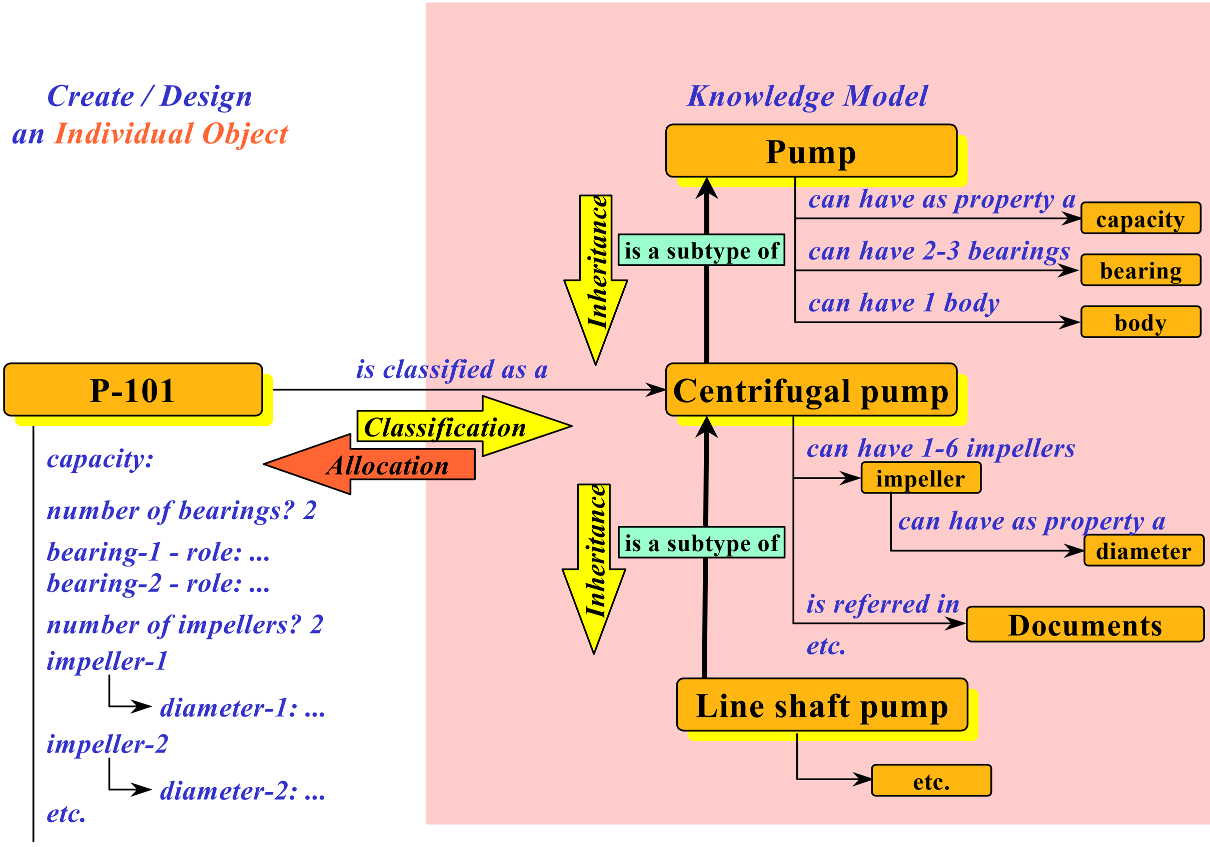


Figure 96, Allocation of aspects to an individual thing and its detailed specification

The figure illustrates that a software application can assist a designer of a pump for example by deriving the following knowledge from the knowledge model of a centrifugal pump:

- The individual artifact P-101 is by definition an assembly of parts of particular kinds, because that knowledge is allocated to it on the basis of the normal composition of centrifugal pump.
- One of the options of the parts is that P-101 must have between one and six impellers, whereas software may ask how many impellers P-101 shall have.
- Other knowledge about the definition of a centrifugal pump and its supertypes is also allocated to P-101, which implies that it will have a capacity, bearings, a body, etc.
- From the definition of the knowledge model it appears that each of the impellers normally has a diameter, material of construction, number of blades, etc. So, software may present the options to the user for detailed specification and selection from allowed values, such as a list of allowed materials of construction.

Note that a designed individual thing not only differs from a kind of thing because choices are made for the options for the aspects of the kind of thing, but also because any individual thing can have aspects that are not defined for the kind of thing. The Gellish language allows such individual aspects and their qualification or quantification as aspects that can be possessed by individual things even if nothing is recorded about such aspects for the kind of thing.

By the definition and usage of such knowledge models and further specialized knowledge models, up to fully specified standard components, it is possible to simplify the design process and to increase the design efficiency significantly. It also enables the exchange designs of parts created by different parties, verify their quality and integrate those parts in larger assemblies.

The above described method of allocation of aspects to individual thing through classification of the individual thing is comparable with ‘instantiation’ (the creation of an instance of a class) in the conventional information technology, whereby also qualified aspects can be allocated and where the knowledge about the class is not defined in the data model or schema as entities and attributes, but is expressed in Gellish as well.

6.3.2.2 Classification by conclusion

Classification by conclusion occurs normally when things from reality are classified according to their observed aspects. The things in reality have those aspects 'from themselves', they are not allocated or derived from the definition of kinds of things. By comparing those aspects with the aspects that are defined for various kinds of things one can conclude whether an individual thing is one of a kind. This is illustrated in Figure 97.

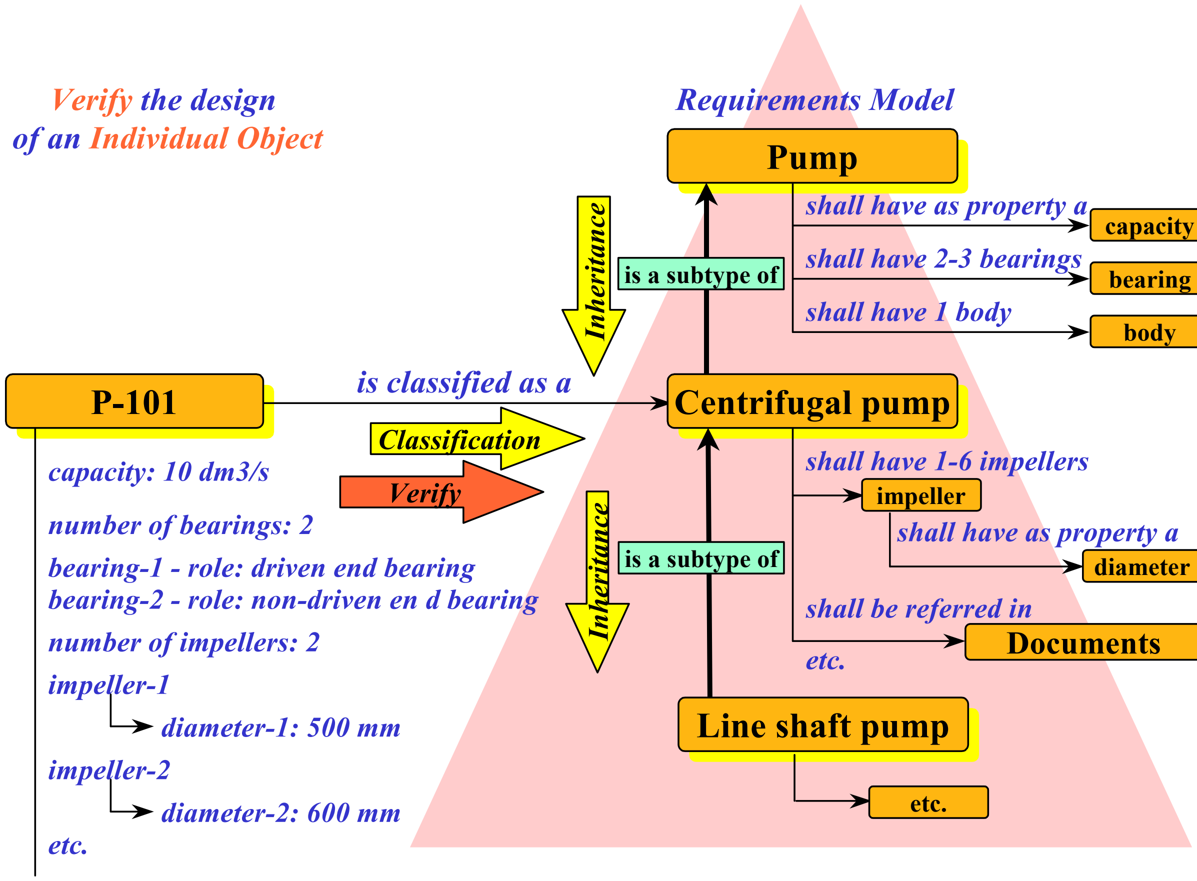


Figure 97, Classification and verification by conclusion

The right hand of the above figure contains not just knowledge, but standard specifications for the kind of thing (not 'can have …' , but 'shall have …' relations). The left hand illustrates either a design or an observed individual thing; a pump in this case. The classification of the left hand object as a centrifugal pump can now be *verified*, by testing whether its aspects are within the definition of the aspects of the knowledge model (Figure 95) and by testing whether its aspects are within the standard specifications for the kind of thing (requirements for a centrifugal pump, Figure 97). The conclusion of the verification process can be: yes, the aspects are within the definition of a centrifugal pump and within the standard specifications or a list can be provided on the aspects that deviate from either of the two.

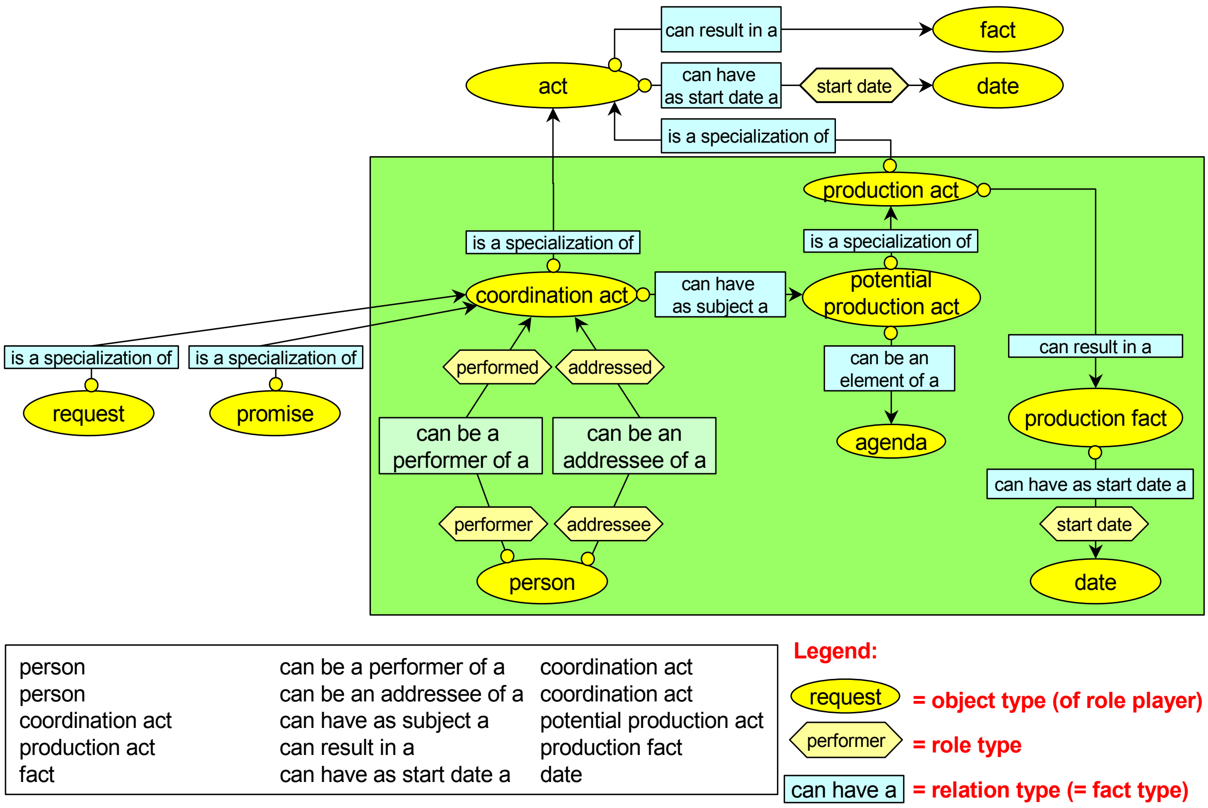
This kind of automated verification can be applied especially for product quality verification. For example, if a product is produced it may be tested against a definition of a kind of product, to verify whether the aspects of the produced product are within the limits allowed for the kind. The overall conclusion will be that it is either on-spec or off-spec. This is similar to the verification whether a produced product is according to a design, where tolerances are defined for the allowed deviation from the design. In the latter case, the reference for the verification is not a kind of thing, but an individual thing. The individual reference product, together with the tolerances and allowed deviations implicitly or indirectly define a kind of thing, being an envelope of all the things that satisfy the criterion of deviating less than those tolerances.

6.3.3 Usage of Gellish for communication about business transactions

In this section it is described how Gellish can be used in transaction processes. It contains a partial transcription of the generic patterns in business processes and business communication as described in the article "The Atoms, Molecules and Fibers of Organizations", by Jan Dietz (2002) into Gellish (presented as Gellish Tables).

Information exchange between computer applications is usually one-way traffic. In general, a message is sent from one application to another, but rarely it comes to a real dialogue. This is opposed to business communication, where transaction processes require usually a sequence of various kinds of coordination acts, each with a communicative intent with which the sender of a message intents to achieve something from the receiver of the message. This section illustrates that it is possible to express these messages in Gellish. This means that a computer application should be able to interpret them and, when it would have the knowledge of the dialogue process also expressed in Gellish, then it must be possible to develop 'agent' software that can interpret a message, act on it (for example by verification stock levels and product availabilities) and then return a response message, again expressed in Gellish.

Jan Dietz (1996) analyzed such transaction processes and documented the results in the DEMO methodology. He concluded that communication processes consists of messages each of which is structured conform a generic pattern of a communicative act. That generic pattern can be expressed in the Gellish language, as is illustrated in Figure 98.



person	can be a performer of a	coordination act
person	can be an addressee of a	coordination act
coordination act	can have as subject a	potential production act
production act	can result in a	production fact
fact	can have as start date a	date

Figure 98, The generic recurrent pattern of a coordination act

The main difference between messages in a dialogue is that each message has a specific communicative intent. For example, a request and a promise are both subtypes of a coordination act and can both be described by the structure of relations indicated in the shaded area and both include the inherited pattern from the generic concept 'act', while the communicative intent of a question is to achieve commitment, whereas a promise expresses such a commitment. In one transaction, both will have as subject the same potential production act.

Furthermore, Dietz (2002) discovered a generic pattern in a process that seems to be applicable for any transaction. That generic process is presented in Figure 99.

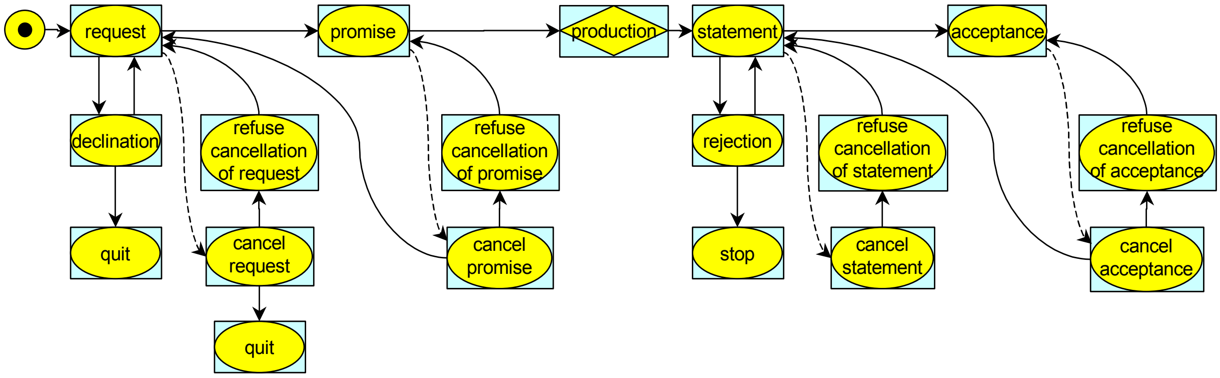


Figure 99, The generic process of a transaction

The above figure illustrates a sequence of coordination acts, including one production act that together form a transaction process (excluding the inquiry phase). Each of these acts conforms to the pattern of Figure 98 and therefore results in a fact that can also be expressed in Gellish.

This section describes the use of Gellish in transaction processes as follows:

- It defines the concepts identified in the DEMO methodology and adds the missing ones to the Gellish dictionary / taxonomy.
- It express the knowledge about electronic business communication as defined in the DEMO methodology in Gellish. This makes the communication process computer interpretable in a system independent way.
- It illustrates the expression of individual business transactions in Gellish, while using those concepts and that knowledge.

Gellish is a dynamic language, just as any living natural language: it grows. Therefore, for communication about business transactions it is extended with the concepts identified in DEMO as far as they were not yet included in the Gellish dictionary / taxonomy before DEMO and Gellish were aligned.
The concepts and processes that appear in DEMO and that are included in Gellish are defined as part of the Gellish dictionary / taxonomy by the specialization relations below.
The examples of individual communications illustrate how Gellish can be applied to express information about specific instances of business processes and transactions.

6.3.3.1 Organization, coordination acts and production acts

DEMO defines an organization as a social system that performs coordination acts as well as production acts. For that purpose DEMO recognizes the following concepts with the knowledge about a possible decomposition, a required role and a role possible player:

Left hand object name	**Cardinalities**	**Fact id**	**Relation type name**	**Cardinalities**	**Right hand object name**	**Additional definition**
social system			is a specialization of		system	that consists of human beings that have a relation with each other.
organization			is a specialization of		social system	with a particular purpose or mission.
person	1,n		can be part of a	0,n	organization	
business process			is a specialization of		process	that consists of a structure of coherent activities intended for a specific business purpose performed by one or more persons representing one or more organizations.

business transaction			is a specialization of		transaction	that starts with a communicative act that states a wanted potential fact. This potential fact is discussed and may be fulfilled by an actual fact. The transaction stops either by a withdrawal of the wanted fact or the fulfillment of the wanted fact by an actual fact and the agreed remuneration.
actor			is a specialization of		role	of a person (subject) that has an amount of authority and ability to perform an act.
communicatio n act			is a specialization of		act	is an act of communicating a message.
communicative act			is a synonym of		communication act	
language act			is a synonym of		communication act	
act			requires a role as a		actor	
person			can have a role as a		actor	
production act			is a specialization of		act	by which one or more persons bring about a material or immaterial good or service that is provided to or delivered to the environment of an organization or to a person within the organization.
coordination act			is a specialization of		act	by which a person interacts with one or more other persons inside or outside an organization by which the person enters into and complies with a commitment or agreement towards those persons regarding the performance of production acts.
request			is a specialization of		coordination act	by which something is asked to someone.
promise			is a specialization of		coordination act	is a statement that something will be done, brought about, or provided.

Once the above concepts are defined the Gellish language can be used to create a message about an individual transaction.

For example, a request message called Request-1 to perform the production act 'Delivery-1' could be expressed in Gellish as follows:

Left hand object name	Fact id	Relation type name	Right hand object name
Delivery-1	1	has as subject	P-1
Delivery-1	2	has as location	Main street 1, Delft
Delivery-1	3	is classified as a	delivery
Request-1	4	has as subject	Delivery-1
Request-1	5	is classified as a	request
P-1	6	is classified as a	collection of items
P-1	7	each of which is classified as a	pipe
Main street 1, Delft	8	is classified as a	address

The above statements 1 through 5 are equivalent to the following expression in DEMO style, whereas the Gellish expressions extent this by making the delivery act explicit and by classification of P1 as a pipe and the text string as an address:

request	P-1		is delivered at	Main street 1, Delft

6.3.3.2 Coordination facts and production facts

DEMO recognizes that business acts cause the creation of facts, whereas DEMO distinguishes between coordination facts, communicative facts and production facts and their corresponding acts that cause them. This is reflected in the expression of knowledge about fact types and their relation to act types in Gellish as follows.

Left hand object name	Fact id	Relation type name	Right hand object name	Additional definition
fact		can be a result of a	act	
coordination fact		is a specialization of	fact	that is the result of a successful performance of a coordination act.
coordination fact		can be a result of a	coordination act	
communicative fact		is a specialization of	fact	that a communication took place by an exchange of a message that expresses an opinion about a ‘potential fact’ or about an ‘actual fact’.
communicative fact		can be a result of a	communicative act	
production fact		is a specialization of	fact	is the result of the successful performance of a production act.
production fact		can be a result of a	production act	

Note that a ‘production fact’ in DEMO is not necessarily ‘something that is the case’; it can also be ‘wanted to be the case’ or ‘claimed to be the case’. A ‘fact’ that is wanted to be the case is an imaginary potential fact, a ‘fact’ that is claimed to be the case is a claimed actual fact.
This means that there are at least two types of production facts:

(1) Initially there is a *wanted potential fact* about which is communicated during a business transaction process. This is an *imaginary fact*. Typically it is only worthwhile to be discussed as long as the originator maintains his position about the imaginary fact. For example: John has the requirement that he wants fact 1: to have pipes P-1 according to a specification delivered at Main street 1 in Delft.
The ‘truth status’ of a potential fact can be: *wanted*, or *not wanted any more*, or *realized (satisfied).*

(2) Later, there can be a *claimed actual fact* (or more than one actual fact) that is subject to a communication process because it is claimed to be the case (and is intended to satisfy the wanted fact). For example: Fred declares fact 15: that P-2 is delivered at Main street 2 in Delft.
The ‘truth status’ of an actual fact can be: *stated* or *claimed*, *doubted*, *denied*, and *agreed to be the case*, *withdrawn to be the case*.

Similarly there is a wanted imaginary act and a real actual act that shall comply with the wanted act.

