A survey of Top-Level Ontologies

To inform the ontological choices for a Foundation Data Model

Version 1

CONSTRUCTION INNOVATION HUB

INDUSTRIAL STRATEGY

UK Research and Innovation

Contents

1 Introduction and Purpose

The Centre for Digital Built Britain has been tasked through the Digital Framework Task Group to develop an Information Management Framework (IMF) to support the development of a National Digital Twin (NDT) as set out in “The Pathway to an Information Management Framework” (Hetherington, 2020). A key component of the IMF is a Foundation Data Model (FDM), built upon a top-level ontology (TLO), as a basis for ensuring consistent data across the NDT.

This document captures the results collected from a broad survey of top-level ontologies, conducted by the IMF technical team. It focuses on the core ontological choices made in their foundations and the pragmatic engineering consequences these have on how the ontologies can be applied and further scaled. This document will provide the basis for discussions on a suitable TLO for the FDM. It is also expected that these top-level ontologies will provide a resource whose components can be harvested and adapted for inclusion in the FDM.

Following the publication of this document, the programme will perform a structured assessment of the TLOs identified herein, with a view to selecting one or more TLOs that will form the kernel around which the FDM will evolve. A further report – The FDM TLO Selection Paper – will be issued to describe this process in late 2020.

2 Approach and contents

The approach has three parts:

1. collect candidate top-level ontologies (2.1)
2. develop assessment framework (2.2)
3. assess candidate top-level ontologies against the framework (2.3)

These are described in more detail below.

A note on the terminology used in the report is contained in 2.4 and Appendix K.

2.1 Collect candidate top-level ontologies

A long list of possible candidates for TLO content that might be useful for the construction of the FDM has been drawn up and reviewed.

Candidate ontologies were identified both through extensive desktop research, and through the experience and domain knowledge of the expert community involved in bringing this report together.

In identifying candidates, the net was thrown as wide as possible to identify as much useful content as possible. Thus, though the focus is on ontological commitment, the list includes data models that are generic in nature (ones without an explicit ontological foundation) as these are likely to have some useful ontological content. The candidates are listed in Appendix D and are available online within the IMF Developers Network on the Digital Twin Hub, www.digitaltwinhub.co.uk.

2.2 Develop assessment framework

In compiling this report, a first-pass assessment framework was developed to facilitate the initial testing of the spectrum of available TLOs and other ontological models against the needs of the programme. This assessment is distinct from the future activities around the further down-selection of TLOs to a selected core for the FDM.

An ontology is (according to Jonathon Lowe in The Oxford Companion to Philosophy) “the set of things whose existence is acknowledged by a particular theory or system of thought.” When we interpret a dataset, working out what the data refers to, we are acknowledging that the dataset commits to these things existing. We are committing to an ontology – making an ontological commitment.

There are a variety of ways of making these commitments. The purpose of a top-level ontology is to enable us to make the choice of ontological commitments in an explicit and consistent way.

There have been a series of attempts to get to grips with the kinds of choices of ontological commitments that ontologies can make. They provide a reasonable starting point but need substantial further work to provide a comprehensive framework for making choices across a broad range of commitments. We have developed a comprehensive choice component for this framework; one that is suitable for assessing information system TLOs.

With this choice component in place, we can look at how these choices shape an ontology’s underlying architecture and so what it can do and how it does it. We can also characterise the candidate TLOs in terms of whether they make a choice and which they choose.

The assessment framework has three levels.

1. A general level, which looks at whether the TLO makes an ontological commitment and the strength of this commitment (see 4.1).
2. A formal level, which looks at how the formal structure of the TLO has been impacted by its ontological commitment (see 4.2).
3. A universal level, which looks at how the TLO addresses the individual universal choices (see 4.3).

Further details on the basis for the assessment process are described in section 4.

2.3 Assessment of candidate top-level ontologies against the framework

The assessment framework described above has been applied to the candidate TLOs. Only a small number of TLOs made their ontological commitments explicit, enabling a clear, simple assessment. In other cases, the choices could be clearly inferred from the documentation available. However, in many cases we could not determine the choice.

It could be that no choice was intended, or that the choice is not documented, or not documented sufficiently clearly in the material we have reviewed or some other reason. Rather than trying to classify the exact reason for each case, where we have not noted a choice, we have marked the cell 'not assessed'. In some cases, typically ISO standards, the documentation is not publicly available, so to make this clear we have marked these 'not available'. The results of the assessment are stored online within the IMF Developers Network on the Digital Twin Hub, www.digitaltwinhub.co.uk. with a summary provided in section 6.

In addition, a brief overview of the candidate top-level ontologies with any useful additional points is given in Appendix F. These include a graphical representation of the TLOs where one has been found. A comparison shows clearly the diversity of top-level structures.

2.4 Terminological note

This is a topic that crosses multiple disciplines, including information systems, computer science, philosophy and linguistics. Confusingly, many terms are used with different senses across these disciplines and even within them. Accordingly, we will attempt, where possible, to use terms with the senses that they have in the disciplines in which they arise – to minimise any increase in the confusion and encourage cross-disciplinary consistency.