DEMO implies also an additional type of ‘coordination fact’ that claims that an actual fact fulfils a wanted potential fact. This is a *claimed fulfillment fact* that is subject to a communication process. It is expressed by a relation that states that actual fact (2) is a fulfillment of imaginary fact (1). For example Delivery 2 in the table below is a different (real) delivery as the wanted Delivery-1. So, a statement that the real Delivery-2 complies with wanted Delivery-1, has a truth status that can be disputed, because P2 is delivered at the wrong address.
This illustrates that fulfillment facts can have various values for their ‘truth status’. Values of the truth status of a fulfillment fact can be: *stated* or *claimed*, *confirmed*, *agreed to fulfill*, *denied*, *agreed not to fulfill*, and *accepted*.

A message which includes the declaration that an actual individual production act Delivery-2 has taken place and includes Request-2 for Payment-2 can be exchanged in Gellish as follows:

Left hand object name	Fact UID	Relation type name	Right hand object name	Truth status of the fact
Delivery-2	14	has as subject	P-2	declared to be the case
Delivery-2	15	has as location	Main street 2, Delft	declared to be the case
Delivery-2	16	has as progress status	completed	declared to be the case
Delivery-2	17	is classified as a	delivery	
Payment-2	18	has as subject	The price of P-2	declared to be the case
Payment-2	19	is classified as a	payment	
Payment-2	20	has as receiver	The Piping Company	declared to be the case
Request-2	21	has as subject	Payment-2	declared to be the case
Request-2	22	is classified as a	request	
P-2	25	is classified as a	collection of items	
P-2	26	each of which is classified as a	pipe	
The price of P-2	27	is classified as a	total price	

A piece of a message about the fulfillment of coordination facts are for example:

P-2	23	complies with	P-1	declared to be the case
Delivery-2	24	complies with	Delivery-1	declared to be the case

An example of an event that describes the relation between an act and the fact that is created by the execution of the act is:

Delivery-2	28	has created fact	16	

It can be argued that not Delivery-2, but Declaration-2 creates fact 16.

The coordination act that declares that Delivery-2 has taken place is expressed in Gellish as follows:

Declaration-2	29	has as subject	Delivery-2
Declaration-2	30	is classified as a	declaration
Declaration-2	31	has as subject status	completed

An example that describes the fact that is the result of the above declaration act is:

Delivery-2	32	has as status	completed	declared to be the case

The above statement 14, 15 and 29 together are equivalent to the following expression in the DEMO style:

declaratior	P-2	is delivered at	Main street 2, Delft

6.3.3.3 State transition

An act that causes a fact also causes that a fact is added to a world. That change is called a state transition (a transition of the state of that world). This is expressed as follows:

act		can cause a	state transition	
state transition		is a specialization of	transition	of a 'world' by adding a fact to the collection of facts or changing a status of a fact.
transition		is a specialization of	event	by which a situation transforms into another situation.

6.3.3.4 Agenda and responsibilities

The fact that persons can have an agenda, responsibilities, authorities and competences is expressed in Gellish as follows:

Left hand object	Fact id	Relation type name	Right hand object	Additional definition
agenda		is a specialization of	collection of facts	of an actor, each of which is an agendum.
agendum		is a specialization of	coordination fact	of an actor, to which the actor is committed to respond.
action rule		is a specialization of	rule	that specifies what should be done in case of a response to an agendum.
person		can be committed to a	response	
commitment		requires a role as a	committer	
commitment		requires a role as a	committed	
coordination fact		can have a role as a	committed	
person		can have a role as a	committer	
competence		is a specialization of	ability	of a person to perform a kind of production act as well as the corresponding coordination acts. In other words he has the necessary and sufficient knowledge, expertise and experience for being a professional in a certain field.
authority		is a specialization of	right	of a person to act on behalf of an organization.
responsibility		is a specialization of	liability	of a person to exert a granted authority in line with the values and norms of the institution one represents and of the social and religious society one is a member of. *(or accountability)*
person		can be employed by a	organization	(subtype of 'can be a part of a'??)
person		can have a	competence	
person		can have a	authority	
person		can have a	responsibility	
plumber		is a qualification of a	competence	in the field of plumbing.

For example:

Left hand object name	Fact id	Relation type name	Right hand object name
Fred	33	is committed to	respond to Request-1
respond to Request-1	34	has as subject	Request-1
Fred	35	has as aspect	competence of Fred
competence of Fred	36	is qualified as	transporter
Fred	37	is classified as a	person
respond to Request-1	38	is classified as a	response
competence of Fred	39	is classified as a	competence

Note that according to DEMO the commitment is to a fact, however it may be that a commitment is a commitment to perform an act that results in the requested fact.

6.3.3.5 Coordination acts

A *coordination act* requires fulfillers of a number of roles. Knowledge about them is expressed in Gellish as follows:

Left hand object name	Fact id	Relation type name	Right hand object name	Additional definition
coordination act		requires a role as a	performer	
performer		is a specialization of	actor	as being a person who executes the activity.
initiator		is a specialization of	performer	as being the one who takes an initiative.
sender		is a specialization of	performer	as being a person who sends a message.
coordination act		requires a role as a	addressee	
addressee		is a specialization of	actor	as being a person to whom a message is addressed.
coordination act		requires a role as a	subject	
person		can have a role as a	author	
person		can have a role as a	addressee	
potential fact		can have a role as a	subject	
actual fact		can have a role as a	subject	
fact		requires a role as a	start date	
start date		can be a role of a	date	
registrar		is a specialization of	performer	as being the person who registers.
registration		requires a role as a	registrar	
registration		requires a role as a	registree	
registree		is a specialization of	actor	
person		can have as role a	registree	

Proposition.
A proposition is a (wanted) potential production fact or actual production fact at a moment in time, which moment indicates the (potential) start of the existence of the (potential or actual) fact. For example: John is member of Library-1 since 2/4/2002 or John is owner of 'pump 12345' since 18 July 2003.
Note: a fact always has a start time. Therefore a proposition seems to be an expression of an opinion that something is the case since (or until) a moment in time. In Gellish this is expressed in two lines as two Gellish 'atomic facts', each with its own identification. There seems to be no reason to have a separate identifier for the combination of those two facts.

The example of a request by John to register his company as a 'Class A customer of Fred's company' (instead of registration as member of a Library-1 in the example in Dietz 2002), becomes in Gellish:

Left hand object name	Fact id	Relation type name	Right hand object name
John	40	is initiator of	request 387
Fred	41	is addressee of	request 387
request 387	42	has as subject	fact / proposition 387
John Ltd	387	is classified as a	Class A customer of FLtd
fact / proposition 387	388	has as start date	1/4/2002
John	43	is classified as a	person
Fred	44	is classified as a	person
request 387	45	is classified as a	request
fact / proposition 387	46	is classified as a	fact
request 387	47	has as progress status	completed

intention		is a specialization of	social attitude	that qualifies a proposition (or 'fact') by the mental objective of the originator of a message in a coordination act.
requested		is a qualification of a	intention	of being asked.
promised		is a qualification of a	intention	of being committed to be realized.
withdrawn		is a qualification of a	intention	of being not committed anymore.
fulfilled		is a qualification of a	intention	of being satisfied by an actual occurrence.
stated		is a qualification of a	intention	of being declared to be the case.
accepted		is a qualification of a	intention	of being adopted as satisfactory.

The layers of communication (figure 4 in Dietz 2002) can be expressed in Gellish as follows:

Left hand object name	Fact id	Relation type name	Right hand object name	Additional definition
performative act		is a specialization of	act	that aim to arrive at social understanding.
performative act		can be part of a	coordination act	
informative act		is a specialization of	act	that aims arrive at intellectual understanding.
informative act		can be part of a	coordination act	
inform		can be part of a	informative act	
confirm		can be part of a	informative act	
performa condition		is a specialization of	condition	whether a social understanding is raised in an addressee.
satisfied		is a qualification of a	condition	of being the case.
not satisfied		is a qualification of a	condition	of not being the case.
performed		is a specialization of	satisfied	by being executed.
not performed		is a specialization of	not satisfied	by not being executed.
understood		is a specialization of	satisfied	by being comprehended.
not understood		is a specialization of	not satisfied	by not being comprehended.

informa condition		is a specialization of	condition	whether an intellectual understanding of a coordination act is established in an addressee.
informative exchange		is a specialization of	exchange	of a message between a sender to a receiver.
exchange		requires as role a	sender	
exchange		requires as role a	receiver	
sender		is a specialization of	performer	by being the one from whom a item originates. Typically a message.
receiver		is a specialization of	performer	by being the one who gets the item. Typically a message.
inform		can be part of a	informative exchange	
confirm		can be part of a	informative exchange	
question		can be part of a	informative exchange	
assertion		can be part of a	informative exchange	

For example: Fred can question and discuss the request of John as follows:

Left hand object name	**Fact id**	**Relation type name**	**Right hand object name**
Fred	48	is originator of	question-1
question-1	49	has as subject	fact / proposition 387
John	50	is addressee of	question-1
John	51	is originator of	assertion-1
assertion-1	52	has as subject	fact / proposition 387
Fred	53	is addressee of	assertion-1
Fred	54	is originator of	confirmation-1
confirmation-1	55	has as subject	fact / proposition 387
John	56	is addressee of	confirmation-1
question-1	57	is classified as a	question
assertion-1	58	is classified as a	assertion
confirmation-1	59	is classified as a	confirmation

Note: a rule defines that a confirmation creates a 'satisfied' condition (to be defined: which and when).

During the communication process above, the request 387 has the following progress statuses:

Request 387	60	has as progress status	questioned
Request 387	61	has as progress status	asserted
Request 387	62	has as progress status	confirmed

The lower 'forma' level is defined as follows:

Left hand object name	Fact id	Relation type name	Right hand object name	Additional definition
forma condition		is a specialization of	condition	that is a collection of an establishment condition and a functioning condition of a communication channel. Such a channel can be used to perform (two way) formative acts that realize informative acts.
establishment condition		is a specialization of	condition	whether a communication channel between a sender and a receiver is established.
establishment condition		can be part of a	forma condition	
functioning condition		is a specialization of	condition	whether a communication channel is well functioning
functioning condition		can be part of a	forma condition	
established		is a specialization of	satisfied	
not established		is a specialization of	not satisfied	
well functioning		is a specialization of	satisfied	
not well functioning		is a specialization of	not satisfied	
formative exchange		is a specialization of	exchange	between actors to verify establishment and well functioning of a communication channel.
express		can be part of a	formative exchange	
transmission		can be part of a	formative exchange	
perceive		can be part of a	formative exchange	

6.3.3.6 Production acts

Delivery-1 in the above examples is a required production act that is defined by the coordination act Request-1 as expressed in fact 5 (the proposition). The request is fulfilled by Delivery-2 as an actual production act that brings about a DEMO production fact that is described by the pair of Gellish facts 6a and 6.

6.3.3.7 The atomic layer: action rules

A person who responds to a coordination fact follows an action rule when performing his coordination act.

Left hand object name	Fact id	Relation type name	Right hand object name
person		can be committed to respond to a	coordination fact
person		can be a performer of a	coordination act
coordination act		can be executed conform a	action rule
condition		can be a role in a	action rule
action on satisfied condition		can be a role in a	action rule
action on unsatisfied condition		can be a role in a	action rule

For every response type there shall be a rule type. For example if Fred's company only delivers carbon steel pipe (and not e.g. stainless steel or aluminium pipe) then the rule definition might be:

Left hand object name	Fact id	Relation type name	Right hand object name
Fred's Ltd order acceptance	65	is a specialization of	coordination act
Fred's Ltd order acceptance	66	can have as subject a	request to deliver pipe
Fred's Ltd order acceptance	67	can be executed conform a	F's Ltd acceptance rule
F's Ltd acceptance rule	68	is a specialization of	action rule
pipe made of CS (fact 100)	69	can have a role as a	condition in F's Ltd rule
pipe	100	shall be made of	carbon steel
promise to deliver	71	can have a role as a	action on satisfied condition in F's Ltd rule
decline to deliver	72	can have a role as a	action on unsatisfied condition in F's Ltd rule
condition in F's Ltd rule	73	is a specialization of	condition
action on satisfied condition in F's Ltd rule	74	is a specialization of	action on satisfied condition
action on unsatisfied condition in F's Ltd rule	75	is a specialization of	action on unsatisfied condition
deliver pipe	76	can be a successor of a	promise to deliver

6.3.3.8 The molecular layer – interaction patterns

Transactions are initiated by an initiator or more specifically by a customer and are executed by an executor or more specifically by a supplier. This is expressed in Gellish as follows:

Left hand object name	Fact id	Relation type name	Right hand object name	Additional definition
transaction		is a specialization of	collection of acts	each of which is a coordination act which appears in a particular sequence of act type and which are concerned with the same wanted production fact and its fulfiller.
transaction		each of which is a	coordination act	
person		can have a role as a	initiator	
initiator		is a specialization of	actor	by being the person who starts an action. Typically who starts a transaction.
person		can have a role as a	executor	
executor		is a specialization of	actor	by being the person who does something. Typically who does what an initiator wants.
customer		is a specialization of	initiator	by being the person who starts a transaction.
supplier		is a specialization of	executor	by being the person who supplies what a customer wants.

Transactions of a certain type appear to be performed according to a pattern of a sequence of types of coordination and production acts. A general pattern of a transaction is defined as follows:

Left hand object name	Relation type name	Right hand object name	Additional definition
declaration	is a specialization of	statement	by making known formally, officially, or explicitly that something is the case, or will be the case.
promise	is a specialization of	commitment	that something will be done, brought about, or provided.
promise	can be a successor in time of a	request	
promise	can be a successor in time of a	refusal of cancellation of a promise	
promise	can be a part of a	transaction	
act	is a specialization of	to act	by doing one thing. Typically taking a short time.
assertion	is a specialization of	statement	in positive and often forceful or aggressive way.
apology	is a specialization of	expression	of feelings offered in explanation or defence. Typically but not necessarily an expression of regret for a mistake or wrong with implied admission of guilt or fault and with or without reference to palliating circumstances.
formulation	is a specialization of	act	by putting into a systematized statement or expression.
interpretation	is a specialization of	apprehension	of an observation by reason or imagination in the light of individual belief, judgment, or circumstance.
apprehension	is a specialization of	becoming aware	with understanding by recognizing a meaning of an encoding.
becoming aware	is a specialization of	becoming	in a state of awareness.
reply	is a specialization of	response	verbally or in writing.
request for a quotation	is a specialization of	request	a receiver to present a quotation.
confirmation of a quotation request	is a specialization of	confirmation	that a request to present a quotation was received. Typically including an interpreted confirmation of the content of the request and possibly an indication of the timing of a planned response.
exchange of a quotation	is a specialization of	exchange	of a message that describes a potential delivery with its terms and conditions.
sending of a quotation	is a specialization of	sending	of a message that describes a potential delivery with its terms and conditions.
receipt of a quotation	is a specialization of	receipt	of a message that describes a potential delivery with its terms and conditions.
confirmation of a receipt of a quotation	is a specialization of	confirmation	that a quotation was received. Possibly including an indication of the timing of a planned response.
exchange of a purchase order	is a specialization of	exchange	of a message that describes a requested delivery with its terms and conditions.
sending of a purchase order	is a specialization of	sending	a message that describes a requested delivery with its terms and conditions.
receipt of a purchase order	is a specialization of	receipt	of a message that describes a requested delivery with its terms and conditions.
declaration of correctness of a delivery	is a specialization of	declaration	that a delivery is in accordance with the terms and conditions of an agreement.
verification of	is a specialization of	verification	whether a delivery is in accordance with the

correctness of a delivery			terms and conditions of an agreement.
quotation request	can be a subject in a	request for a quotation	
quotation request	is a specialization of	request	that describes goods or services for which a quotation is requested.
quotation request confirmation message	can be a subject in a	confirmation of a quotation request	
quotation request confirmation message	is a specialization of	confirmation message	that states that a quotation was received.
quotation message	can be a subject in a	exchange of a quotation	
quotation message	can be a subject in a	sending of a quotation	
quotation message	can be a subject in a	receipt of a quotation	
quotation message	is a specialization of	message	that declares that goods or services can be delivered subject to specified terms and conditions.
quotation receipt message	can be a subject in a	confirmation of receipt of a quotation	
quotation receipt message	is a specialization of	receipt message	that confirms that a quotation was received.
delivery correctness declaration message	can be a subject in a	declaration of correctness of a delivery	
delivery correctness declaration message	can be a result of a	verification of correctness of a delivery	
delivery correctness declaration message	is a specialization of	declaration message	that goods or services are correctly delivered.
confirmation message	is a specialization of	message	that confirms that something is the case.
declaration message	is a specialization of	message	that declares that something is the case.
receipt message	is a specialization of	message	that confirms that a described object was received.
communicative act	is a specialization of	act	of communicating a message.
perception	is a specialization of	act	by becoming aware through the senses.
waiving	is a specialization of	giving up	of a claim.
cancellation of a commitment	is a specialization of	cancellation of a promise	with an obligation to do what is promised. Typically by giving something instead.
delegation	is a specialization of	entrusting	to another person.
entrusting	is a specialization of	giving	somebody custody, care or charge of something.
delegation of a commitment	is a specialization of	delegation	of a commitment to execute or arrange the execution of an activity. With or without keeping the responsibility for the performance.
assignment of a claim	is a specialization of	assignment	of a claim to another person or organization who becomes the new creditor.
fulfilment	is a specialization of	act	in accordance with a specification or requirement.
fulfilment of a commitment	is a specialization of	fulfilment	of a commitment to another person or organization.
cancellation of a request	is a specialization of	cancellation	of a request by stating that the earlier request is withdrawn.
cancellation of a request	can be a successor in time of a	request	
cancellation of a request	can be a part of a	transaction	
cancellation of a promise	is a specialization of	cancellation	a promise by stating that an earlier promise to another person or organization is withdrawn.
cancellation of a promise	can be a successor in time of a	promise	

cancellation of a promise	can be a part of a	transaction	
cancellation of a statement	is a specialization of	cancellation	a statement by stating that an earlier statement to another person or organization is withdrawn.
cancellation of a statement	can be a successor in time of a	statement	
cancellation of a statement	can be a part of a	transaction	
cancellation of an acceptance	is a specialization of	cancellation	an acceptance by stating that an earlier acceptance to another person or organization is withdrawn.
cancellation of an acceptance	can be a successor in time of a	acceptance	
cancellation of an acceptance	can be a part of a	transaction	
rejection	is a specialization of	act	by refusing to act or to accept.
quitting	is a specialization of	act	by interrupting a process and bringing it to an abnormal end.
quitting	can be a successor in time of a	to decline	
quitting	can be a successor in time of a	cancellation of a request	
quitting	can be a part of a	transaction	
refusal of a cancellation of a request	is a specialization of	refusal	that a person or organisation that made a request would cancel the request.
refusal of a cancellation of a request	can be a successor in time of a	cancellation of a request	
refusal of a cancellation of a request	can be a part of a	transaction	
refusal of a cancellation of a promise	is a specialization of	refusal	that a person or organisation that made a promise would cancel the promise.
refusal of a cancellation of a promise	can be a successor in time of a	cancellation of a promise	
refusal of a cancellation of a promise	can be a part of a	transaction	
refusal of a cancellation of a statement	is a specialization of	refusal	that a person or organisation that made a statement would cancel the statement.
refusal of a cancellation of a statement	can be a successor in time of a	cancellation of a statement	
refusal of a cancellation of a statement	can be a part of a	transaction	
refusal of a cancellation of an acceptance	is a specialization of	refusal	that a person or organisation that accepted a delivery would cancel the acceptance.
refusal of a cancellation of an acceptance	can be a successor in time of a	cancellation of an acceptance	
refusal of a cancellation of an acceptance	can be a part of a	transaction	
rejection of a statement	is a specialization of	rejection	by stating that a statement is incorrect or unacceptable.
rejection of a statement	can be a successor in time of a	statement	
rejection of a statement	can be a part of a	transaction	
transaction	is a specialization of	business process	that starts with an ordering phase which may be followed by an execution phase and is normally terminated by a acceptance phase.

7 The linguistic perspective

This section compares the Gellish language with natural languages. Although Gellish defines concepts in a language independent way, it is required for human communication to use names which those concepts have in some natural language. Therefore this section illustrates that Gellish English is a structured subset of natural English, although it also holds that Gellish Dutch (Nederlands) is a structured subset of natural Dutch.

Gellish English is an artificial, formal language, which means that its concepts are explicitly defined and no concepts may be used that are not explicitly defined and added to the language with a given unique Gellish identifier (UID). However, the concepts that are defined in Gellish are not different from the concepts that are designated by terms as used in natural languages. Therefore, in that respect the Gellish English dictionary / taxonomy is not different from an ordinary English dictionary. However, *in a number of aspects the Gellish English dictionary / taxonomy has advantages over an ordinary English dictionary* for the following reasons:

- It defines concepts and not just terms. This means that many concepts are defined which name is composed of a number of terms. For example, the concept 'horizontal triple screw pump' is defined in the Gellish language, but will rarely be defined in any other language. It also makes explicit whether a term is a synonym of another term or in what way concepts are related to each other.
- It provides definitions that indicate the direct supertype concept of the concept that is defined. This ensures that definitions are not circular, as is sometimes the case in other dictionaries. It also means that the dictionary represents a systematic taxonomy, which implies that definitions and characteristics of concepts that are higher in the subtype/supertype hierarchy are inherited by the concepts that are lower in the hierarchy.
- It provides standard phrases (sentence fragments) that are expressions of concepts of fact types (kinds of relations). These standard phrases enable the creation of (English) expressions that express facts. For example, the phrases 'is a part of' and 'can have as aspect a' and 'can be a performer of a' are phrases that represent (single) semantic concepts that are missing in ordinary dictionaries.
- It provides unique identifiers of the concepts, irrespective of the natural language that is used. This enables automated translation of expressions of facts between English and any other language for which a Gellish dictionary is available.

Just as with a natural language, the grammar and semantics of Gellish is open and extendable, because the concepts and relation types that are defined in the Gellish dictionary / taxonomy are part of the Gellish language definition and any new concept that is defined can immediately be used in Gellish sentences. This is much less the case with most other artificial data definition and data manipulation languages that are defined in information science, such as SQL, EXPPRESS, OML or OWL. Those artificial languages generally are closed meta languages and a meta language is usually not extendable as it is standardised on a fixed set of only a limited number of generic concepts, together with a generic structure (or grammar). Although those concepts can be used to define other concepts, the definition of such additional concepts do not make them an extension of the meta languages.

7.1 The missing common language in information technology

When people communicate with each other in writing via computers, they may communicate in a natural language, or they may communicate in an artificial language. For example, they communicate in a natural language when they exchange electronic mail. Nevertheless the natural language that is used by the sender (the collection of expressions) is converted, usually in a binary encoding and carried as an electronic signal. That signal is transported to the receiver where the encoding is converted in a reverse way to a reconstruction of the original expression. But during those conversions the structure of the expressions remains the natural one. The computer is not able to act on the content of such messages as it is not able to interpret natural languages, because it is not 'aware' of the structure of the language.

Such an email communication differs from indirect communication between people when they use a database system for storage and retrieval of information, because that information is not stored in a natural language structure, but in an artificial database structure. Such a database structure can therefore be regarded to be an artificial language, although usually a very primitive one.

When 'data' is stored in an electronic database, this is usually done by filling predefined tables. The database structure that is essential for the interpretation of the meaning of the relations between the individual data elements is then contained in the following aspects:

1. The meaning of the columns in those database tables.
2. The nature of the mutual relations between the columns and their relation with the table as a whole.
3. The pointer-relations between columns in different tables.

Together these table column definitions and relations form the database structure, or in other words, the database definition. Such a database structure actually defines the grammar of the 'language' that is used for storage and retrieval of information in the database. *Knowledge of the (language) structure is necessary for a correct interpretation of the content of the database.* Software that 'understands' that grammar is able to interpret the data, due to that knowledge, and is therefore able to execute the appropriate operations. Such table structures are always defined by a database designer or programmer during the design of the database and of the software. This means that in doing so he or she has defined the 'language' that is to be applied when using that database.

In courses on database design it is common practice to leave the definition of those 'languages' completely in the freedom of the database designer or programmer! Generally there are no guidelines given to unify those 'languages' or to reduce the number of different 'languages' that are created this way. As a result, databases that are designed by different persons appear in practice to have completely different structures. In other words: *each database designer creates his or her own languages*, with their own grammars (= database structures) and their own dictionaries (= table-column definitions). Furthermore, the database designer usually leaves the column content (the user-dictionary) to the freedom of the database users, or the software ensures that the content is conform pick lists with predefined terminology or codes.
Such database systems may perform well as isolated systems, as islands of automation, but the consequences for the exchange of data between systems or for integration of data that stem from different systems is disastrous, because *all 'languages' used by the various systems are different.* The consequence therefore is that data cannot be exchanged between database systems without costly conversion of those data from the conventions of one 'language' to the other. Even different implementations of the same system still contains different definitions and different concepts due to differences in the various 'customizations'. This is apparent from the facts that quite a number of large companies have great difficulty to exchange data between their various ERP system implementations, and large projects have to be defined to migrate the data in various systems into one integrated system, because the data in the various ERP implementations are defined and named differently.

This is the main cause of the giant communication problem in information technology. It obstructs the interpretation of data that is exchanged between computer systems and necessitates the creation of many costly dedicated interfaces.
In other words, *in information technology there does not exist a common language; neither a common database structure, nor a common user language.*

This research aims to solve this problem by the definition and provision of Gellish as a common language or universal data structure that is more close to a natural language.

7.2 The Gellish grammar and semantics

In linguistics the syntax or grammar (being the structure of, or the rules for the structure of the words and sentences) is distinguished from the semantics (being the meaning of the words and sentences). Such a grammar is different per natural language. So, when particular information (being meaning, irrespective of the way in which it is expressed) is expressed in various languages, then for each language one has to apply different grammatical rules for the composition of the sentences. In the general comparative linguistics, a lot of research is spent on the rules for the creation of proper sentences. For example, resulting in the theory of the generative grammar, described by Noam Chomsky (1957) in his book 'Syntactic Structures'. In that book Chomsky describes his theory that sentences in a natural language can be generated through a process of successive simple 'grammatical transformations' of basic sentences ('terminal strings' or 'sequences of morphemes').
It should be noted however, that in the end communication deals with the transfer of *meaning*. The sender of an electronic message expresses information (or meaning) in a particular language. The result of that process is an expression, which is encoded in a signal that becomes the carrier of the meaning aspect. At a later stage and at another location that expression can be used to derive the meaning through an interpretation process and the interpretation results can be stored by the receiving party. In this process we can distinguish at least three forms of information- or meaning *carriers*, the written language, the signal and the stored bits, but their aspects all intent to represent only one meaning.

The definition of the concepts that are used in a natural language are descriptive, in the sense that a linguist has to describe the general nature of and the recurring patterns in existing, historic language expressions. To some extent this differs from the definition of Gellish English. On one hand Gellish English uses normal English names of concepts, but on the other hand it defines semantic concepts, especially the phrases that form 'names' of the relation types, that are the result of an analysis of the kinds of relations that are required to express knowledge and information. The Gellish ontology therefore identified the semantic concepts, and systematically arranged those concepts in a specialization hierarchy. Furthermore, the phrases that are suitable to designate those semantic concepts were determined. In this way, the language definition of Gellish is integrated with the description of the associated ontology.

7.3 Natural language independent semantic concepts

A description of the reality can consist of a composite of expressions, in which each expression expresses something that is the case, so each component expresses a fact. When somebody tells that something is the case then he or she expresses a fact, with the consequence that the receiver of the expression also knows that something is the case, then that expression contains information that has meaning for the sender as well as for the receiver. If the sender had expressed the same fact in another language, which was also understood by the receiver, then that second expression apparently contained the same information, because it has the same meaning for the receiver. This indicates that the information and meaning are probably natural language independent.
For example, the expression of the design of a pump consists of a collection of expressions of atomic facts. The information contained in the expressions should be independent of the natural language in which the design is expressed, so that a pump that is fabricated will be independent of the language that is used for the expressions.

Syntactic structures of natural languages are very diverse and language dependent. A collection of facts can be expressed in a natural language as complex sentences, which can be arranged in a 'story' according to the large variety of constructs that is provided by the grammar of a natural language. In the Gellish subset grammar of a language this variety is limited to such an extent that facts are expressed only as a collection of atomic expressions. So, Gellish English allows only a limited number of grammatical constructs and it has a syntactic structure that is simple and language independent. But nevertheless Gellish English enables to express a lot of what can be expressed in natural English. This is achieved by a strong focus on the semantics and by the reduction of complex sentences to a network of simple, atomic expressions, whereas each atomic expression expresses the semantics of an atomic fact. For example, in Gellish the syntax is simplified in the following ways:

- Verbs are treated in the same way as their corresponding nouns that indicate kinds of occurrences. For example, ‘to act’ and (an) ‘act’ both refer to a kind of occurrence.
- Conjugations are ignored as they duplicate information that is already captured in other ways, such as the performer(s), the time of occurrence, etc. For example, walk, walks, walking and walked all refer to the same semantic concept of an act of walking, whereas the performer(s) and the time of occurrence are sufficient for a semantic interpretation.
- A combination of an adjective and a noun refers to a subtype of the thing that is indicated by the noun. For example, the combination ‘centrifugal’ and ‘ pump’ refers to a subtype of pump, called ‘centrifugal pump’, which subtype has an operating principle that is by definition centrifugal.
- Indirect references such as by the words ‘that’, which, who, etc. as well as implicit references are replaced by explicit direct references in separate expressions. For example, the complex natural language sentence ‘P-1 is a pump which is centrifugal’ contains an indirect reference represented by the word ‘which’ and an implicit reference to the operating principle of P-1, which principle is qualified as centrifugal. Therefore, the sentence is replaced in Gellish by four atomic sentences: ‘P-1 is classified as a pump’ and ‘P-1 has as aspect A1’, ‘A1 is classified as operating principle’ and ‘A1 is qualified as centrifugal’. Although this seems to add complexity it simplifies the syntax and simplifies the interpretation.
- The concept ‘a’ which precedes a kind of thing indicates an implicit reference to an individual thing that is classified by the kind of thing. This implicit thing is made explicit in Gellish. For example, ‘a height’ indicates an individual aspect, say h1, that is classified as height.

So, in natural English we can have a complex expression such as:

- the Eiffel tower has *a height which* is 300 m

In Gellish English the same complex fact can be expressed, but in three separate atomic expressions:

-	the Eiffel tower	has aspect	h1
-	h1	is classified as a	height
-	h1	is qualified as	300 m

Note that in the Gellish English expressions:

- The individual aspects are made explicit.
- The classification of the individual things is made explicit. This is equivalent to the concept ‘a’ as a predecessor of a kind.
- The relations between the atomic expressions is implied by the fact that things are reused at different lines. This reuse makes concepts such as ‘which’ and ‘and’ superfluous.

The explicit nature of Gellish English has the following consequences:

- Each atomic expression can be interpreted independent of the others.
- The sequence in which the atomic expressions are provided is irrelevant. This enables that additional atomic expressions can be added to the collection at any location in the list, so that information integration simply means combination (union) of two collections of atomic expressions. This is more powerful than what is required for integration of two stories in a natural language!

We will now eliminate the expression in a natural language and focus on the identification of *natural language independent semantic concepts*, especially those that enable to express facts. After that we will discuss how those semantic concepts are expressed in natural languages and in the Gellish subsets.

As is described above, it appears that kinds of relations determine the meanings of the atomic expressions of facts about the related things. These kinds of relations are by nature not dependent on their names in natural languages. Therefore, apparently natural language independent concepts of kinds of relations exist. For ease of language independent reference to them we will give them a unique Gellish identifier (UID). The value of such a UID is in itself meaningless. It only identifies a natural language independent semantic concept. The definition of the meaning of such a concept is given by relations between the concepts. In order to be able to communicate between people in different languages it is required that the names of each concept in the various languages is associated with the unique identifier of the concept. For example, the concepts identified by their unique Gellish identifier and their English names are:

- 1727	has in English the name	has aspect
- 550126	has in English the name	height
- 1225	has in English the name	is classified as a
- 5020	has in English the name	is qualified as

The concepts have other names in other languages and have other textual definitions (with shall have identical meanings) in those languages, but their relations to other concepts are language independent! It we assume that the UID of the Eiffel tower is 111000111, the UID of h1 is 111000112 and the UID of 300 m is 111000113, then the above message becomes in full Gellish English:

- 111000111	the Eiffel tower	1727	has aspect	111000112	h1
- 111000112	h1	1225	is classified as a	550126	height
- 111000112	h1	5020	is qualified as	111000113	300 m

This means that in Gellish it is possible to exchange information using concept 550126, 1225 and 5020 without mentioning their name or their textual definition in any language, once the sender and the receiver 'know' which concepts are meant, because they possess a Gellish dictionary that provides the names of those concepts in their own languages.
It also means that a message that is sent using the names of concepts in one language, can replace those names by the names of those concepts in his own language, using his Gellish dictionary.
For example, a Dutch receiver of the above English message about the height of the Eiffel tower can apply Browser software that can automatically translate the message on the basis of the information in his own Gellish Dutch dictionary. If we assume that that receiver has no Dutch name for the concepts Eiffel tower, h1 and 300 m in his Gellish Dutch dictionary, then the message can be displayed in Dutch as:

- the Eiffel tower	heeft aspect	h1
- h1	is geclassificeerd als een	hoogte
- h1	is gekwalificeerd als	300 m

The semantics of those language independent concepts is defined in the Gellish language as well. For example, the definition of concept 1225 (in English named as 'is classified as a') is defined as follows: it has a role-1 (in English at the left hand of the expression) that shall be played by an individual thing and a role-2 (in English at the right hand of the expression) that shall be played by a kind of thing. Thus the semantics is contained in the kind of relation that is used in an expression.

Examples of the definition of semantics of kinds of relations that are intended to express knowledge are:

- individual thing	can be classified as a	kind of thing
- individual thing	can be composed of	individual thing
- thing	can an elements of	collections of things
- totalities	can possess an	aspect
- *etc.*		

Other kinds of semantics do not deal with knowledge about possible facts, but about realities about individual things. For example, expressions that are similar to:

- *an individual thing*	is classified as a	individual thing
- *an individual thing*	is a part of	*an individual thing*
- *etc.*		

All these concepts have language independent unique identifiers in Gellish.

Further research is required to determine whether the Gellish approach can help in natural language interpretation. Maybe the tabular presentation of small expressions as is used in Gellish English can be transformed into more complex natural English sentences, for example by building on the results of the work of Noah Chomsky. This would require that the syntactical transformation makes use of the Gellish semantic concepts that are expressed as phrases such as 'is a part of' in stead of words, as are currently used as a basis in Chomsky's work. If that appears to be possible, then the inverse process of natural language interpretation might benefit from the semantic concepts identified in the Gellish language. Gellish enables a man-machine interface on the basis of simple natural language expressions. Further work might expand this to a man-machine interface that is based on an extended part of natural language interpretation.

8 Conclusions and recommendations

It is obvious that many people are of the opinion that the development of a general applicable model or computer interpretable language is impossible or at least an endless task. However, natural language, in combination with engineering design conventions (e.g. for drawings, tables and standard forms) has that capability, although a natural language is too flexible and allows too many variation in expressions and ambiguity to make them suitable for unambiguous computer interpretation. These research results illustrate that a (standardized) formal subset of natural language grammar, in combination with a systematic dictionary / taxonomy can result in a subset of a natural language that combines wide applicability with unambiguous computer interpretability. It is up to the reader of this document to judge the generality and the applicability for his or her applications.

From this research the following conclusions can be drawn:

- Gellish English (as any other Gellish variant) is relatively easy to grasp because it is self contained and does not make use of a meta-language or a meta-meta-language as is common practice in information technology. Furthermore, because it is a structured subset of natural English and avoids typical information technology terminology and naming conventions (section 1.2).
- The Gellish language specification in combination with the standard Gellish Table can be used as a standard general interface specification for the exchange of application data between systems. Such an interface would be application system independent (section 5).
- The Gellish language is a further development and integration of the ISO 10303-221 (AP221) and ISO 15926-2 data exchange and data integration data models and of the ISO 15926-4 ‘reference data library’. It has extended semantics, is simpler to implement and integrates the data models and the dictionary / taxonomy / knowledge base (section 1.2).
- The use of natural language independent identifiers in Gellish enables the use of synonyms and homonyms, as well as presentation of product information, standard specifications and expressed knowledge into any language for which a Gellish dictionary is available. Such a translation leaves the models unchanged, because the models are expressed as language independent relations between identifiers (section 2.2).
- The Gellish language can replace and integrate data models. Such a replacement would enable the merging and integration of the data stored according to existing data models and makes the creation of new data models largely superfluous (section 4.2 and 4.3).
- The Gellish language is more flexible than conventional (fixed) data models and it is easier to add semantics to the language, simply by adding definitions of kinds of relations to the Gellish dictionary / taxonomy (section 2.4.3).
- The Gellish dictionary / taxonomy is suitable as standard reference data, to be stored for example as instances of legacy data models (e.g. in ERP systems). It then provides common terminology and/or standard classifications for kinds of things that can be used to customize existing systems. If required, a subset of concepts can be selected, with its own grouping and/or hierarchy, especially if no use is made of the inheritance capabilities of the Gellish taxonomy (section 5.3.2).
- The Gellish language is equivalent to a very large data model and is more than a data model, because it allows semantically correct expressions of facts for which no predefined data structure at class level is available (section 4.2).
- The Gellish language enables the consistency verification between the expression of knowledge, standard specifications (standardized requirements), product design information and actual product information (as observed) (section 3.1 and 6.3.1).

- The Gellish language can be implemented as a Gellish Table database with good performance, as is proven for medium size data volumes by the performance of the Gellish Browser. Further research on the implementation of large databases should clarify possible implementation constraints (section 5.1).
- Although the Gellish language is developed with a focus on technical artifacts and their functions, processes and behavior, including component catalogues as well as integration of data about large assemblies such as machines, transport equipment, civil structures and process plants, it is validated and applicable at least also on the modeling of business processes and commercial transactions (section 6.3.3).
- The Gellish language is human readable and may provide a basis for the development of textual and verbal man-machine interfaces, as well as for intelligent agent software that interprets and acts on incoming messages (section 5.6 and 6.3.3).

A philosophically justified model is never completed, because philosophers (ontologists) will never stop thinking about the structure of reality. Furthermore, many philosophers depart from different axioms and ‘schools of thought’ and it seems reasonable to expect that that will remain to be the case. Similarly, a language is never completed, because new concepts arise and things are expressed differently over time, especially about the expression of emotions and opinions. However, there appears to be light at the horizon, as it seems that there is a basic grammar and language independent semantic primitives that does not change over time, especially with respect to the expression of facts about physical phenomena, technical artifacts and structured business processes. This research has identified many of those stable semantic primitives and the Gellish language is integration of them into a coherent subset language. Further research and development may merge various other ontologies, dictionaries and product model libraries into the Gellish dictionary / ontology / knowledge base and thus extend the expression power of the Gellish language.

Further research and integration of the results of general linguistics, such as the semantic analysis of Anna Wierzbicka, may result in an extension of the semantic and grammatical richness of Gellish, which may reduce the gap with the natural languages, without introducing ambiguity nor loosing computer interpretability.

9 Literature

1. Andries van Renssen (1998), **Guide on the extension of Gellish,** http://sourceforge.net/projects/gellish.
2. Andries van Renssen (1999), **Implementation of the Management of Information,** see above.
3. Andries van Renssen (2005), **Gellish Application Handbook,** see above.
4. Andries van Renssen (2002), **The Gellish language for the Semantic Web,** see above.
5. Alvin Plantinga (2000), **Warranted Christian Belief**, Oxford University Press, New York.
6. Anna Wierzbicka (1996), **Semantics, Primes and Universals**; Oxford University Press, Oxford.
7. Asuncion Gomez-Perez, Mariano Fernandez-Lopez and Oscar Corcho (2004), **Ontological Engineering** with examples from the area of Knowledge Management, e-Commerce and the Semantic Web, Springer-Verlag London.
8. Barry Smith (1999), **Ontology: Philosophical and Computational**, http://wings.buffalo.edu/philosophy....../smith/articles/ontologies.htm.
9. Barry Smith (1986), **Parts and Moments**, Studies in Logic and Formal Ontology, Pholosophia Verlag, Munchen.
10. Brickley D., Guha R.V. (2003). **RDF Vocabulary Description Language** 1.0: RDF Schema. W3C Working Draft. http://www.w3.org/TR/PR-rdf-schema.
11. Dean M., Schreiber G. (2003). **OWL Web Ontology Language Reference**, W3C Working Draft. http://www.w3.org/TR/owl-ref/.
12. Dooyeweerd (1953), '**A new critique of theoretical thought**', ISBN 0-88815-152-7, 2nd (English) edition, Paideia Press Ltd., Jordon Station, Ontario, Canada (4 parts in two volumes). This is an extended and revised version of the original Dutch edition of 1935/1936 with the title '**De Wijsbegeerte der Wetsidee**'.
13. Eli Hirsch (1982), The concept of identity, The Oxford University Press, New York.
14. Ludwig Wittgenstein (1961), '**Tractatus Logico-Phylosophicus**', Routledge & Kegan Paul Ltd., London. Originally published in German. Dutch translation and original German text, by W.F. Hermans, Atheneum, Amsterdam, 6e druk, 1998.
15. **E-cl@ss**, see www.eclass.de.
16. **Gellish dictionary / taxonomy** (STEPlib), available free of charge under open source license conditions via http://gellish.sourceforge.net/.
17. Genesereth M.R., Fikes R.E., (1992). **Knowledge Interchange Format**, Version 3.0, Reference Manual. Technical Report Logic-92-1, Computer Science Department, Stanford University, California.
18. Gielingh, W.F. (1988). **General AEC Reference Model** *(GARM)*. TNO Building and Construction Research, BI-88-150, Delft, The Netherlands.
19. Gruber T.R. (1992). **Ontolingua**: A Mechanism to Support Portable Ontologies. Technical report KSL-91-66, Knowledge Systems Laboratories, Stanford University, California.
20. Hart (1984), '**Understanding our World, an integral ontology**', ISBN 0-8191-4257-3 (or 0-8191-4258-1), University Press of America, Lanham, United States.
21. Horrocks I., Fensel D., van Harmelen F., Decker S., Erdmann M., Klein M., (2000). **OIL in a Nutshell**. In Dieng R., Corby O. (eds) 12th International Conference on Knowledge Engineering and Knowledge Management (EKAW'00), Juan-Les-Pins, France. Springer-Verlag, Berlin, Germany., pp. 1-16.

22. **IEC 1131-3** (1985), International Electrotechnical Committee (an international standardization organization for the electrotechnical industry), standard nr IEC 1131-3.

23. ISO/IEC 7498-1 (1994), **Open Systems Interconnection (OSI) – Reference Model**.

24. Jan Dietz (1996), **Introductie tot DEMO**, door Prof. Dr. ir. J.L. Dietz, Samson Bedrijfsinformatie, Alphen aan de Rijn (in Dutch).

25. Jan Dietz (2002), **The Atoms, Molecules and Fibers of Organizations**, Delft University of Technology.

26. John Mylopoulos at al (1990), **TELOS: Representing Knowledge About Information Systems**, ACM Transactions on Information Systems, Vol. 8 No. 4, October 1990, Pages 325-362.

27. John F. Sowa (1984), **Conceptual Structures**: Information Processing in Mind and Machine; Addison-Wesley Publishing Company Inc.

28. John F. Sowa (2000), **Knowledge Representation**, Logical, Philosophical and Computational Foundation, Brooks/Cole, Pacific Grove.

29. Martin Stokhof (2000), **Taal en betekenis**; Boom, Amsterdam (in Dutch).

30. Matthew West (1994), **Developing High Quality Data Models**, vol. 1, 2 and 3, Report No. IC94-033/034/035, Shell International Petroleum Company Ltd., London, March 1994.

31. Mike Uschold, Martin King, Stuart Moralee and Yannis Zorgios (1997), **The Enterprise Ontology**, AIAI, The University of Edinburgh, United Kingdom. m.uschold@ed.ac.uk; martin.king@uk.ibm.com; moralee.s@uk.co.urpsl; tcsyzz@uk.co.lreg.aie.

32. NIST (1993), **Integration Definition for Function Modelling (IDEF0)**, Federal Information Processing Standards Publication 183, NIST, December 1993.

33. Noam Chomsky (1957), **Syntactic Structures**, 12th ed.; Mouton, The Hague – Paris, 1957; ISBN 90 279 3385 5.

34. **OWL Web Ontology Language, Overview** (2003), www.w3.org/TR/owl-features/

35. Pease, A., Niles, I., (2002), **IEEE Standard Upper Ontology**: A Progress Report. The Knowledge Engineering Review 17(1):65-70.

36. Pease, A., Niles, I and Li, J. (2002), **The Suggested Upper Merged Ontology**: A large Ontology for the Semantic Web and its Applications. In Working Notes of the AAAI-2002 Workshop on Ontologies and the Semantic Web, Edmonton, Canada, July 28-August 1, 2002.

37. Peter Kroes (1996), **Technical and Contextual Constraints in Design**; an essay on determinants of technological change. In 'The role of design in the shaping of technology', Proceedings from the COST A4 workshop, Lyon, France, February 1995.

38. Peter Simons (1986), **Number and Manifolds**, in Parts and Moments, Studies in Logic and Formal Ontology, (Barry Smith, ed), Pholosophia Verlag, Munchen.

39. Rene van Woudenberg (2000), **Het mysterie van de identiteit**; Sun, Nijmegen (in Dutch).

40. **Rosettanet**, see www.rosettanet.org.

41. Rumbaugh J., Jacobson I., Booch G. (1998), **The Unified Modeling Language Reference Manual**, Addison-Wesley, Boston, Massachusetts.

42. Stafleu (1989), **De verborgen structuur**, ISBN 90-6064-715-7, Buijten & Schipperheijn, Amsterdam (in Dutch).

43. Stafleu (2002), **Een wereld vol relaties,** Karakter en zin van natuurlijke dingen en processen, Buijten & Schipperheijn, Amsterdam (in Dutch).

44. Terry Halpin (1996), **Business Rules and Object Role Modelling (ORM)**, in Database Programming & Design, October 1996.

45. Thomas R. Gruber and Gregory R. Olsen (1994), **An Ontology for Engineering Mathematics**, in Jon Doyle, Pierro Torasso & Erik Sandewall, eds., Fourth International Conference on Principles of Knowledge Representation and Reasoning, Gustav Stresemann Institute, Bonn, Germany, Morgan Kaufmann. gruber@ksl.stanford.edu.

46. Tim Berners-Lee, James Hendler and Ora Lassila (2001), **The Semantic Web**, Scientific American, May 2001, http://www.sciam.com/2001/0501issue/0501berners-lee.html.

47. United Nations Standard Products and Services Code (**UNSPSC**), www.unspsc.org

48. Van Griethuysen, J.J. (ed.) (1982), **Concepts and Terminology for the Conceptual Schema and the Information Base**, ISO TC97/SC5/WG3, Eindhoven.

49. Van Reijswoud (1999), **DEMO modeling Handbook**, V.E. van Reijswoud and J.L.G. Dietz, www.demo.nl/documents/handbook.pdf.

50. VDI/VDE 3696 (1995), **Manufacturer independent configuration of digital control systems**, part 1, 2 and 3, Verein Deutsche Ingenieure / Verband Deutscher Electrotechniker, VDI/VDE-Gesellschaft Mesz- und Automatisierungstechnik (ICS 25.040.40), Dusseldorf (in German).

10 Appendices

10.1 Appendix A, Upper ontology specialization hierarchy with played roles in relations

The formal definition of the upper ontology part of the Gellish language is documented in a Gellish Table with the upper ontological facts (the TOPini table). The relations that express those facts as relations between concepts define the grammar of the Gellish language.

This appendix presents a view on the upper ontology part of the Gellish language definition in English. The view is generated by an automated procedure from the Gellish Table with the formal definition.

This view presents:

- The specialization/generalization (subtype/supertype) hierarchy of concepts in the upper ontology part of the Gellish language definition.
- The relations between those concepts and the kinds of roles that those concepts play.
- The relations between those kinds of roles and the kinds of relations that require those roles.

More precisely said, the second item presents the kinds of roles that classify roles that are either played by individuals that are members of the concepts or that define the nature of roles that are played by classes when knowledge is expressed as relations between classes.

Note that all the subtypes of these concepts inherit from their parents what kinds of roles they or their members can play and thus what kinds of relations they can have according to the grammar of the Gellish language. For example, the concept ‘pump’ is a sub-subtype of physical object and that is a sub-subtype of anything. That means that an individual pump can have all the roles inherited from both supertypes that are applicable for members of those concepts and thus can have relations of the kinds that require those roles (not the conceptual roles that are applicable for relations between classes).

This appendix does not present the branch of the hierarchy that forms the concept hierarchy of roles, nor the branch with the hierarchy of relation types. The latter is presented in Appendix B.

There are also other relations between these concepts defined that are not presented, in particular synonyms of concept names. The full Gellish dictionary/taxonomy contains subtypes of the concepts in this view and contains generally recognized individual things that are classified by one or more concepts that are defined in the specialization/generalization hierarchy.

It should be noted that the Gellish language is regularly extended, so that the actual status of the language definition should be retrieved from the website.

Specialization hierarchy (players of roles)	Roles (played in relations)	Identifier of relation type	Gellish relations	Inverse Gellish relations
anything				
	guarded	1,533	is custodian of	is guarded by
	referenced	1,770	is a reference for	is referenced by
	derived	1,269	is derived from	is original of
	source for derivation	1,269	is derived from	is original of
	predecessor	1,385	is a successor of	is a predecessor of
	successor	1,385	is a successor of	is a predecessor of
	original	1,393	is a version of	is a previous version of
	version	1,393	is a version of	is a previous version of
	about	1,273	is information about	is described via information
	referenced within information	1,369	is referred in	includes a reference to
	commonly about	1,911	with as information a	can be an information about
	included	1,912	can be included in a	can include
	conceptually referenced	1,919	can be an indicator of a	can be indicated by a
	uniqueness context for common reference	1,920	shall be unique in the context of	is a uniqueness context for a
	represented	2,071	is represented by	represents
	described	4,682	is described by	is a description of
	defined	4,685	is defined by	is a definition of
	relator	2,850	might be related to	might be related with
	uniqueness context for individual reference	4,776	is unique in context of	is a uniqueness context for
	collected	2,846	is collected in	is a collection for
	application context	4,818	is applicable in the context of	is a context for validity of
	conceptually occurred	5,339	can occur at a	can be an occurrence date of a
	related	2,850	might be related to	might be related with
	conceptual possessor	2,069	can have a	can be an aspect of a
	conceptually guarded	5,004	can be a custodian of a	can be guarded by a
	composing list items	5,314	are components in list	is a list of
real object				
concept				
	conceptual player of a role	4,714	can have a role as a	can be a role for a
	supertype	1,146	is a specialization of	is a generalization of
	subtype	1,146	is a specialization of	is a generalization of
	classified class	1,224	is a particular	is a kind of class of
	common possessor	2,070		
	qualifier	1,726	is a qualification of	is the nature of
	nature	1,726	is a qualification of	is the nature of
	common possessor of reference aspect	2,018		
	conceptual function	4,648	can have a role in a	can involve a
	collected class	4,730	is an element in collection of classes	is a collection of classes including
	conceptually classified	4,991	can be classified by a	can classify a
	conceptual classifier	4,991	can be classified by a	can classify a
	constraining class	5,095	each of which is a specialization of	is a generalization of each element of
	related class	4,719	is related to a	can be related with
	conceptual possessor of a role	5,229	has conceptually a role as	is conceptually a role of
abstract object				
class				
	classifier for a class	1,224	is a particular	is a kind of class of
conceptual class				
qualitative class				
qualitative aspect				
plural object				
	united	2,853	is disjoint union of	contains elements that are united in
	union	2,853	is disjoint union of	contains elements that are united in
	collecting plurality	2,846	is collected in	is a collection for
	collectively classified	4,843	is classifier of each element of	each of which is classified as a
	difference collection	2,847	is the difference of sets	are sets with as difference
	compared collections	2,847	is the difference of sets	are sets with as difference
	composed list	5,314	are components in list	is a list of

Specialization hierarchy (players of roles)	Roles (played in relations)	Identifier of relation type	Gellish relations	Inverse Gellish relations
plural individual				
	classifier for a plural individual	5,043	are collectively classified as a	is classifier of collection
	classified plural individual	5,043	are collectively classified as a	is classifier of collection
	collecting individual plurality	1,227	is an element of	is a plural individual for
plural aspect				
	options	4,850	is qualified as one of the	are options for qualification of
	compliancy criteria for members of class	4,950	are specifications for a	shall commonly be compliant with one of the
collection of information elements				
	collective informer	5,047	each of which includes information about	is described within information collection
collection of properties				
list of properties				
collection of roles				
options				
set of records				
collection of physical objects				
collection of materials				
collection of items				
collection of occurrences				
to occur				
collection of classes				
	common options	4,846	can be either of the	are options for a
	collecting plural class	4,730	is an element in collection of classes	is a collection of classes including
	constrained collection	5,095	each of which is a specialization of	is a generalization of each element of
collection of qualitative aspects				
collection of conceptual aspects				
plural relation				
options for qualification of an aspect			is qualified as one of the	are options for qualification of
	optionally qualified	4,850	is qualified as one of the	are options for qualification of
	options	4,850	is qualified as one of the	are options for qualification of
list				
	defining list of items	5,296	is defined by the items in list	is the list that defines the items of
	defining list of possessed aspects	5,297	is defined by the possessed aspects in list	is the list that defines the possessed aspects of
table				
	defined table by items	5,296	is defined by the items in list	is the list that defines the items of
	defined table by possessed aspects	5,297	is defined by the possessed aspects in list	is the list that defines the possessed aspects of
physical object - aspect table				
equipment summary				
kind of physical object - aspect table				
list of physical objects				
list of kinds of aspects				
list of qualitative aspects				
list of units of measure				
list of kinds of physical objects				
list of kinds of possessed aspects				
list of aspects				
structured collection				
	arranged collection	4,661	is arranged in	is arrangement for
single object				
single individual				
relation			might be related to	might be related with
	conceptual requirer of a role by a relation	2,076	can require as role a	can be required by a
	relator	2,850	might be related to	might be related with
	related	2,850	might be related to	might be related with
	applicable	4,818	is applicable in the context of	is a context for validity of
	intended relation for purpose	4,889		
	conceptual involver in relation	4,900		
individual object				
	conceptually referenced within information	1,370	can be referenced within a	can include a reference to a

Specialization hierarchy (players of roles)	Roles (played in relations)	Identifier of relation type	Gellish relations	Inverse Gellish relations
	conceptually described	1,910	can be described via a	can be information about a
	conceptually referred	1,917	can be a referencer for a	can have as reference a
	conceptual possessor of structure	4,795	can have as structure a	can be a structure of a
	classifier for an individual	1,225	is classified as a	is a classifier of
	conceptual part	1,261	can be a component of a	can be a whole for a
	conceptual whole	1,261	can be a component of a	can be a whole for a
	common part for individual	1,728	has a part that is classified as a	is a classifier of a part of
	conceptual possessor of reference aspect	1,799	can have as reference aspect a	can be a reference aspect of a
	conceptual collected whole	1,228	can be an element of a	can be a plural whole for a
	conceptual assembled whole	1,191	can be a part of a	can have as part a
	classified individual	1,225	is classified as a	is a classifier of
	whole	1,260	is a component of	is totality of
	part	1,260	is a component of	is totality of
	possessor of an aspect	1,727	has aspect	is an aspect of
	whole for kind of part	1,728	has a part that is classified as a	is a classifier of a part of
	possessor of time aspect	4,800		
	assembled whole	1,190	is a part of	is whole of
	possessor of reference aspect	1,768	has as reference aspect	is a reference aspect of
	representer	2,071	is represented by	represents
	conceptual fulfiller of function	1,314	can fulfil a function as a	can be a function of a
	fulfiller of function	1,969	fulfils function	is a function of
	conceptually quantified	2,047	can be quantified by number of items	can be a quantification of a
	conceptually specified	4,751	shall be a specification for a	shall be compliant with a
	required complier	4,753	shall individually be compliant with	is a compliancy criterion for
	commonly described in information carrier	4,723	contains information about a	can be described in
	conceptually described in an information carrier	4,726	can contain information about a	can be described in a
	function	4,767	is involved in	involves
	operand	4,825	can be an operand in a	can have as operand a
	collective classifier	4,843	is classifier of each element of	each of which is classified as a
	happened	4,871	occurs within	is time frame of
	individually related	4,658	is related to	is related with
	individual relator	4,658	is related to	is related with
	conceptual subject of correlation	4,922	can have as correlation a	can be a correlation for a
	conceptually represented	4,924	can be represented by a	can represent a
	conceptually representing	4,924	can be represented by a	can represent a
	conceptually compliant	4,902	can contain a criterion for a	can be compliant with a
	commonly compliant	4,950	are specifications for a	shall commonly be compliant with one of the
	conceptually involved in relation	4,900		
	relating individual	4,719	is related to a	can be related with
	collected individual	1,227	is an element of	is a plural individual for
	possessor of purpose	1,366	is existing for purpose	is the purpose for existence of
	conceptual possessor of purpose	1,609	can have as purpose a	can be a purpose of a
single individual				
specific individual				
typical individual				
whole individual				
physical object				
	classifier for a physical object	1,286		
	conceptually connected from	1,407	can be connected to a	can have a connection with a
	conceptually connected to	1,407	can be connected to a	can have a connection with a
	conceptual destination	1,409	can be a destination of a	can end at a
	conceptual container of route	1,411	can contain as route a	can be a route in a
	common possessor of decomposition structure	1,412		
	common possessor of topologic sequence structure	1,413	can have as arrangement a	can arrange a
	conceptually segregated	1,418	can be segregated from a	can be segregated (inverse) from a
	conceptually segregated from	1,418	can be segregated from a	can be segregated (inverse) from a
	conceptual source	1,419	can be a source of a	can have as source a
	commonly used for segregation	1,422	can be used in segregation	can use in segregation a

Specialization hierarchy (players of roles)	Roles (played in relations)	Identifier of relation type	Gellish relations	Inverse Gellish relations
	conceptual follower of route	1,529	can follow a	can be followed by a
	connected member	1,629	is connected with a	is classifier of item connected with
	used member	1,664	can be used in connection	is a connection using a
	conceptually used	1,724	can use as connection material a	can be connection material in a
	conceptual protector	1,415	can be protected by a	can protect a
	conceptually protected	1,415	can be protected by a	can protect a
	conceptual topologic predecessor	1,421	can be located after a	can be located before a
	conceptual topologic successor	1,421	can be located after a	can be located before a
	conceptual whole physical object	1,437		
	conceptual part of a physical object	1,437		
	conceptually referring physical object	1,715	can refer to a	can be referred from a
	conceptually referred physical object	1,715	can refer to a	can be referred from a
	common role player	1,900	can be the role of a	can have as role
	conceptually installed	1,903	can be installed at the position of a	can be a position for installation of a
	classified physical object	1,286		
	installed	1,313	is installed for	is place of installation of
	whole physical object	1,436		
	part of a physical object	1,436		
	connected	1,487	is connected with	is connected to
	destination	1,490	is the destination of	has as destination
	protected	1,497	is protecting	is protected by
	protector	1,497	is protecting	is protected by
	segregated	1,504	is segregated from	is segregated from (inverse)
	segregated from	1,504	is segregated from	is segregated from (inverse)
	departure location	1,505	is the source of	has as source
	duplicated	1,506		
	redundant	1,506		
	topologic predecessor	1,508	is located after	is located before
	topologic successor	1,508	is located after	is located before
	used for segregation	1,509	is using for segregation	is used in segregation
	connection material	1,510	is using as connection material	is connection material in
	follower of route	1,527	follows route	route is followed by
	connected individual	1,629	is connected with a	is classifier of item connected with
	referring physical object	1,713	refers to	is referred from
	referred physical object	1,713	refers to	is referred from
	possessor of decomposition structure	1,767	has as decomposition structure	is decomposition structure of
	container of route	1,787	is a route through	contains route
	possessor of topologic sequence structure	1,789	has topological structure	is topologic structure of
	player of a role	5,234	is a role of	is player of
	whole physical object with common role	1,906	can be a role of a part of	can be the whole for a part with as role a
	individual whole for kind of physical feature	1,908		
	possessor of capability	1,972	has the capability to act as a	is a kind of role that can be fulfilled by
	subject of correlation	4,886	has as correlation	is correlation for
	content	4,692	is contained by	is container of
	container	4,692	is contained by	is container of
	individual information carrier	4,723	contains information about a	can be described in
	conceptual information carrier	4,726	can contain information about a	can be described in a
	conceptual carrier	4,810	can be a displayer of a	can be displayed on a
	conceptually carried	4,810	can be a displayer of a	can be displayed on a
	possessor of an aspect of a part	5,248	has a part with aspect	is an aspect of a part of
	conceptually contained	4,942	can be contained by a	can be a container of a
	conceptual container	4,942	can be contained by a	can be a container of a
	presenter of information	4,996	is presented on	is presenter of
	conceptual presenter of information	4,999	can be presented on a	can be a presenter of a
	potentially involved	5,066		
	performer	4,761	is performer in	is performed by
	addressed	5,070	has address	is address of
	conceptual possessor via part	5,247	can have a part with as aspect a	can be an aspect of a part of a

Specialization hierarchy (players of roles)	Roles (played in relations)	Identifier of relation type	Gellish relations	Inverse Gellish relations
	conceptual possessor via assembly	5,249	can have a whole with as aspect a	can be an aspect of a whole of a
imaginary physical object				
	functional position	1,313	is installed for	is place of installation of
	conceptual installation position	1,903	can be installed at the position of a	can be a position for installation of a
materialized physical object				
route				
	whole route	1,439		
	terminated route	1,490	is the destination of	has as destination
	started route	1,505	is the source of	has as source
	followed route	1,527	follows route	route is followed by
	contained route	1,787	is a route through	contains route
	conceptually terminated route	1,409	can be a destination of a	can end at a
	conceptually started route	1,419	can be a source of a	can have as source a
	conceptually followed route	1,529	can follow a	can be followed by a
	conceptual whole route	1,405		
	conceptually contained route	1,411	can contain as route a	can be a route in a
binary encoded object				
	binary representator			
ASCII encoded binary				
	binary representator of text			
physical feature				
	conceptual feature part	1,410	can be a feature of a	can have a feature like
	used member feature	1,710		
	common feature part of an individual	1,908		
	feature part	1,492	is feature of	has feature
	connection feature			
matter				
material				
	reference location	4,668	is location of	is located at
solid item				
artefact				
assembly				
connection assembly				
	connecting assembly	1,487	is connected with	is connected to
	conceptual part of route	1,405		
	conceptual user of connection material	1,724	can use as connection material a	can be connection material in a
	part of route	1,439		
	user of connection material	1,510	is using as connection material	is connection material in
area (surface)				
	address	5,070	has address	is address of
inanimate physical object				
physical space				
physical point				
annotation element				
	carried	1,427	is displayed on	displays
	source annotation	1,449		
	derived annotation	1,449		
	invisible in view	1,581		
point marker symbol				
connection assembly of annotation elements				
	using annotation connection	1,526		
symbol				
connector annotation element				
page connector				
off-page connector				
on-page connector				
intrapage connector				
	referring intrapage connector	1,513		

Specialization hierarchy (players of roles)	Roles (played in relations)	Identifier of relation type	Gellish relations	Inverse Gellish relations
	referred intrapage connector	1,513		
text				
annotation area				
annotation curve				
annotation point				
terminator symbol				
language construct				
expression				
mathematical expression				
lifeform				
	adopter	4,813		
	provider	4,916	is provider of	is provided by
organization				
	approver	1,187	is approver of	is approved by
	custodian	1,533	is custodian of	is guarded by
	controlling authority for common reference	1,918	is the controlling authority of a	is a kind of reference that is controlled by
	managing organization	4,908	is managing organization of	is managed by organization
	conceptual custodian	5,004	can be a custodian of a	can be guarded by a
	potential supplier	5,154	can be supplier of a	can be supplied by
	potential manufacturer	5,157	can be the manufacturer of a	can be manufactured by
	customer	4,915	is customer in	is required by customer
	manufacturer	5,152	is manufacturer of	is manufactured by
person				
	approver	1,187	is approver of	is approved by
	manager	4,912	is manager of	is managed by
	born	4,866	is born at	is birth date of
	committer	5,037	is committer of	is committed by
	conceptual committer	5,040	can be a committer of a	can be committed by a
	parent	5,284	is a parent of	is a child of
	child	5,284	is a parent of	is a child of
man				
	father	5,287	is a father of	has as father
	son	5,291		is a son of
	husband	990,125	is married with	is married by
woman				
	mother	5,289	is a mother of	has as mother
	daughter	5,293		is a daughter of
	wife	990,125	is married with	is married by
real individual				
aspect				
	parameter	4,962	is a parameter in	has as parameter
	coordinate	1,777	is a coordinate of	has as coordinate
	conceptual part aspect	1,257		
	conceptual whole aspect	1,257		
	classifier for an aspect	1,287		
	commonly possessed	2,070		
	conceptual expressor	1,798	can be symbolized by a	can be a symbolic representation of a
	possessed by part	5,248	has a part with aspect	is an aspect of a part of
	excepted	1,147		
	whole aspect	1,262		
	part of an aspect	1,262		
	classified aspect	1,287		
	fulfiller of required aspect	1,295	fulfills	is fulfilled by
	fulfilled aspect	1,295	fulfills	is fulfilled by
	intended aspect for common purpose	1,365	is intended for a	can be a purpose of
	boundary	4,835	has as boundary	is boundary of
	bounded	4,835	has as boundary	is boundary of
	possessed aspect	1,727	has aspect	is an aspect of

Specialization hierarchy (players of roles)	Roles (played in relations)	Identifier of relation type	Gellish relations	Inverse Gellish relations
	conceptually possessed by part	5,247	can have a part with as aspect a	can be an aspect of a part of a
	quantified aspect	2,044	is quantified as	is quantification of
	conceptually possessed	2,069	can have a	can be an aspect of a
	correlated	4,679	is correlated by	is a correlated to
	conceptual parameter	4,740	can be a parameter in a	can have as parameter a
	common options possessor	4,846	can be either of the	are options for a
	optionally qualified	4,850	is qualified as one of the	are options for qualification of
	interpreted	1,760	is interpreted as	is an interpretation of
	common boundary	4,838	has as common boundary	is a common boundary of a
	commonly bounded	4,838	has as common boundary	is a common boundary of a
	conceptually possessed via assembly	5,249	can have a whole with as aspect a	can be an aspect of a whole of a
actual aspect				
temporal boundary of state				
	caused	1,922	has as effect	is caused by
	located in time	1,785	begins or ends at point in time	is point in time of
	located in space	1,786	begins or ends at location	is location where
	triggered	2,026		
	triggering	2,026		
begin of existence				
end of existence				
point approximation of spatial aspect				
	whole point approximation	1,743		
	possessed point approximation of spatial aspect	1,780	has point approximation	is approximate point of
	commonly possessed point approximation of spatial aspect	1,802	can have as point approximation a	can be a point approximate of a
	conceptual whole point approximation	1,819		
point approximation of mathematical space				
	whole point approximation of mathematical space	1,750		
	approximator	1,754		
	conceptual approximator	1,805	mathematical space can have as a point approximation a	can be a point approximate of a mathematical space
	conceptual whole point approximation of mathematical space	1,818		
mathematical space				
	part of mathematical space	1,746		
	whole mathematical space	1,746		
	approximated	1,754		
	boundary of mathematical space	1,773	has as mathematical boundary	is mathematical boundary of
	bounded mathematical space	1,773	has as mathematical boundary	is mathematical boundary of
	qualified mathematical space	1,823		
	conceptual boundary of mathematical space	1,804	can have as mathematical constraint a	can be a mathematical constraint of a
	conceptually bounded mathematical space	1,804	can have as mathematical constraint a	can be a mathematical constraint of a
	conceptually approximated	1,805	mathematical space can have as a point approximation a	can be a point approximate of a mathematical space
	conceptual part mathematical space	1,817		
	conceptual whole mathematical space	1,817		
	nature of mathematical space	1,823		
	parameter list			
	coefficient list			
	classifier for an aspect	1,287		
	conceptual quantifier	2,047	can be quantified by number of items	can be a quantification of a
	quantifier for aspect	2,044	is quantified as	is quantification of
	conceptual coefficient			
	common quantifier of a characteristic	1,791		
mathematical point				
	succeeding adjoint mathematical point	1,751	is an adjacent point after	is an adjacent point before
	preceding adjoint mathematical point	1,751	is an adjacent point after	is an adjacent point before
	common adjoint mathematical point 1	1,850		
	common adjoint mathematical point 2	1,850		
mathematical curve				
mathematical surface				
mathematical region				

Specialization hierarchy (players of roles)	Roles (played in relations)	Identifier of relation type	Gellish relations	Inverse Gellish relations
representation of time				
	year in Gregorian calendar			
	month in year			
	day in month			
	hour in day			
	minute in hour			
	second in minute			
Gregorian time value				
UTC representation of time				
continuous space				
linear space				
	quantifying scale value	1,733		
pivoted mathematical value				
	lower tolerance			
	nominal value			
	upper tolerance			
double bounded range				
	lower bound			
	upper bound			
lower bounded range				
	lower bound			
upper bounded range				
	upper bound			
number				
	exponent			
	multiplier			
	offset			
	lower tolerance			
	nominal value			
	upper tolerance			
	lower bound			
	upper bound			
	coefficient			
	numerator			
	denominator			
	quotient			
	multiplied			
	product			
	raised			
	exponentiation result			
	log base			
	subject in logarithmic function			
	logarithm			
	base for addition			
	increment			
	sum			
	base for subtraction			
	decrement			
integer number				
	role 1 life lower cardinality			
	role 1 life upper cardinality			
	role 2 life lower cardinality			
	role 2 life upper cardinality			
	role 1 simultaneous lower cardinality			
	role 1 simultaneous upper cardinality			
	role 2 simultaneous lower cardinality			
	role 2 simultaneous upper cardinality			
	year in Gregorian calendar			

Specialization hierarchy (players of roles)	Roles (played in relations)	Identifier of relation type	Gellish relations	Inverse Gellish relations
	month in year			
	day in month			
	hour in day			
	minute in hour			
real number				
	second in minute			
proposition				
role				
	conceptually played role	4,714	can have a role as a	can be a role for a
	conceptually required role by a relation	2,076	can require as role a	can be required by a
	played role	5,234	is a role of	is player of
	played role by individual	1,900	can be the role of a	can have as role
	common role of a part of an individual	1,906	can be a role of a part of	can be the whole for a part with as role a
	conceptual part role	1,978		
	conceptual whole role	1,978		
	possibly fulfilled role of a kind	1,972	has the capability to act as a	is a kind of role that can be fulfilled by
	classifier for a role	1,588	has a role as a	is a kind of role fulfilled by
	part of a role	1,979		
	whole role	1,979		
	possible role	4,709	is a possible role for	has as possible role
	intended role	4,715	has as intended role	is the intended role for
	conceptually possible role	5,229	has conceptually a role as	is conceptually a role of
usage				
property space				
	coordinate system	1,876	is a coordinates point in	is the coordinate system for
3D property space				
3D space				
4D property space				
4D space-time				
2D property space				
2D space				
2D box				
clipping box				
text box				
characteristic				
	conceptual source list	1,271		
	conceptually derived	1,271		
	commonly quantified characteristic	1,791		
	conceptually referenced characteristic	1,851	can be compared to a	can be a reference for comparison of a
	conceptually compared characteristic	1,851	can be compared to a	can be a reference for comparison of a
	source characteristic list	1,270		
	derived characteristic	1,270		
	compared	1,756	is compared with	is comparison for
	referenced characteristic	1,756	is compared with	is comparison for
	possessed comparison	1,771	has comparison of	is comparison of
	quantified characteristic	1,733		
	compared characteristic for class	4,892		
	reference common characteristic for comparison	4,892		
	conceptually quantifiable characteristic	4,757	can be quantified on scale	can be a scale for a
spatial aspect				
	common possessor of point approximation of spatial aspect	1,802	can have as point approximation a	can be a point approximate of a
	conceptually possessed spatial aspect	1,711	can have as shape a	can be a shape of a
	part of spatial aspect	1,749		
	whole spatial aspect	1,749		
	possessor of point approximation of spatial aspect	1,780	has point approximation	is approximate point of
	place of begin or end	1,786	begins or ends at location	is location where
	quantified spatial aspect	1,779		
direction range 2d				

Specialization hierarchy (players of roles)	Roles (played in relations)	Identifier of relation type	Gellish relations	Inverse Gellish relations
vector in space				
space				
1D space				
curve				
centreline				
	terminated spatial curve	1,514		
inner boundary				
	possessed inner boundary	2,002		
outer boundary				
	possessed outer boundary	2,003		
2D space				
2D box				
clipping box				
text box				
3D space				
spatial point				
	preceding point	1,744	is an adjacent position after	is an adjacent position before
	succeeding point	1,744	is an adjacent position after	is an adjacent position before
	displaced	1,772	is positioned relative to	is reference for position of
	reference point	1,772	is positioned relative to	is reference for position of
	conceptually displaced	1,889	can be positioned relative to a	can be a reference for the position of a
	conceptual reference point	1,889	can be positioned relative to a	can be a reference for the position of a
	common adjoint before	1,891	is by definition adjacent after	is by definition adjacent before
	common adjoint after	1,891	is by definition adjacent after	is by definition adjacent before
	conceptually located	1,801	can be a coordinate in a	can have as coordinate a
	terminator spatial point	1,514		
temporal aspect				
	classifier for a temporal aspect	1,781		
	conceptually possessed time aspect	1,807	can have as time aspect a	can be a point in time of a
	conceptually possessed time aspect of state	1,551		
	part of temporal aspect	1,748		
	whole temporal aspect	1,748		
	possessed time aspect	4,800		
	classified temporal aspect	1,781		
	time of begin or end	1,785	begins or ends at point in time	is point in time of
	time of state change	1,762		
period in time				
	ending period in time	1,281	terminates at	is termination time of
	starting period in time	1,384	begins at	is starting time of
	conceptual starting period in time	1,808	can start at a	can be a starting time of a
	conceptual ending period in time	1,809	can terminate at a	can be a termination time of a
	happening time frame	4,871	occurs within	is time frame of
	occurrence period	4,872	occurs during	is occurrence period of
date				
	occurrence date	5,198	occurs at	is occurrence date of
point in time				
	end point in time	1,281	terminates at	is termination time of
	start point in time	1,384	begins at	is starting time of
	quantified time	1,783		
	conceptual start point	1,808	can start at a	can be a starting time of a
	conceptual end point	1,809	can terminate at a	can be a termination time of a
	conceptually quantified time			
property				
	possessed displacement property	1,774		
	quantified property	1,825		
	conceptually possessed property	1,810	can have as property a	can be a property of a
spatial point				
	preceding point	1,744	is an adjacent position after	is an adjacent position before

Specialization hierarchy (players of roles)	Roles (played in relations)	Identifier of relation type	Gellish relations	Inverse Gellish relations
	succeeding point	1,744	is an adjacent position after	is an adjacent position before
	displaced	1,772	is positioned relative to	is reference for position of
	reference point	1,772	is positioned relative to	is reference for position of
	conceptually displaced	1,889	can be positioned relative to a	can be a reference for the position of a
	conceptual reference point	1,889	can be positioned relative to a	can be a reference for the position of a
	common adjoint before	1,891	is by definition adjacent after	is by definition adjacent before
	common adjoint after	1,891	is by definition adjacent after	is by definition adjacent before
	conceptually located	1,801	can be a coordinate in a	can have as coordinate a
	terminator spatial point	1,514		
vector property				
vector property in 2D				
distance				
pitch				
relative property				
ratio				
scale in 2D				
property range				
1D space				
curve				
centreline				
	terminated spatial curve	1,514		
inner boundary				
	possessed inner boundary	2,002		
outer boundary				
	possessed outer boundary	2,003		
quality				
	conceptually possessed quality	2,050	can have as quality a	can be a quality of a
	conceptual condition	4,972	can be a condition in a	can be a condition evaluator of a
plurality				
impression				
appearance				
text appearance				
structure				
	conceptually possessed structure	4,795	can have as structure a	can be a structure of a
decomposition structure of an occurrence				
	conceptual whole occurrence structure	1,259		
	whole occurrence structure	1,258		
	possessed decomposition structure of an occurrence	1,788	has occurrence structure	is occurrence structure of
decomposition structure				
	commonly possessed decomposition structure	1,412		
	conceptual whole decomposition structure of a physical object	1,435		
	whole decomposition structure of a physical object	1,438		
	possessed decomposition structure	1,767	has as decomposition structure	is decomposition structure of
crystalline structure				
style				
information presentation style				
substance				
phase				
arrangement				
pattern				
tiling pattern				
line pattern				
configuration				
topology				
topologic sequence structure				
	commonly possessed topologic sequence structure	1,413	can have as arrangement a	can arrange a
	conceptual whole topologic sequence	1,816		
	whole topologic sequence	1,441		

Specialization hierarchy (players of roles)	Roles (played in relations)	Identifier of relation type	Gellish relations	Inverse Gellish relations
	possessed topologic sequence structure	1,789	has topological structure	is topologic structure of
encoding aspect				
	classifier for an encoding aspect	1,735		
	conceptual describer	1,806	can be described by a	can be a description of a
	commonly derived encoding	1,914	can be transcribed into	can be a transcription of
	common source of encoding	1,914	can be transcribed into	can be a transcription of
	expressor of information	1,293	is expressed by	is representer of
	classified encoding aspect	1,735		
	possessed encoding	1,763	has as pattern	is pattern of
	derived encoding	1,764	is transcribed into	is transcription of
	source encoding	1,764	is transcribed into	is transcription of
	common possessor of expression			
	conceptual expressor as encoding aspect	1,913	can be expressed by a	can be interpretated as a
	alias term	1,980		
	base term	1,980		
	partial class reference	2,019		
	describer	4,682	is described by	is a description of
	definer	4,685	is defined by	is a definition of
binary encoding aspect				
	binary representator			
ASCII encoding aspect				
textual encoding aspect				
	binary representator of text			
	partial class reference	2,019		
textual aspect				
information				
	classifier for information	1,285		
	conceptually referring	1,370	can be referenced within a	can include a reference to a
	conceptual provision	1,910	can be described via a	can be information about a
	common informer	1,911	with as information a	can be an information about
	includer	1,912	can be included in a	can include
	informer	1,273	is information about	is described via information
	classified information	1,285		
	referencer within information	1,369	is referred in	includes a reference to
	expressed information	1,760	is interpreted as	is an interpretation of
	commonly expressed information	1,798	can be symbolized by a	can be a symbolic representation of a
	compliancy criterion	4,753	shall individually be compliant with	is a compliancy criterion for
	adopted	4,813		
	conceptually required compliancy criterion	4,751	shall be a specification for a	shall be compliant with a
	presented information	4,996	is presented on	is presenter of
	conceptually presented information	4,999	can be presented on a	can be a presenter of a
reference aspect				
	classified reference aspect	1,761		
	referencer	1,770	is a reference for	is referenced by
	classifier for a reference aspect	1,761		
	conceptual referrer	1,917	can be a referencer for a	can have as reference a
	conceptual referencer	1,919	can be an indicator of a	can be indicated by a
	conceptually possessed reference aspect	1,799	can have as reference aspect a	can be a reference aspect of a
	partial class reference	2,019		
	possessed reference aspect	1,768	has as reference aspect	is a reference aspect of
reference aspect of class				
	commonly possessed reference aspect	2,018		
	whole class reference	2,019		

10.2 Appendix B, Upper ontology of relations with their roles and role players

This appendix presents a second view on the upper ontology part of the Gellish language definition in English. The view is from the perspective of the hierarchy of kinds of relations. Therefore it start with the concept ‘relation’ that already appears as subtype of ‘anything’ in the hierarchy of Appendix A. Therefore from a specialization/generalization hierarchy perspective Appendix B is just a branch of Appendix A.
This view is also generated by an automated procedure from the same Gellish Table with the formal definition as Appendix A.

This view presents:

- The specialization/generalization (subtype/supertype) hierarchy of kinds of relations in the upper ontology part of the Gellish language definition.
- The relations between those kinds of relations and the (two) kinds of roles that those kinds of relations require.
    - Note that n-ary kinds of relations are defined by a collection elementary binary kinds of relations. So, the n-ary kinds of relations appear in the hierarchy without roles, but their elementary kinds of relations appear in the hierarchy with their required roles. Those elementary kinds of relations express the kinds of roles that are required to be played in the n-ary kind of relations.
- The relations between those kinds of roles and the concepts that play (or whose members play) roles of those kinds.

The Gellish language is based on the idea that things exist only in relationship with other things. In line with that it is assumed that all facts and occurrences can be expressed by relations of various kinds. This ‘world full of relations’ (see Stafleu 2003) implies that the Gellish ontology includes an ontology of those kinds of relations. Most of the basic relations (or semantic primitive kinds of relations) are already presented as related to the concepts in the hierarchy of Appendix A.

Each kind of relation has a name that is direction independent and a Gellish English synonym phrase that is read from left to right and an inverse Gellish phrase that can be used to read the same expression from right to left.

It should be noted that the hierarchy also contains subtypes of kinds of relations that do not have Gellish synonyms. Those subtypes do not define Gellish grammar concepts. The reason why they are nevertheless presented as part of this view is that they facilitate mapping of legacy systems that include them (for example systems based on the data model of ISO 15926-2).

Specialization hierarchy of relations	Identifier of relation type	Gellish phrase	Inverse Gellish phrase	First role	Player of first role	Second role	Player of second role
relation	2,850	might be related to	might be related with	*relator*	anything	*related*	anything
denying relation	1,267						
probabilistic relation	1,268						
relation between individual things	4,658	is related to	is related with	*individual relator*	individual object	*individually related*	individual object
association	1,334	is associated with	is associated to				
exception of aspect from composition	1,147			*excepted*	aspect	*composed*	composition of an aspect
alternative association between objects	1,186	is alternative for	is preferred above	*alternative*		*alternative for*	
approval of object	1,187	is approver of	is approved by	*approver*	organization	*approved*	
change of state	1,199	is changed into	is changed from	*prestate*	state	*poststate*	state
composition of an individual	1,260	is a component of	is totality of	*part*	individual object	*whole*	individual object
composition of an occurrence	1,254			*partial occurrence*	occurrence	*whole occurrence*	occurrence
composition of a decomposition structure of an o	1,258			*part of an occurrence structure*	composition of an occurrence	*whole occurrence structure*	decomposition structure of an occurrence
composition of an aspect	1,262			*part of an aspect*	aspect	*whole aspect*	aspect
composition of temporal aspect	1,748			*part of temporal aspect*	temporal aspect	*whole temporal aspect*	temporal aspect
composition of spatial aspect	1,749			*part of spatial aspect*	spatial aspect	*whole spatial aspect*	spatial aspect
composition of a role	1,979			*part of a role*	role	*whole role*	role
composition of a physical object	1,436			*part of a physical object*	physical object	*whole physical object*	physical object
physical feature part of a physical object	1,492	is feature of	has feature	*feature part*	physical feature		
route through a physical object	1,787	is a route through	contains route	*contained route*	route	*container of route*	physical object
composition of decomposition structure of a phys	1,438			*part of decomposition structure of a physica*	composition of a physical object	*whole decomposition structure of a*	decomposition structure
composition of route	1,439			*part of route*	connection assembly	*whole route*	route
composition of topologic sequence structure of a	1,441			*part of topologic sequence*	topologic sequence of a physical ob	*whole topologic sequence*	topologic sequence structure
composition of spatial aspect point approximatio	1,743			*part of point approximation*	adjacency of spatial point	*whole point approximation*	point approximation of spatial aspect
composition of reference aspect of class	2,019			*partial class reference*	textual encoding aspect	*whole class reference*	reference aspect of class
organization of an individual	4,662						
assembly of an individual	1,190	is a part of	is whole of			*assembled whole*	individual object
amendment of an individual	4,742	is an amendment of	has as amendment	*amendment*		*amended*	
arrangement of an individual	4,661	is arranged in	is arrangement for			*arranged collection*	structured collection
network of relations	4,745						
derivation association between objects	1,269	is derived from	is original of	*derived*	anything	*source for derivation*	anything
derivation of annotation element	1,449			*derived annotation*	annotation element	*source annotation*	annotation element
hatching derivation for annotation element	1,580						
tiling derivation for annotation element	1,583						
view derivation for annotation element	1,586						
conversion of encoding aspect	1,764	is transcribed into	is transcription of	*derived encoding*	encoding aspect	*source encoding*	encoding aspect
realization of imaginary individual object	4,988	is a realization of	is realized by				
derivation of characteristic	1,270			*derived characteristic*	characteristic	*source characteristic list*	characteristic
information about object	1,273	is information about	is described via information	*informer*	information	*about*	anything
inclusion of information about object	5,046	includes information about	is described in				
fulfilment of an aspect	1,295	fulfills	is fulfilled by	*fulfilled aspect*	aspect	*fulfiller of required aspect*	aspect
installation of a physical object for imaginary physic	1,313	is installed for	is place of installation of	*installed*	physical object	*functional position*	imaginary physical object
purpose of existence	1,366	is existing for purpose	is the purpose for existence of	*possessor of purpose*	individual object	*purpose*	occurrence
reference to object within information	1,369	is referred in	includes a reference to	*referenced within information*	anything	*referencer within information*	information
endorsement	4,808	is endorsed by	is an endorsement of				
supplement	4,809	is supplemented by	is a supplement of				
succession association between objects	1,385	is a successor of	is a predecessor of	*successor*	anything	*predecessor*	anything
temporal sequence of an occurrence	1,388	occurs after	occurs before	*temporal successor*	occurrence	*temporal predecessor*	occurrence
version association between objects	1,393	is a version of	is a previous version of	*version*	anything	*original*	anything
revision association between objects	5,125	is a revision of	is a basis of revision for			*revised version*	
connection relation	1,487	is connected with	is connected to	*connecting assembly*	connection assembly	*connected*	physical object
logical connection of a physical object	1,445	is logically connected to	is logically connected with				
physical connection of a physical object	1,446	is physically connected to	is physically connected with				
destination of a route	1,490	is the destination of	has as destination	*destination*	physical object	*terminated route*	route
protection of a physical object	1,497	is protecting	is protected by	*protector*	physical object	*protected*	physical object
segregation of a physical object	1,504	is segregated from	is segregated from (inverse)	*segregated*	physical object	*segregated from*	physical object
source of a route	1,505	is the source of	has as source	*departure location*	physical object	*started route*	route
redundancy of a physical object	1,506			*redundant*	physical object	*duplicated*	physical object
usage of intermediate physical object for segregatio	1,509	is using for segregation	is used in segregation	*using segregation*	segregation of a physical object	*used for segregation*	physical object
usage of a physical object in a connection	1,510	is using as connection material	is connection material in	*user of connection material*	connection assembly	*connection material*	physical object
physical object following route	1,527	follows route	route is followed by	*follower of route*	physical object	*followed route*	route
custodianship	1,533	is custodian of	is guarded by	*custodian*	organization	*guarded*	anything
invisible annotation element in derived view	1,581			*invisible in view*	annotation element	*derived view*	
reference between physical objects	1,713	refers to	is referred from	*referring physical object*	physical object	*referred physical object*	physical object
display of annotation element on information carr	1,427	is displayed on	displays	*carried*	annotation element	*annotation carrier*	

Specialization hierarchy of relations	Identifier of relation type	Gellish phrase	Inverse Gellish phrase	First role	Player of first role	Second role	Player of second role
reference between intrapage connectors	1,513			*referring intrapage connector*	intrapage connector	*referred intrapage connector*	intrapage connector
carrying of information about physical object	4,720	contains information about	is described on	*information carrier*		*described in information carrier*	
spare relation between physical objects	4,804	is spare for	is spared by				
installed spare relation between physical obje	4,805	is installed spare for	is spared by installed				
comment on document	4,861	is comment on	has as comment				
address of a physical object	5,070	has address	is address of	*addressed*	physical object	*address*	area (surface)
visit address of a physical object	5,072	has visit address	is visit address of				
postal address of a physical object	5,073	has postal address	is postal address of				
e-mail address of a physical object	5,074	has e-mail address	is e-mail address of				
telephone address of a physical object	5,075	has telephone	is telephone of				
fax address of a physical object	5,076	has fax	is fax of				
being supplier of a physical object	5,150	is supplier of	is supplied by	*supplier*		*supplied*	
being maker of a physical object	5,160	is maker of	is made by	*maker*		*made*	
being manufacturer of a physical object	5,152	is manufacturer of	is manufactured by	*manufacturer*	organization	*manufactured*	
being made from raw material	5,268	is made from	is raw material for			*raw material*	
interaction between physical objects	5,368	interacts with	is interacted upon by				
being designer of a physical object	5,386	is designer of	is designed by				
reference to object by reference aspect	1,770	is a reference for	is referenced by	*referencer*	reference aspect	*referenced*	anything
role in life of a physical object	1,899						
cause of begin or end by occurrence	1,922	has as effect	is caused by	*causer*	occurrence	*caused*	temporal boundary of state
cause of begin of state	4,671	is the cause of begin	has as cause of begin				
cause of end of state	4,672	is the cause of end	is terminated by				
fulfilment of function by physical object	1,969	fulfils function	is a function of	*fulfiller of function*	individual object	*fulfilled function*	function
triggering of begin or end	2,026			*triggering*	temporal boundary of state	*triggered*	temporal boundary of state
representation	2,071	is represented by	represents	*represented*	anything	*representer*	individual object
representation of a physical object	1,515						
symbolization of object by annotation element	2,072	is symbolized by	symbolizes	*symbolized*		*symbolizer*	
representation of name of object by symbol	5,056	which name is represented by	represents the name of				
representation of value of a characteristic on sca	5,057	which value on scale is represente	represents the value on scale of				
location relative to a physical object	4,668	is location of	is located at	*reference location*	material	*located*	
relative location in a physical object	5,138	is located in	is location with in it				
containment of an individual	4,692	is contained by	is container of	*content*	physical object	*container*	physical object
correlation	4,679	is correlated by	is a correlated to	*correlated*	aspect	*correlated to*	
end point of period in time	1,281	terminates at	is termination time of	*ending period in time*	period in time	*end point in time*	point in time
start point of period in time	1,384	begins at	is starting time of	*starting period in time*	period in time	*start point in time*	point in time
approximation of mathematical space by points	1,754			*approximated*	mathematical space	*approximator*	point approximation of mathematical space
comparison of a characteristic with reference	1,756	is compared with	is comparison for	*compared*	characteristic	*referenced characteristic*	characteristic
displacement of spatial point	1,772	is positioned relative to	is reference for position of	*displaced*	spatial point	*reference point*	spatial point
expression of information by aspect	1,760	is interpreted as	is an interpretation of	*interpreted*	aspect	*expressed information*	information
expression of information by encoding aspect	1,293	is expressed by	is representer of	*expressor of information*	encoding aspect		
correlation between temporal aspect and state	1,762			*time of state change*	temporal aspect	*changing state*	state
correlation between start temporal boundary a	4,673	is the begin time of state	state begins at				
correlation between end temporal boundary a	4,674	is the end time of state	state terminates at				
correlation between time of change and state	5,278	is the time of change of state	state changed at				
quantification of an aspect by mathematical spac	2,044	is quantified as	is quantification of	*quantified aspect*	aspect	*quantifier for aspect*	mathematical space
scale	1,733			*quantified characteristic*	characteristic	*quantifying scale value*	linear space
geometric scale	1,779			*quantified spatial aspect*	spatial aspect	*quantifying scale value for spatial aspect*	
time scale	1,783			*quantified time*	point in time		
property scale	1,825			*quantified property*	property		
equality between magnitude of an aspect a	5,025	maps on scale to	is on scale the value of				
magnitude of an aspect greater than numb	5,026	has on scale a value greater than	is a scale value less than				
magnitude of an aspect less than number	5,027	has on scale a value less than	is a scale value greater than				
dimension of shape	4,664						
proportional correlation	4,735	is proportional with	is proportional to				
direct proportional correlation	4,737	is directly proportional with	is directly proportional to				
inverse proportional correlation	4,736	is inversely proportional with	is inversely proportional to				
bounding of an aspect	4,835	has as boundary	is boundary of	*bounded*	aspect	*boundary*	aspect
termination of a centreline	1,514			*terminated spatial curve*	centreline	*terminator spatial point*	spatial point
conditioning of an aspect	5,224	is conditioned by	is condition for	*conditioned*		*conditioning value*	
conditioning of equality of an aspect	5,271	has as equality condition	is equality condition of				
conditioning of inequality of an aspect	5,272	has as inequality condition	is inequality condition of				
conditioning on lower boundary	5,273	has as lower boundary condition	is lower boundary condition of				
conditioning on upper boundary	5,274	has as upper boundary condition	is upper boundary condition of				
conditioning on approximate value	5,275						

Specialization hierarchy of relations	Identifier of relation type	Gellish phrase	Inverse Gellish phrase	First role	Player of first role	Second role	Player of second role
conditioning on range	5,276						
physical correlation	552,199						
property point	551,005						
coordinates	552,764						
2D coordinates	5,376						
two dimensional cartesian coordinat	553,066						
3D coordinates	5,377						
three dimensional cartesian coordin	553,068						
location coordinates	552,765						
two dimensional cartesian coordinat	553,066						
three dimensional cartesian coordin	553,068						
point in a 4D space-time space	5,385						
mathematical function	920,188						
arithmetic function	910,786						
absolute value function	910,714						
modulus function	911,088						
arc cosine function	911,548						
arc sine function	911,550						
arc tangent function	911,552						
cosine function	911,560						
differentiation function	911,564						
integration function	911,576						
logarithm function	911,580						
exponentiation function	911,598						
sine function	911,606						
tangent function	911,614						
addition function	911,684						
division function	911,685						
multiplication function	911,687						
proportional function	924,647						
subtraction function	920,044						
averaging function	920,218						
count function	920,222						
linear function	920,229						
linear conversion between scales	1,867	can be converted in	can be a conversion of	*converted scale*	scale	*target scale*	scale
maximum function	920,231						
minimum function	920,233						
nonlinear function	920,234						
proportioning and integrating and differenti	920,237						
selection function	920,240						
wait function	920,242						
volume in a 4D space	5,372						
surface in a 3D space	5,373						
curve in a 2D space	5,374						
curve in a 3D space	5,375						
common exception of an aspect from composition	4,774			*commonly excepted*		*commonly composed*	composition of an aspect
control of reference by authority	4,775			*controlled*	reference to object by reference asp	*controlling authority of reference*	
context for unique reference	4,776	is unique in context of	is a uniqueness context for	*reference*	reference to object by reference asp	*uniqueness context for individual re*	anything
purpose of relation	4,889			*intended relation for purpose*	relation	*purpose*	occurrence
purpose of approval	1,364	is approved for	is the purpose of approval of	*intended approval for purpose*	approval of object		
purpose of protection	1,740						
purpose of segregation	1,741						
ownership	4,907	is owner of	is owned by	*owner*		*owned*	
conditional occurrence function	4,959						
presentation of information on physical object	4,996	is presented on	is presenter of	*presented information*	information	*presenter of information*	physical object
information about object in collection of information	5,047	each of which includes information	is described within information colle	*collective informer*	collection of information elements	*collectively about*	
authorship	5,126	is author of	is written by	*author*		*written*	
compliancy to a criterion	5,128	is satisfied criterion for	is compliant with	*criterion*		*complier*	
conceptual fulfilment of required fact	5,165	can fulfil a	can be fulfilled by a	*fulfilled requirement*		*fulfiller of requirement*	
association between parent and child	5,284	is a parent of	is a child of	*parent*	person	*child*	person
association between father and child	5,287	is a father of	has as father	*father*	man		
association between mother and child	5,289	is a mother of	has as mother	*mother*	woman		
association between parent and son	5,291		is a son of			*son*	man
association between parent and daughter	5,293		is a daughter of			*daughter*	woman

Specialization hierarchy of relations	Identifier of relation type	Gellish phrase	Inverse Gellish phrase	First role	Player of first role	Second role	Player of second role
marriage	990,125	is married with	is married by	*husband*	man	*wife*	woman
possession of an aspect by an individual thing	1,727	has aspect	is an aspect of	*possessor of an aspect*	individual object	*possessed aspect*	aspect
possession of recognition aspect	1,729	is recognised by	is a recognition aspect of			*possessed recognition aspect*	
possession of encoding aspect by an individual	1,763	has as pattern	is pattern of			*possessed encoding*	encoding aspect
possession of reference aspect by an individual	1,768	has as reference aspect	is a reference aspect of	*possessor of reference aspect*	individual object	*possessed reference aspect*	reference aspect
possession of result of comparison between charac	1,771	has comparison of	is comparison of	*possessor of comparison result*	comparison of a characteristic with	*possessed comparison*	characteristic
possession of property by displacement	1,774			*possessor of displacement property*	displacement of spatial point	*possessed displacement property*	property
possession of point approximation by spatial aspect	1,780	has point approximation	is approximate point of	*possessor of point approximation of spatial*	spatial aspect	*possessed point approximation of s*	point approximation of spatial aspect
possession of spatial aspect by begin or end	1,786	begins or ends at location	is location where	*located in space*	temporal boundary of state	*place of begin or end*	spatial aspect
possession of spatial aspect by begin	5,088	is begin at location	is location of begin				
possession of spatial aspect by end	5,090	is end at location	is location of end				
possession of decomposition structure by occurrenc	1,788	has occurrence structure	is occurrence structure of	*possessor of decomposition structure of an*	occurrence	*possessed decomposition structure*	decomposition structure of an occurrence
possession of topologic sequence structure of a phy	1,789	has topological structure	is topologic structure of	*possessor of topologic sequence structure*	physical object	*possessed topologic sequence stru*	topologic sequence structure
possession of inner boundary	2,002			*possessor of inner boundary*		*possessed inner boundary*	inner boundary
possession of outer boundary	2,003			*possessor of outer boundary*		*possessed outer boundary*	outer boundary
possession of spatial aspect	2,073						
possession of a role	4,708						
requirement of function by occurrence	1,991	requires function	is played in	*requirer of function*	occurrence	*required function*	function
role of an individual	4,713						
possession of a characteristic	4,797						
possession of property	4,798						
possession of quality	4,799						
possession of structure	4,793	has as structure	is the structure of				
possession of decomposition structure by	1,767	has as decomposition structure	is decomposition structure of	*possessor of decomposition structure*	physical object	*possessed decomposition structure*	decomposition structure
possession of atomic structure	4,794	is of substance	is material of construction of				
possession of temporal aspect	4,800			*possessor of time aspect*	individual object	*possessed time aspect*	temporal aspect
possession of temporal aspect by begin or end	1,785	begins or ends at point in time	is point in time of	*located in time*	temporal boundary of state	*time of begin or end*	temporal aspect
possession of temporal aspect by begin	5,087	is begin at point in time	is point in time of start				
possession of temporal aspect by end	5,089	is end at point in time	is point in time of end				
occurrence within period	4,871	occurs within	is time frame of	*happened*	individual object	*happening time frame*	period in time
occurred within period	5,077	happened within	was time frame of				
required occurrence within period	5,079						
expected occurrence within period	5,081						
occurrence at date	5,198	occurs at	is occurrence date of	*occurred*		*occurrence date*	date
creation at date	4,862	is created at	is creation date of	*created*		*creation date*	
expiration at date	4,863	is expired at	is expiration date of	*expired*		*expiration date*	
termination at date	4,864	terminates at date	is termination date of	*terminated*		*termination date*	
completion at date	5,124	has as completed date	is completion date of	*completed*		*completion date*	
out of service at date	5,206	is out of service at	is out of service date of	*out of service*		*out-of-service date*	
end at date	5,208	ended at	is end date of	*ended*		*end date*	
revision at date	4,865	has as revision date	is revision date of	*revised*		*revision date*	
birth at date	4,866	is born at	is birth date of	*born*	person	*birth date*	
arrival at date	4,867	arrival at	is arrival date of	*arrived*		*arrival date*	
departure at date	4,869	departure at	is departure date of	*departed*		*departure date*	
release at date	4,870	release at	is release date of	*released*		*release date*	
issue at date	4,868	issue at	is issue date of	*issued*		*issue date*	
start at date	5,098	is started at	is start date of	*started*		*start date*	
in service at date	5,200	is in service at	is in service date of	*in service*		*in-service date*	
submission at date	5,139	submission at	is submission date of	*submitted*		*submission date*	
acceptance at date	5,199	is accepted at	is acceptance date of	*being accepted*		*acceptance date*	
testing at date	5,201	is tested at	is test date of	*tested*		*test date*	
delivery at date	5,202	is delivered at	is delivery date of	*delivered*		*delivery date*	
addition at date	5,203	is added at	is addition date of	*added*		*addition date*	
installation at date	5,204	is installed at	is installation date of	*installed at*		*installation date*	
modification at date	5,205	is modified at	is modification date of	*modified*		*modification date*	
purchase at date	5,207	is purchased at	is purchase date of	*purchased*		*purchase date*	
registration at date	5,209	is registered at	is registration date of	*registered*		*registration date*	
creation within period	5,393	is created within	is creation time frame of				
occurrence during period	4,872	occurs during	is occurrence period of	*occurring*	occurrence	*occurrence period*	period in time
occurred during period	5,078	happened during	was occurrence period of				
required occurrence during period	5,080						
expected occurrence during period	5,082						
possession of a role by an individual object	5,234	is a role of	is player of	*player of a role*	physical object	*played role*	role
aspect of a part of a physical object	5,248	has a part with aspect	is an aspect of a part of	*possessor of an aspect of a part*	physical object	*possessed by part*	aspect

Specialization hierarchy of relations	Identifier of relation type	Gellish phrase	Inverse Gellish phrase	First role	Player of first role	Second role	Player of second role
role of a part of a physical object	1,300	is a function for part of	has a part with role				
possible role for individual thing	4,709	is a possible role for	has as possible role			*possible role*	role
intended role for individual thing	4,715	has as intended role	is the intended role for			*intended role*	role
foundation relation	4,738	is based on	is basis for				
involvement in relation	4,961						
involvement in an occurrence	4,767	is involved in	involves	*function*	individual object	*involver*	occurrence
involvement as subject	4,760	is subject in	has as subject	*subject*			
involvement as performer	4,761	is performer in	is performed by	*performer*	physical object		
involvement as provider	4,916	is provider of	is provided by	*provider*	lifeform		
involvement as contractor	4,911	is contractor of	is contracted by	*contractor*			
involvement as engineering contractor	4,910	is engineering contractor of	is engineered by contractor	*engineering contractor*			
involvement as controller	4,762	is controller in	is controlled by	*controller*			
involvement as managing organization	4,908	is managing organization of	is managed by organization	*managing organization*	organization		
involvement as managing contractor	4,909	is managing contractor of	is managed by managing contracto	*managing contractor*			
involvement as manager	4,912	is manager of	is managed by	*manager*	person		
involvement as aid	4,764	is an aid in	uses as aid	*aid*			
involvement as tool	4,763	is tool in	uses as tool	*tool*			
involvement as input	4,785	is input in	has as input				
involvement as output	4,786	is output of	has as output				
involvement as customer	4,915	is customer in	is required by customer	*customer*	organization		
involvement as committer	5,037	is committer of	is committed by	*committer*	person	*committed*	act
involvement as place of occurrence	5,083	is place of occurrence of	has as place of occurrence	*place of occurrence*			
involvement as planner	5,142	is planner of	is planned by	*planner*			
involvement as scheduler	5,144	is scheduler of	is scheduled by	*scheduler*			
involvement as enabler	5,169	is enabler of	uses as enabler	*enabler*			
subjection to correlation	4,886	has as correlation	is correlation for	*subject of correlation*	physical object	*governing correlation*	correlation
involvement in correlation	4,962	is a parameter in	has as parameter	*parameter*	aspect		
coordinate in coordinates point	1,777	is a coordinate of	has as coordinate	*coordinate*	aspect	*involver of coordinate*	property point
coordinates in a coordinate system	1,876	is a coordinates point in	is the coordinate system for	*coordinates point*	property point	*coordinate system*	property space
condition for occurrence	4,960	is condition in	is evaluating the condition	*condition*	conditioning of an aspect	*conditioned function*	conditional occurrence function
action on satisfied condition	4,966	is required action in	is triggering function of	*required action*	occurrence	*triggering function*	conditional occurrence function
action on unsatisfied condition	4,969	is alternative action in	is alternative action triggering funct	*alternative action*	occurrence	*alternative action triggering functior*	conditional occurrence function
coordinates point of individual thing	5,378	is a coordinates point of	has as coordinates point	*locating coordinates point*		*located at coordinates point*	
experience of an aspect by an individual thing	5,058	experiences aspect	is aspect experienced by	*experiencer*		*experienced aspect*	
objective of activity	5,115	is objective of	is intended to achieve	*objective*	state	*means*	occurrence
sequency relation	5,332	is the next element after	is the previous element before	*succeeding element*		*preceding element*	
topologic sequence of a physical object	1,508	is located after	is located before	*topologic successor*	physical object	*topologic predecessor*	physical object
adjacency of spatial point	1,744	is an adjacent position after	is an adjacent position before	*succeeding point*	spatial point	*preceding point*	spatial point
adjacency of mathematical point	1,751	is an adjacent point after	is an adjacent point before	*succeeding adjoint mathematical point*	mathematical point	*preceding adjoint mathematical poi*	mathematical point
last sequency relation	5,338	is the last element after	is the one but last element before				
state	790,123						
occurrence	193,671						
act	193,277						
description of object	4,682	is described by	is a description of	*described*	anything	*describer*	encoding aspect
definition of object	4,685	is defined by	is a definition of	*defined*	anything	*definer*	encoding aspect
relation between classes	4,718						
specialization of class	1,146	is a specialization of	is a generalization of	*subtype*	concept	*supertype*	concept
qualification of concept	1,726	is a qualification of	is the nature of	*qualifier*	concept	*nature*	concept
qualification of mathematical space	1,823			*qualified mathematical space*	mathematical space	*nature of mathematical space*	mathematical space
qualification of physical object	5,396	is a model of	is the nature of model	*model*			
partial specialization of class	5,277	is a partial subtype of	is a composed supertype of				
classification of a class	1,224	is a particular	is a kind of class of	*classified class*	concept	*classifier for a class*	class
common alias for encoded information	1,980			*alias term*	encoding aspect	*base term*	encoding aspect
common synonym for encoded information	1,981	is a synonym of	is a synonym for				
common abbreviation for encoded information	1,982	is an abbreviation of	is abbreviated as				
common code for encoded information	1,983	is a code for	is coded as				
common page code for encoded information	5,053	is a page number for	has as page number				
common sheet code for encoded information	5,054	is a sheet number for	has as sheet number				
common revision code for encoded information	5,055	is a revision code for	has as revision code				
common stream number for encoded informatior	5,064	is a stream number for	has as stream number				
common location number for encoded informatio	5,065	is a location number for	has as location number				
common noun form of verb	1,984	is a noun form of	is a verb form of				
common passive form of verb	1,985	is a passive form of	is an active form of				
common inverse of encoding aspect	1,986	is an inverse of	is inverted from				

Specialization hierarchy of relations	Identifier of relation type	Gellish phrase	Inverse Gellish phrase	First role	Player of first role	Second role	Player of second role
common translation of encoding aspect	4,691	is a translation of	is a translation for				
common name for encoded information	5,117	is a name of	is called				
common long name for encoded information	5,122	is a long name of	has as long name				
common serial number for encoded information	5,123	is a serial number of	has as serial number				
common title for encoded information	5,350	is a title of	has as title				
common subtitle for encoded information	5,351	is a subtitle of	has as subtitle				
common identifier for encoded information	5,352	is an identifier of	has as identifier				
conceptual relation between members of classes	4,698			*conceptual relator*		*conceptually related*	
conceptual association	1,231						
common assertion	1,230						
usage of member of a subtype of physical object	1,664	can be used in connection	is a connection using a	*used member*	physical object	*using member*	conceptual connection with a physical objec
usage of member of a subtype of physical fea	1,710			*used member feature*	physical feature		
conceptual usage of a physical object in a conne	1,724	can use as connection material a	can be connection material in a	*conceptual user of connection material*	connection assembly	*conceptually used*	physical object
conceptual requirement for usage of a physica	5,406						
common correlation	2,067						
common quantification of a characteristic on s	1,791			*commonly quantified characteristic*	characteristic	*common quantifier of a characterist*	mathematical space
common equality between magnitude of ar	5,279	can have as scale value	can be a scale value for a				
common magnitude of an aspect greater th	5,280	can have on scale a value greater t	can be a scale value less than a				
common magnitude of an aspect less than	5,281	can have on scale a value less tha	can be a scale value greater than a				
common bounding of an aspect	4,838	has as common boundary	is a common boundary of a	*commonly bounded*	aspect	*common boundary*	aspect
bounding of mathematical space	1,773	has as mathematical boundary	is mathematical boundary of	*bounded mathematical space*	mathematical space	*boundary of mathematical space*	mathematical space
common termination by lower boundary	4,917	has as lower boundary	is lower boundary of				
common termination by upper boundary	4,918	has as upper boundary	is upper boundary of				
conceptual carrying of information about class	4,726	can contain information about a	can be described in a	*conceptual information carrier*	physical object	*conceptually described in an inform*	individual object
required carrying of information about class	5,102	shall contain information about a	shall be described in a				
common context for applicability of fact	4,822						
common context for unique reference	1,920	shall be unique in the context of	is a uniqueness context for a	*common reference*	conceptual reference to object by re	*uniqueness context for common ref*	anything
conceptual relation	1,554						
conceptual association	1,231						
common assertion	1,230						
usage of member of a subtype of physical obj	1,664	can be used in connection	is a connection using a	*used member*	physical object	*using member*	conceptual connection with a physical objec
usage of member of a subtype of physical	1,710			*used member feature*	physical feature		
conceptual usage of a physical object in a cor	1,724	can use as connection material a	can be connection material in a	*conceptual user of connection material*	connection assembly	*conceptually used*	physical object
conceptual requirement for usage of a phy	5,406						
common correlation	2,067						
common quantification of a characteristic o	1,791			*commonly quantified characteristic*	characteristic	*common quantifier of a characterist*	mathematical space
common equality between magnitude o	5,279	can have as scale value	can be a scale value for a				
common magnitude of an aspect greate	5,280	can have on scale a value greater t	can be a scale value less than a				
common magnitude of an aspect less th	5,281	can have on scale a value less tha	can be a scale value greater than a				
common bounding of an aspect	4,838	has as common boundary	is a common boundary of a	*commonly bounded*	aspect	*common boundary*	aspect
bounding of mathematical space	1,773	has as mathematical boundary	is mathematical boundary of	*bounded mathematical space*	mathematical space	*boundary of mathematical space*	mathematical space
common termination by lower boundary	4,917	has as lower boundary	is lower boundary of				
common termination by upper boundar	4,918	has as upper boundary	is upper boundary of				
conceptual carrying of information about class	4,726	can contain information about a	can be described in a	*conceptual information carrier*	physical object	*conceptually described in an inform*	individual object
required carrying of information about clas	5,102	shall contain information about a	shall be described in a				
common context for applicability of fact	4,822						
common context for unique reference	1,920	shall be unique in the context of	is a uniqueness context for a	*common reference*	conceptual reference to object by re	*uniqueness context for common ref*	anything
conceptual denying relation	1,236						
conceptual probabilistic relation	1,243						
conceptual composition of an individual	1,261	can be a component of a	can be a whole for a	*conceptual part*	individual object	*conceptual whole*	individual object
conceptual assembly of an individual	1,191	can be a part of a	can have as part a			*conceptual assembled whole*	individual object
conceptual requirement for assembly of ar	4,989	shall be a part of a	shall have as part a				
conceptual presence of assembly of an inc	5,003	whether being part of a	shall indicate the presence of a				
conceptual collection of an individual thing	1,228	can be an element of a	can be a plural whole for a			*conceptual collected whole*	individual object
conceptual requirement for collection of ar	5,174	shall be an element of a	shall be a plural whole for a				
conceptual composition of an occurrence	1,255			*conceptual part of an occurrence*	occurrence	*conceptual whole occurrence*	occurrence
conceptual composition of an aspect	1,257			*conceptual part aspect*	aspect	*conceptual whole aspect*	aspect
composition of mathematical space	1,746			*part of mathematical space*	mathematical space	*whole mathematical space*	mathematical space
composition of mathematical space point a	1,750			*part of point approximation of mathematica*	adjacency of mathematical point	*whole point approximation of mathe*	point approximation of mathematical space
conceptual composition of mathematical s	1,817			*conceptual part mathematical space*	mathematical space	*conceptual whole mathematical spa*	mathematical space
conceptual composition of a role	1,978			*conceptual part role*	role	*conceptual whole role*	role
conceptual composition of a decomposition st	1,259			*conceptual part of an occurrence structure*	composition of an occurrence	*conceptual whole occurrence struct*	decomposition structure of an occurrence
conceptual composition of route	1,405			*conceptual part of route*	connection assembly	*conceptual whole route*	route
conceptual composition of decomposition stru	1,435			*conceptual part of decomposition structure*	composition of a physical object	*conceptual whole decomposition st*	decomposition structure

Specialization hierarchy of relations	Identifier of relation type	Gellish phrase	Inverse Gellish phrase	First role	Player of first role	Second role	Player of second role
conceptual composition of a physical object	1,437			*conceptual part of a physical object*	physical object	*conceptual whole physical object*	physical object
conceptual physical feature part of a physi	1,410	can be a feature of a	can have a feature like	*conceptual feature part*	physical feature		
conceptual route through a physical object	1,411	can contain as route a	can be a route in a	*conceptual container of route*	physical object	*conceptually contained route*	route
conceptual composition of topologic sequenc	1,816			*conceptual part of topologic sequence*	topologic sequence of a physical ob	*conceptual whole topologic sequen*	topologic sequence structure
conceptual composition of point approximatio	1,818			*conceptual part of point approximation of m*	adjacency of mathematical point	*conceptual whole point approximati*	point approximation of mathematical space
conceptual composition of spatial aspect poin	1,819			*conceptual part point approximation*	adjacency of spatial point	*conceptual whole point approximati*	point approximation of spatial aspect
conceptual requirement for composition of an	4,901	shall be a component of a	shall be a whole for a				
conceptual reference to object within information	1,370	can be referenced within a	can include a reference to a	*conceptually referring*	information	*conceptually referenced within infor*	individual object
conceptual requirement for reference to objec	5,371	shall be referenced within a	shall include a reference to a				
conceptual temporal sequence of an occurrence	1,389	can occur after a	can occur before a	*conceptual temporal successor*	occurrence	*conceptual temporal predecessor*	occurrence
conceptual requirement for temporal sequenc	5,118	shall occur after a	shall occur before a				
conceptual connection relation	1,407	can be connected to a	can have a connection with a	*conceptually connected from*	physical object	*conceptually connected to*	physical object
conceptual logical connection of a physical ob	1,447	can be logically connected to a	can be logically connected with a				
conceptual physical connection of a physical	1,448	can be physically connected to a	can be physically connected with a				
required connection relation	5,110	shall be connected to a	shall have a connection with a				
conceptual destination of a route	1,409	can be a destination of a	can end at a	*conceptual destination*	physical object	*conceptually terminated route*	route
conceptual requirement for a destination of a	5,409	shall be a destination of a	shall have as destination a				
conceptual protection of a physical object	1,415	can be protected by a	can protect a	*conceptually protected*	physical object	*conceptual protector*	physical object
conceptual segregation of a physical object	1,418	can be segregated from a	can be segregated (inverse) from a	*conceptually segregated*	physical object	*conceptually segregated from*	physical object
conceptual source of a route	1,419	can be a source of a	can have as source a	*conceptual source*	physical object	*conceptually started route*	route
conceptual requirement for a source of a rout	5,408	shall be a source of a	shall have as source a				
conceptual usage of intermediate physical object	1,422	can be used in segregation	can use in segregation a	*commonly used for segregation*	physical object	*commonly using for segregation*	conceptual segregation of a physical object
conceptual physical object following route	1,529	can follow a	can be followed by a	*conceptual follower of route*	physical object	*conceptually followed route*	route
conceptual change of state	1,550	can be a prestate of a	can be a poststate of a	*conceptual prestate*		*conceptual poststate*	
conceptual purpose of existence	1,609	can have as purpose a	can be a purpose of a	*conceptual possessor of purpose*	individual object	*conceptual purpose*	occurrence
conceptual requirement for a purpose of exist	5,412	shall have as purpose a	shall be a purpose of a				
conceptual reference between physical objects	1,715	can refer to a	can be referred from a	*conceptually referring physical object*	physical object	*conceptually referred physical obje*	physical object
conceptual requirement for reference betwee	5,120	shall refer to a	shall be referred from a				
conceptually being made from raw material	5,266	can be made from a	can be raw material for making a			*conceptually raw material*	
conceptual interaction between physical obje	5,369	can interact with a	can be interacted upon by a				
conceptual requirement for interaction bet	5,404	shall interact with a	shall be interacted upon by a				
conceptual requirement for interaction with	5,405	shall interact with something that is	shall have a subtype of which a member is interacted upon by a				
conceptual address of a physical object	5,414	can be an address of a	can have as address a				
conceptual requirement for an address of a	5,415	shall be an address of a	shall have as address a				
conceptual comparison of a characteristic	1,820						
conceptual displacement of spatial point	1,889	can be positioned relative to a	can be a reference for the position	*conceptually displaced*	spatial point	*conceptual reference point*	spatial point
conceptual installation of a physical object for an	1,903	can be installed at the position of a	can be a position for installation of	*conceptual installation position*	imaginary physical object	*conceptually installed*	physical object
conceptual requirement for installation of a ph	5,407	shall indicate to be installed at the	shall indicate to be a position of installation of a				
conceptual role of a part of an individual physica	1,906	can be a role of a part of	can be the whole for a part with as	*common role of a part of an individual*	role	*whole physical object with common*	physical object
conceptual physical feature of an individual phys	1,908			*common feature part of an individual*	physical feature	*individual whole for kind of physical*	physical object
conceptual information about members of class	1,910	can be described via a	can be information about a	*conceptually described*	individual object	*conceptual provision*	information
required information about members of class	5,103	requires as information a	is a kind of information required by a				
conceptual information about object	1,911	with as information a	can be an information about	*commonly about*	anything	*common informer*	information
conceptual inclusion of information	1,912	can be included in a	can include	*included*	anything	*includer*	information
conceptual conversion of encoding aspect	1,914	can be transcribed into	can be a transcription of	*common source of encoding*	encoding aspect	*commonly derived encoding*	encoding aspect
conceptual reference by reference aspect	1,917	can be a referencer for a	can have as reference a	*conceptual referrer*	reference aspect	*conceptually referred*	individual object
conceptual requirement for reference by refer	5,116	shall be referenced by a	shall be a reference of a				
conceptual reference to object by reference aspe	1,919	can be an indicator of a	can be indicated by a	*conceptual referencer*	reference aspect	*conceptually referenced*	anything
conceptual mathematical function	2,038						
conceptual factorization relation	4,825	can be an operand in a	can have as operand a	*operand*	individual object	*factorized*	correlation
conceptual denominator relation	4,826	can be a denominator in a	can have as denominator a				
conceptual exponent relation	4,827	can be an exponent in a	can have as exponent a				
conceptual multiplied relation	4,828	can be multiplied in a	can have as multiplied a				
conceptual multiplier relation	4,829	can be a multiplier in a	can have as multiplier a				
conceptual numerator relation	4,830	can be a numerator in a	can have as numerator a				
conceptual base for exponentiation relatio	4,831	can be exponentiated in a	can have as exponentiation basis a				
conceptual product relation	4,833	can be a product in a	can have as product a				
conceptual quotient relation	4,834	can be a quotient in a	can have as quotient a				
conceptual correlation	2,066	can be correlated to a	can be correlated with a				
conceptual derivation of characteristic	1,271			*conceptual source list*	characteristic	*conceptually derived*	characteristic
conceptual correlation between temporal asp	1,551			*conceptual possessor of time aspect*	state	*conceptually possessed time aspec*	temporal aspect
conceptual scale	1,797	can be quantified by a	can be a quantifier of a				
conceptual geometric scale	1,860						
conceptual coordinates in coordinate sy	1,875						

Specialization hierarchy of relations	Identifier of relation type	Gellish phrase	Inverse Gellish phrase	First role	Player of first role	Second role	Player of second role
conceptual time scale	1,861						
conceptual expression of information by aspe	1,798	can be symbolized by a	can be a symbolic representation o	*conceptual expressor*	aspect	*commonly expressed information*	information
conceptual expression of information by er	1,913	can be expressed by a	can be interpretated as a	*conceptual expressor as encoding aspect*	encoding aspect		
conceptual requirement for expression	5,168	shall be expressed by a	shall be an expression of a				
conceptual approximation of mathematical sp	1,805	mathematical space can have as a	can be a point approximate of a ma	*conceptually approximated*	mathematical space	*conceptual approximator*	point approximation of mathematical space
conceptual start point of period in time	1,808	can start at a	can be a starting time of a	*conceptual starting period in time*	period in time	*conceptual start point*	point in time
conceptual end point of period in time	1,809	can terminate at a	can be a termination time of a	*conceptual ending period in time*	period in time	*conceptual end point*	point in time
conceptual comparison of a characteristic witl	1,851	can be compared to a	can be a reference for comparison	*conceptually compared characteristic*	characteristic	*conceptually referenced characteris*	characteristic
conceptual bounding of an aspect	5,032	can have as boundary a	can be a boundary of a	*conceptually bounded*		*conceptual boundary*	
conceptual bounding of mathematical spac	1,804	can have as mathematical constrai	can be a mathematical constraint o	*conceptually bounded mathematical space*	mathematical space	*conceptual boundary of mathematic*	mathematical space
conceptual conditioning of an aspect	5,227	can have as condition a	can be a condition for a				
conceptual representation of correlation by funct	2,068			*represented conceptual correlation*	conceptual correlation	*representing function*	mathematical function
conceptual display of annotation element on info	4,810	can be a displayer of a	can be displayed on a	*conceptual carrier*	physical object	*conceptually carried*	physical object
conceptual representation	4,924	can be represented by a	can represent a	*conceptually represented*	individual object	*conceptually representing*	individual object
conceptual requirement for a representation	5,410	shall be represented by a	shall represent a				
conceptual presentation of information on physic	4,999	can be presented on a	can be a presenter of a	*conceptually presented information*	information	*conceptual presenter of information*	physical object
conceptual requirement for presentation of inf	5,121						
conceptual custodianship	5,004	can be a custodian of a	can be guarded by a	*conceptual custodian*	organization	*conceptually guarded*	anything
conceptual requirement for a custodianship	5,012	shall be custodian of a	shall have as custodian a				
conceptual ownership	5,008	can be an owner of a	can be owned by a	*conceptual owner*		*conceptually owned*	
conceptual requirement for ownership	5,011	shall be the owner of a	shall be owned by a				
conceptual realization of fact	5,091	can be a realization of a	can be realized by a				
conceptual begin of an occurrence	5,104	can begin with a	can be a beginner of a	*conceptually started*		*conceptual starter*	
conceptual termination of an occurrence	5,107	can terminate with a	can be a terminator of a	*conceptually terminated*		*conceptual terminator*	
conceptual sequency relation	5,335	can be a next element after a	can be a previous element before a	*conceptual succeeding element*		*conceptual preceding element*	
conceptual topologic sequence of a physical (	1,421	can be located after a	can be located before a	*conceptual topologic successor*	physical object	*conceptual topologic predecessor*	physical object
conceptual requirement for topologic sequ	5,119	shall be located after a	shall be located before a				
common adjacency of mathematical point	1,850			*common adjoint mathematical point 1*	mathematical point	*common adjoint mathematical point*	mathematical point
common adjacency of spatial point	1,891	is by definition adjacent after	is by definition adjacent before	*common adjoint before*	spatial point	*common adjoint after*	spatial point
conceptual description by an encoding aspect	1,806	can be described by a	can be a description of a			*conceptual describer*	encoding aspect
conceptual requirement for a description by an e	5,413	shall be described by a	shall be a description of a				
conceptual possession of an aspect	2,069	can have a	can be an aspect of a	*conceptual possessor*	anything	*conceptually possessed*	aspect
conceptual possession of a decomposition struct	1,412			*common possessor of decomposition struc*	physical object	*commonly possessed decompositic*	decomposition structure
conceptual possession of a topologic sequence s	1,413	can have as arrangement a	can arrange a	*common possessor of topologic sequence*	physical object	*commonly possessed topologic sec*	topologic sequence structure
conceptual possession of a spatial aspect	1,711	can have as shape a	can be a shape of a			*conceptually possessed spatial asp*	spatial aspect
conceptual possession of recognition aspect	1,759	can be recognised by a	can be a recognition aspect for a				
conceptual possession of reference aspect by ar	1,799	can have as reference aspect a	can be a reference aspect of a	*conceptual possessor of reference aspect*	individual object	*conceptually possessed reference a*	reference aspect
conceptual possession of point approximation by	1,802	can have as point approximation a	can be a point approximate of a	*common possessor of point approximation*	spatial aspect	*commonly possessed point approxi*	point approximation of spatial aspect
conceptual possession of result of comparison b	1,813	can be a comparison result of a	can have as comparison result a	*conceptual possessor of comparison result*	comparison of a characteristic with reference		
conceptual possession of reference aspect by cl	2,018			*common possessor of reference aspect*	concept	*commonly possessed reference as*	reference aspect of class
conceptual quantification by mathematical space	2,047	can be quantified by number of iten	can be a quantification of a	*conceptually quantified*	individual object	*conceptual quantifier*	mathematical space
common possession of an aspect	2,070			*common possessor*	concept	*commonly possessed*	aspect
common possession of an aspect by definitio	4,816	is by definition	is a qualifying aspect of a				
common possession of atomic structure	5,270	is by definition made of	is material of construction of a				
common qualification of an aspect in role	5,283	is by definition qualified as	is by definition a qualification of a				
conceptual role for members of a subtype of indi	4,714	can have a role as a	can be a role for a	*conceptual player of a role*	concept	*conceptually played role*	role
conceptual fulfilment of function by physical o	1,314	can fulfil a function as a	can be a function of a	*conceptual fulfiller of function*	individual object	*conceptually fulfilled function*	function
conceptual possible role for members of a sul	2,075						
conceptual intended role for members of a su	4,717	is intended to play a role as a	is an intended kind of role for a				
conceptual required role for members of a sul	4,732	requires a role as a	is a required role of a	*conceptual requirer of a role in relation*			
conceptual first required role for members	4,731	requires a role-1 as a	is a first required role of a				
conceptual second required role for memb	4,733	requires a role-2 as a	is a second required role of a				
conceptual definition of a role of an aspect	5,343	is defined to have a possible role a	is defined as a possible role of a				
conceptual possession of a characteristic	4,792					*conceptually possessed characteristic*	
conceptual possession of temporal aspect	1,807	can have as time aspect a	can be a point in time of a			*conceptually possessed time aspec*	temporal aspect
conceptual occurrence within period	4,927						
conceptual creation at date	4,928	can be created at a	can be a creation date of a	*conceptually created*		*creation date*	
conceptual revision at date	4,931	can be revised at a	can be a revision date of a	*conceptually revised*		*revision date*	
conceptual occurrence at date	5,339	can occur at a	can be an occurrence date of a	*conceptually occurred*	anything	*occurrence date*	date
conceptual possession of property	1,810	can have as property a	can be a property of a			*conceptually possessed property*	property
conceptual possession of property by disp	1,811			*common possessor of displacement proper*	displacement of spatial point		
conceptual possession of quality	2,050	can have as quality a	can be a quality of a			*conceptually possessed quality*	quality
conceptual possession of structure	4,795	can have as structure a	can be a structure of a	*conceptual possessor of structure*	individual object	*conceptually possessed structure*	structure
conceptual possession of decompositio	1,800	can have as decomposition structu	can be a decomposition structure of a				

Specialization hierarchy of relations	Identifier of relation type	Gellish phrase	Inverse Gellish phrase	First role	Player of first role	Second role	Player of second role
conceptual possession of atomic structi	4,796	can be of substance	can be a material of construction of a				
conceptual requirement for possess	4,995	shall be of substance	shall be a material of construction of a				
conceptual role for set of classes	4,945	can together have a role as a	can be a role for collection of a	*conceptual possessors of a role*		*conceptually played role*	role
conceptual requirement for possession of an asp	4,956	shall have as aspect a	shall be an aspect of a				
conceptual requirement for aspect of possess	5,239	shall be an aspect of a possessor c	shall have a possessor with as aspect a				
conceptual role of an individual thing	5,229	has conceptually a role as	is conceptually a role of	*conceptual possessor of a role*	concept	*conceptually possible role*	role
conceptual aspect of a part of a physical object	5,247	can have a part with as aspect a	can be an aspect of a part of a	*conceptual possessor via part*	physical object	*conceptually possessed by part*	aspect
conceptual role of a part of a physical object	1,301	can be a purpose of a part of a	can have a part with as role a				
conceptual requirement for role of a part o	5,237	shall be a function of a part of a	shall have a part with as role a				
conceptual role of an aspect of a part of a phy	5,258	can have a part with as aspect in rc	can be a role of an aspect of a part of a			*conceptually possessed role by part*	
conceptual requirement for role of an aspe	5,238	shall have a part with as aspect in (	shall be a role of an aspect of a part of a				
conceptual requirement for a role of a subtype	5,346	shall have a part of a type with as r	shall be a role of type of a part of a				
conceptual aspect of a whole of a physical objec	5,249	can have a whole with as aspect a	can be an aspect of a whole of a	*conceptual possessor via assembly*	physical object	*conceptually possessed via asseml*	aspect
conceptual role of an aspect of a whole of a p	5,259	can have a whole with as aspect in	can be a role of an aspect of a whole of a				
conceptual requirement for role of an aspe	5,252	shall have a whole with as aspect i	shall be a role of an aspect of a whole of a				
conceptual possession of a possessed aspect	5,317	is by definition a possessor of a	is by definition a possessed aspect of a				
conceptual requirement for a possession of a	5,282	commonly has as possessed aspe	shall be a possessed aspect of a				
conceptual requirement for a possessed aspe	5,344	shall be a possessed aspect of a pl	shall have a player with as aspect in role a				
definition of a type as role of a subtype of physic	5,347						
conceptual requirement of a role by a relation	2,076	can require as role a	can be required by a	*conceptual requirer of a role by a relation*	relation	*conceptually required role by a rela*	role
conceptual involvement in an occurrence	4,648	can have a role in a	can involve a	*conceptual function*	concept	*conceptual involver*	occurrence
conceptual involvement as subject	4,649	can be a subject in a	can have as subject a	*conceptual subject*			
conceptual requirement for involvement as su	5,017	shall be a subject in a	shall have as subject a				
conceptual requirement for a player of a role i	5,387	shall be related to something that is	shall be a role of something that is involved in an activity regarding a				
conceptual requirement for a performer of	5,389	shall be related to someone who is	shall be a role of someone who performs an activity on a				
conceptual requirement for a player of a role i	5,390	shall have a part that shall be relate	shall be a role of something that is involved in an activity regarding a part of a				
conceptual requirement for a performer of	5,391	shall have a part that shall be relate	shall be a role of someone who performs an activity on a part of a				
conceptual involvement as performer	4,650	can be a performer of a	can have as performer a	*conceptual performer*			
conceptual requirement for involvement as pe	5,019	shall be a performer of a	shall have as performer a				
conceptual involvement as controller	4,651	can be a controller of a	can be controlled by a	*conceptual controller*			
conceptual requirement for involvement as cc	5,171	shall be a controller of a	shall be controlled by a				
conceptual involvement as aid	4,653	can be an aid in a	can have as aid a	*conceptual aid*			
conceptual involvement as tool	4,652	can be a tool in a	can use as tool a	*conceptual tool*			
conceptual involvement as input	4,783	can be an input for a	can have as input a				
conceptual involvement as consumed	5,172	can be consumed by a	can be a consumer of a				
conceptual involvement as output	4,784	can be an output of a	can have as output a				
conceptual involvement as product	5,173	can be produced in a	can be a production process of a				
conceptual involvement as committer	5,040	can be a committer of a	can be committed by a	*conceptual committer*	person	*conceptually committed*	act
conceptual involvement as place of occurrence	5,085	can be a place of occurrence of a	can have as place of occurrence a	*conceptual place of occurrence*			
conceptual involvement as planner	5,146	can be a planner of a	can be planned by a	*conceptual planner*			
conceptual involvement as scheduler	5,148	can be a scheduler of a	can be scheduled by by a	*conceptual scheduler*			
conceptual involvement as enabler	5,170	can be an enabler of a	can use as enabler a	*enabler*			
common relation	4,699						
common possession of plural aspect	4,940						
common options for possession of an aspect	4,846	can be either of the	are options for a	*common options possessor*	aspect	*common options*	collection of classes
conceptual purpose of relation	4,898			*conceptually intended relation*	conceptual relation between members of classes		
conceptual purpose of approval	1,608	can be approved for a	can be a reason for	*intended approval for common purpose*	approval of object		
conceptual purpose of protection	1,814	can be meant as protection against	can be a purpose of protection by a	*conceptually intended protection*	conceptual protection of a physical object		
conceptual purpose of segregation	1,815	can be meant for segragation of a	can be a purpose for segregation b	*conceptually intended segregation*	conceptual segregation of a physical object		
conceptual involvement in relation	4,900			*conceptually involved in relation*	individual object	*conceptual involver in relation*	relation
conceptual coordinate in coordinates point	1,801	can be a coordinate in a	can have as coordinate a	*conceptually located*	spatial point	*conceptual reference coordinates*	conceptual coordinates in coordinate syster
conceptual involvement in correlation	4,740	can be a parameter in a	can have as parameter a	*conceptual parameter*	aspect	*conceptual correlator*	correlation
factorization of scale	1,866	has as scale factor a	is a scale factor in	*factor*	scale	*mathematical factorization product*	scale
conceptual involved role in relation	4,787	can have in the first role a	can play the first role in a				
conceptual quantification of a characteristic o	4,757	can be quantified on scale	can be a scale for a	*conceptually quantifiable characteristic*	characteristic	*conceptually quantifying scale*	scale
conceptual requirement for quantification c	5,051	shall be presented on scale	shall be the scale for a				
conceptual involver role in relation	4,841	can have in the second role a	can play the second role in a				
common involvement in relation	4,937						
common being in state	1,607	is by definition in state	is a kind of state of a			*commonly possessed state*	state
conceptual condition for occurrence	4,972	can be a condition in a	can be a condition evaluator of a	*conceptual condition*	quality	*conceptually conditioned function*	conditional occurrence function
conceptual action on satisfied condition	4,976	can be a required action in a	can be a triggering function of a	*conceptually required action*	occurrence	*conceptually triggering function*	conditional occurrence function
conceptual action on unsatisfied condition	4,979	can be an alternative action in a	can be an alternative action triggeri	*conceptual alternative action*	occurrence	*conceptual alternative action trigge*	conditional occurrence function
conceptual compliancy to a criterion	4,902	can contain a criterion for a	can be compliant with a	*conceptual compliancy criterion*		*conceptually compliant*	individual object
conceptual requirement for compliancy to a criter	4,751	shall be a specification for a	shall be compliant with a	*conceptually required compliancy criterion*	information	*conceptually specified*	individual object

Specialization hierarchy of relations	Identifier of relation type	Gellish phrase	Inverse Gellish phrase	First role	Player of first role	Second role	Player of second role
common requirement for compliancy to a crite	5,398	is a specification for a	shall commonly be compliant with				
common requirement for qualification of an aspe	4,984	shall be constrained by	is a constraint for a				
common maximum requirement for qualificati	4,985	shall be at most	shall be a maximum qualified aspect of a				
common minimum requirement for qualificatic	4,986	shall be at least	shall be a minimum qualified aspect of a				
common equality requirement for qualification	4,987	shall be	shall be a qualified aspect of a				
common compliancy to a criterion	5,399	is a criterion to which compliancy is	is commonly compliant with				
conceptual requirement for compliancy of a relat	5,422	shall be a specification for somethi	shall be related to something that shall be compliant with a				
conceptual being in state	4,934	can have as state a	can be a state of a			*conceptually possessed state*	state
conceptual subjection to correlation	4,922	can have as correlation a	can be a correlation for a	*conceptual subject of correlation*	individual object	*conceptually governing correlation*	correlation
conceptual containment of an individual	4,942	can be contained by a	can be a container of a	*conceptually contained*	physical object	*conceptual container*	physical object
conceptual experience of an aspect by an individual	5,061	can experience as aspect a	can be experienced by a	*conceptual experiencer*		*conceptually experienced*	
conceptual requirement for experience of an asp	5,245	shall sense a	shall be experienced by a				
conceptual experience of an aspect in role	5,355	can experience as aspect in role a	is by definition a role of an aspect experienced by a				
conceptual location relative to a physical object	5,362	can be located at a	can be a location of a	*conceptually referred locator*		*conceptually located*	spatial point
conceptual requirement for a location relative to	5,394	shall be located at a	shall be a location of a				
conceptual requirement for a location in a phy	5,400	shall be located in a	shall be a location with in it a				
common location relative to a physical object	5,403	is by definition located at a	is a definition location of a				
conceptual requirement for interaction with a physic	5,370	is defined as an aspect of somethir	shall interact with something with as aspect a				
conceptual requirement for an involvement in an oc	5,401	shall have a role in a	shall involve a				
conceptual requirement for an involvement as in	5,417	shall have a role as input in a	shall have as input a				
conceptual requirement for an involvement as ou	5,418	shall have a role as output of a	shall have as output a				
definition of the kind of elements in a kind of pluralit	5,402	is defined by elements each of whi	can define the elements of a				
conceptual classification of an individual	4,991	can be classified by a	can classify a	*conceptually classified*	concept	*conceptual classifier*	concept
conceptual requirement for classification of an indivi	4,994	shall be classified by a subtype of	has a subtype that shall classify a				
conceptual requirement for classification of a par	5,392	shall have a part that shall be class	has a subtype that shall classify a part of a				
conceptual requirement for classification of a	5,419	shall have a part that shall be class	has a model os subtype that shall classify a part of a				
conceptual requirement for classification of a too	5,420	shall have a tool that shall be class	has a subtype that shall classify a tool for a				
conceptual requirement for classification of a	5,421	shall have a tool that shall be class	has a model as subtype that shall classify a tool for a				
conceptual requirement for classification by elemen	5,006	shall be classified by one of the	are options for classification of a			*conceptual classifiers*	
conceptual requirement for classification of an indivi	5,397	shall be classified by a model of	has a model as subtype that shall classify a				
conceptual classification of any element in plurality	5,013	can collectively be classified by a	can be classifier of each element o	*conceptually collectively classified*		*conceptual classifier of collection*	
conceptual requirement for classification of any eler	5,014						
hierarchical relation between classes	5,052			*narrower concept*		*wider concept*	
required specialization of class	5,111	shall be defined to be a specializati	shall be defined to be a generalizat	*defining class*		*defined class*	
constraining a collection of classes	5,095	each of which is a specialization of	is a generalization of each element	*constrained collection*	collection of classes	*constraining class*	concept
possible kind of role for kind of thing	5,325						
intended kind of role for kind of thing	5,326						
relation between an individual and a class	4,719	is related to a	can be related with	*relating individual*	individual object	*related class*	concept
classification of an individual thing	1,225	is classified as a	is a classifier of	*classified individual*	individual object	*classifier for an individual*	individual object
classification of an occurrence	1,283			*classified occurrence*	occurrence	*classifier for an occurrence*	occurrence
essential classification of an individual	1,284						
classification of a physical object	1,286			*classified physical object*	physical object	*classifier for a physical object*	physical object
classification of an aspect	1,287			*classified aspect*	aspect	*classifier for an aspect*	mathematical space
classification of information	1,285			*classified information*	information	*classifier for information*	information
classification of a role	1,626						
classification of encoding aspect	1,735			*classified encoding aspect*	encoding aspect	*classifier for an encoding aspect*	encoding aspect
classification of recognition aspect	1,736			*classified recognition aspect*		*classifier for a recognition aspect*	
classification of reference aspect	1,761			*classified reference aspect*	reference aspect	*classifier for a reference aspect*	reference aspect
qualification of an aspect	4,703	is constrained by a	is a common constraint of			*qualifying aspect*	
equality of an aspect to qualitative aspect	5,020	is qualified as	is common value of				
inequality of an aspect	5,021	is unequal in height as	is a common value unequal to				
relative magnitude above qualitative aspect	5,022	is higher than	is common value that is lower than				
relative magnitude below qualitative aspect	5,023	is lower than	is common value that is higher than				
approximation of an aspect	5,024	is approximately equally high as	is common value that is approximately equal to				
qualification of an aspect of being within rang	5,028	is within range	is range around aspect				
incidental classification of an individual	1,308	is temporary classified as	is indefinite classifier of				
classification of a physical object by role	1,588	has a role as a	is a kind of role fulfilled by			*classifier for a role*	role
scale for quantification of an aspect	1,732	is quantified on scale	is scale of	*qualified quantification*	quantification of an aspect by mathe	*qualifier for quantification*	scale
qualification of geometric scale	1,854			*qualified geometric scale*	geometric scale	*qualifier for geometric scale*	geometric scale
qualification of time scale	1,855			*qualified time scale*	time scale	*qualifier for time scale*	time scale
classification of collection of individuals	5,043	are collectively classified as a	is classifier of collection	*classified plural individual*	plural individual	*classifier for a plural individual*	plural individual
conceptualization of an individual thing	5,093	is conceptually a	is a conceptualization of				
conceptualization of an aspect	4,702						
conceptualization of fact	5,094						

Specialization hierarchy of relations	Identifier of relation type	Gellish phrase	Inverse Gellish phrase	First role	Player of first role	Second role	Player of second role
conceptual purpose of an individual aspect	1,365	is intended for a	can be a purpose of	*intended aspect for common purpose*	aspect	*commonly envisaged by an individu*	occurrence
conceptual connection with a physical object	1,629	is connected with a	is classifier of item connected with	*connected individual*	physical object	*connected member*	physical object
conceptual logical connection with a physical object	1,634	is logically connected with a	can have a logical connection with				
conceptual physical connection with a physical obje	1,907	is physically connected with a	can have a physical connection with				
composition of an individual from a part with a classific	1,728	has a part that is classified as a	is a classifier of a part of	*whole for kind of part*	individual object	*common part for individual*	individual object
role in life of members of a subtype of physical object	1,900	can be the role of a	can have as role	*played role by individual*	role	*common role player*	physical object
control of common reference by authority	1,918	is the controlling authority of a	is a kind of reference that is control	*controlling authority for common reference*	organization	*commonly controlled*	conceptual reference to object by reference
conceptual role for individual	4,712						
conceptual function in individual occurrence	1,992	requires as function a	can be played in	*requirer of conceptual function*	occurrence	*conceptually required function*	function
possible conceptual role for individual	4,710						
capability of a physical object to fulfil a role of a k	1,972	has the capability to act as a	is a kind of role that can be fulfilled	*possessor of capability*	physical object	*possibly fulfilled role of a kind*	role
capability of person to fulfil a role of a kind	4,654	has skill to act as a	is a capability of				
intended conceptual role for individual	4,711						
carrying of information about class	4,723	contains information about a	can be described in	*individual information carrier*	physical object	*commonly described in information*	individual object
possession of conceptual aspect	4,746	has aspect that is conceptualized a	is a conceptualization of an aspect of				
required compliancy to a criterion	4,753	shall individually be compliant with	is a compliancy criterion for	*required complier*	individual object	*compliancy criterion*	information
adoption of information	4,813			*adopted*	information	*adopter*	lifeform
possession of common aspect	4,853	has aspect that is qualified as	is a qualification of an aspect of				
common purpose of relation	4,890			*intended relation for common purpose*	relation between individual things		
common purpose of protection	1,738	is protection against a	can be prevented by				
common purpose of segregation	1,739	is segregation for a	is common purpose of segregation				
comparison of a characteristic with reference class	4,892			*compared characteristic for class*	characteristic	*reference common characteristic fo*	characteristic
common compliancy to one of the criteria	4,950	are specifications for a	shall commonly be compliant with c	*compliancy criteria for members of class*	plural aspect	*commonly compliant*	individual object
involvement in a subtype of occurrence	5,066			*potentially involved*	physical object	*potentially involving class*	occurrence
involvement as performer of a subtype of occurrenc	5,067	is performer of a	can be performed by	*potential performer*		*potentially performed class*	
involvement as subject of a subtype of occurrence	5,131	is subject of a	can have as subject	*potential subject*		*potentially subjecting class*	
involvement as place of a subtype of occurrence	5,395	is a place of occurrence of a	can be performed at				
potentially being supplier of members of a subtype of p	5,154	can be supplier of a	can be supplied by	*potential supplier*	organization	*potentially supplied*	
potentially being maker of members of a subtype of ph	5,162	can be maker of a	can be made by	*potential maker*		*potentially made*	
potentially being manufacturer of members of a sub	5,157	can be the manufacturer of a	can be manufactured by	*potential manufacturer*	organization	*potentially manufactured*	
relation between collections	4,748						
difference of sets	2,847	is the difference of sets	are sets with as difference	*difference collection*	plural object	*compared collections*	plural object
union in set	2,853	is disjoint union of	contains elements that are united ir	*united*	plural object	*union*	plural object
subset of set	4,856	is a subset of	is a superset of	*subset*		*superset*	
intersection of sets	5,255	is an intersection of	are sets with as intersection	*intersection*		*intersected collections*	
definition of a table by a list of items	5,296	is defined by the items in list	is the list that defines the items of	*defined table by items*	table	*defining list of items*	list
definition of a table by possessed aspects	5,297	is defined by the possessed aspect	is the list that defines the possesse	*defined table by possessed aspects*	table	*defining list of possessed aspects*	list
quantification of list of aspects on scales	5,311	is a list that is quantified on scales	are scales for the quantification of	*quantified list*		*quantifying scales*	
definition of a table by relation types	5,318	is defined by the relation types in lis	is the list that defines the relation ty	*defined table by relation types*		*defining list of relation types*	
plural relation	4,756						
options for qualification of an aspect	4,850	is qualified as one of the	are options for qualification of	*optionally qualified*	aspect	*options*	plural aspect
context for applicability of fact	4,818	is applicable in the context of	is a context for validity of	*applicable*	relation	*application context*	anything
close relation	5,230	is closely related to	is closely related with				
loose relation	5,231	is loosely related to	is loosely related with				
relation between a single thing and a plurality	5,322			*single relator*		*related plurality*	
collection relation	2,846	is collected in	is a collection for	*collected*	anything	*collecting plurality*	plural object
collection of an individual thing	1,227	is an element of	is a plural individual for	*collected individual*	individual object	*collecting individual plurality*	plural individual
collection in set of classes	4,730	is an element in collection of classe	is a collection of classes including	*collected class*	concept	*collecting plural class*	collection of classes
instantiation	4,734	is an instance of	is an entity with instance				
composition of list	5,314	are components in list	is a list of	*composing list items*	anything	*composed list*	plural object
arrangement relation between a first thing and its pl	5,331	is the first element in	is a list with as first element				
classification of any element in plurality	4,843	is classifier of each element of	each of which is classified as a	*collective classifier*	individual object	*collectively classified*	plural object
individuals correlated by individual correlation	5,348	has as parameter list	is the parameter list of				
parameters in correlation	5,349	has by definition as parameter list a	shall have as parameter list a				

10.3 Appendix C, Upper ontology concept definition example

The tables in Appendix A and Appendix B are both derived from a Gellish Table with the formal definition of the Gellish English language. This appendix presents a part of that table with the definition of the top of the Gellish Ontology in English and its translation in Dutch (Nederlands).

Note that the relations have their hierarchy, the roles have their hierarchy and the objects that play the roles have their hierarchy, whereas all hierarchies are consistent and end in the top concept called 'anything'.

					Gellish grammar definition example							
54	71	16	2	44	101	1	60	3	15	45	201	4
Language of left hand object name	ID of Context for left hand object name	Name of Context for left hand object name	Unique id of left hand object	Simultaneous left hand cardinalities	Left hand object name	Fact id	Relation type ID	Relation type name	Unique id of right hand object	Simultaneous right hand cardinalities	Right hand object name	Full definition
English	193,259	ontology	4,990		concept	1,001,005	1,146	is a specialization of a	730,000		anything	is a anything which is a commonality of things.
English	193,259	ontology	2,852		single object	1,001,079	1,146	is a specialization of a	4,990		concept	is a concept that has a count of one.
English	193,259	ontology	2,850		relation	1,001,074	1,146	is a specialization of a	2,852		single object	is a single object that indicates that two things have something to do with each other. It is an expression of a single fact. If one of the related things is a plurality, then the relation implies multiple facts.
English	193,259	ontology	2,850		relationship	1,006,688	1,981	is a synonym of	2,850		relation	
English	193,259	ontology	2,850	1,1	relation	1,006,209	4,731	requires a role-1 as a	4,729	1,1	relator	
English	193,259	ontology	2,850	1,1	relation	1,006,700	4,733	requires a role-2 as a	4,824	1,1	related	
English	492,015	Gellish English	2,850		might be related to	1,008,567	1,981	is a synonym of	2,850		relation	
English	492,015	Gellish English	2,850		might be related with	1,008,568	1,986	is an inverse of	2,850		relation	
nederlands	492,016	Gellish nederlands	2,850		zou gerelateerd kunnen zij	1,008,569	1,981	is a synonym of	2,850		relation	
nederlands	492,016	Gellish nederlands	2,850		zou gerelateerd kunnen zij	1,008,570	1,986	is an inverse of	2,850		relation	
English	193,259	ontology	2,850	1,1	relation	1,006,754	4,714	can have a role as a	4,820	0,1	applicable	
English	193,259	ontology	2,850	1,1	relation	1,007,015	4,714	can have a role as a	4,897	0,1	intended relation for purpose	
English	193,259	ontology	2,850		relation	1,007,530	5,229	has conceptually a role as	4,983		conceptual involver in relation	
English	193,259	ontology	2,850	1,1	relation	1,001,108	5,229	has conceptually a role as	2,080	0,1	conceptual requirer of a role by a relation	
English	193,259	ontology	4,718		relation between classes	1,006,168	1,146	is a specialization of a	2,850		relation	is a relation that indicates that members of a class have commonality of this kind in their relations to members of the related class.
English	193,259	ontology	1,146		specialization of class	1,001,142	1,146	is a specialization of a	4,718		relation between classes	is a relation between classes that indicates that the subclass has more constrained criteria for membership than the superclass; and each member of the subclass is also a member of the superclass.
English	193,259	ontology	1,146	1,1	specialization of class	1,003,487	4,733	requires a role-2 as a	3,817	1,1	supertype	
English	193,259	ontology	1,146	1,1	specialization of class	1,003,488	4,731	requires a role-1 as a	3,818	1,1	subtype	
English	492,015	Gellish English	1,146		is a specialization of	1,001,002	1,981	is a synonym of	1,146		specialization of class	
English	492,015	Gellish English	1,146		is a generalization of	1,001,003	1,986	is an inverse of	1,146		specialization of class	
English	492,015	Gellish English	1,146		is a subtype of	1,004,601	1,981	is a synonym of	1,146		specialization of class	
English	492,015	Gellish English	1,146		is a supertype of	1,004,602	1,986	is an inverse of	1,146		specialization of class	
nederlands	492,016	Gellish nederlands	1,146		is een specialisatie van	1,004,603	1,981	is a synonym of	1,146		specialization of class	
nederlands	492,016	Gellish nederlands	1,146		is een generalisatie van	1,004,604	1,986	is an inverse of	1,146		specialization of class	
English	193,259	ontology	4,729		relator	1,006,215	1,146	is a specialization of a	160,170		role	is a role that is played by an object involved in a fact that is expressed by the relation tha requires the role. Typically in an active role.
English	193,259	ontology	730,000	1,n	anything	1,006,236	4,714	can have a role as a	4,729	0,1	relator	
English	193,259	ontology	4,824		related	1,006,761	1,146	is a specialization of a	160,170		role	is a role that is played by an object involved in a fact that is expressed by the relation tha requires the role.
English	193,259	ontology	730,000	1,n	anything	1,007,422	4,714	can have a role as a	4,824	0,1	related	
English	193,259	ontology	4,990	1,1	concept	1,001,122	5,229	has conceptually a role as	3,817	0,1	supertype	
English	193,259	ontology	4,990	1,1	concept	1,001,123	5,229	has conceptually a role as	3,818	0,1	subtype	
English	193,259	ontology	3,818		subtype	1,002,690	1,146	is a specialization of a	5,260		narrower concept	is a narrower concept which indicates the class that is narrower and has more constraining criteria for membership than the superclass.
English	193,259	ontology	5,260		narrower concept	1,008,728	1,146	is a specialization of a	4,729		relator	is a relator which indicates the lower class in a non circular hierarchy.
English	193,259	ontology	3,817		supertype	1,002,689	1,146	is a specialization of a	5,261		wider concept	is a wider concept which specifies the class that is broader and has less constrained criteria for membership than the subclass.
English	193,259	ontology	5,260		narrower concept	1,008,728	1,146	is a specialization of a	4,729		relator	is a relator which indicates the lower class in a non circular hierarchy.
English	193,259	ontology	5,261		wider concept	1,008,729	1,146	is a specialization of a	4,824		related	is a related which indicates the upper class in a non circular hierarchy.

Curriculum vitae

Andries van Renssen

Born at Utrecht on 9 March 1946.

1958 – 1964	Secondary school HBS-B of the 'Christelijk Lyceum' at Utrecht.
1964 – 1970	Study Mechanical Engineering and Physical Technology at the Eindhoven University of Technology. Thesis: 'The economic optimization of the design of a fish-protein concentrate plant'.
1968 – 1971	Teacher in Physics at the 'Gereformeerde Scholengemeenschap Guido de Bres' at Amersfoort.
1971 – 1976	Software engineer at Comprimo Engineers and Contractors B.V.
1976 – 1980	(Senior) process engineer at Comprimo Engineers and Contractors B.V.
1980 – 1981	Department head Information analysis at Systware B.V.
1981 – now	Various jobs at Shell, among which: - Department head software engineering - Department head artificial intelligence systems - Department head refinery logistics systems - Department head information management - Principal engineer / consultant information management

The research for this PhD thesis was performed at:
- Shell Internationale Petroleum Maatschappij B.V.,
- Shell Global Solutions International B.V.,
- and as a member of the International Standardization Organization ISO TC184/SC4.