There is one critical case where there is little consistency, this is a term for objects in general. These are sometimes also known as entities or things – through all three of these terms have restricted senses in various sub-disciplines. We propose to use the term 'object' here, rather than 'entities' or 'things' unless the context requires it as this is the term most consistently used in philosophy, and can be found with this sense as far back as the 17th century (Locke, 1975). Where needed we will qualify the term – for example, material objects. If we are using the term in a different sense, we will clearly note this. Of course, all three terms will appear in the extracts from the TLO documentation.

Other terms are defined in the glossary in Appendix K.

3 Assessment framework – development basis

The development of the assessment framework is driven by two broad considerations:
general ontological requirements (3.1); and
the requirement for an overarching ontological architecture (3.2)
These are described below.

3.1 General ontological requirements

The general requirements provide a context for the framework and comprise:

1. the need for an ontological framework (3.1.1);
2. how the need for an ontological framework translates into making choices of ontological commitments (3.1.2 and 3.1.3);
3. the requirements that arise from the lack of prior knowledge (3.1.4); and
4. the need for consistent independent (federated) development (3.1.5).

3.1.1 Real world ontology framework

If one wants to share data from different systems, then one needs to have something like a common framework within which to share it. When the data is in this common framework, its meaning needs to be clear and unambiguous to the systems sharing it. This is often called semantic interoperability. For example, it needs to be clear and unambiguous whether data items (for example, rows on a table) from two systems are referring to the same object or different objects in the 'real world' – for example the UNICLASS Code 'Ac_05_50_91 – Timber sourcing' is marked as mapping to NBS Code '45-60-90/340 – Timber procurement'. To implement this systematically, one needs to be clear and unambiguous about what these objects in the real world are. This involves knowing the ontology, in other words, knowing the set of objects that the common framework assumes exist.

3.1.2 Choices of ontological commitments

Unfortunately, when one starts to look closely, it is neither clear nor unambiguous exactly what the objects in the real world are. Ontologically, there are a variety of ways that one can take the real world to be. However, for our assessment, these can be crystallised into a small number of focussed choices – called ontological commitments – which build up into an integrated ontological architecture.

A key purpose of this paper is to provide a framework for understanding the range and nature of the ontological commitments and apply this to the collected top-level ontologies. Thus, providing the groundwork for the choice of an appropriate ontological architecture.

3.1.3 Implicit and explicit choices

Understandably, most of the datasets currently available are not clear about their ontological architecture, which of these ontological commitments have been made – their choices, such as they are, are implicit. In practice, datasets often make these choices implicitly – choosing, without realising, one way in one area and another way in another area. This point is often made in philosophy textbooks, see, for example, (Lowe, 1998).

The assessment framework gives a clear picture of the range of these choices. With this in hand, when selecting or developing a top-level ontology, one can be clear which choices are made (and which left to chance); and so have some idea how this will be, in turn, reflected in the data structures and data that are implemented.

3.1.4 Lack of prior knowledge

Usually the developers of a system of systems (which will include behavioural, societal and human elements) have prior knowledge of some of the systems that will use the common framework. However, often other requirements will arise as new systems are added to the common framework – of which it will have no prior knowledge. Hence, the framework needs to be sufficiently expressive to accommodate them. More specifically, care needs to be taken not to adopt ontological commitments which unnecessarily restrict its ability to express meanings that probably will occur in data from new source systems. For example, the commitment choices include whether to restrict the types to first order (where types cannot have types as instances). One cannot just assume that as the current data set only has first order types, then one can restrict oneself to these – one also needs some confidence that a requirement for higher order types will not emerge in the future. In this case, there is then a requirement to be sensitive to how a choice of ontological commitment might restrict useful expressivity.

3.1.5 Consistent independent (federated) development

Many systems for connecting systems, like the NDT, have a hub and spoke structure where spoke systems map their data into the central hub system. It is likely that these mappings will be done independently. In many cases, the content from different systems will overlap. Where this happens, the mappings produced should be equivalent. Adopting an ontological approach is a big step towards achieving this because it provides an independent basis for establishing identity between the systems. Fine tuning the choice of ontological commitments to ensure a clear notion of what is referred to is a good further step. In this case, there is then a requirement to be sensitive to how a choice of ontological commitment can be clearer about what is referred to and so give rise to equivalent independent mappings.

3.2 Overarching ontological architecture framework

As mentioned earlier, there have been several attempts to get to grips with the kinds of choices of ontological commitments that TLOs should make; these are listed in Appendix G. These attempts provide a good starting point and we refer to them when this is useful, usually in the technical appendices. However, all these lists are partial and, in some cases, not based upon sufficient familiarity with the relevant research. Furthermore, none of them provide an over-arching organising structure; one that provides a framework for understanding and assessing choices across a range of commitments. We develop this framework in 3.2.1 below.

3.2.1 Basis

There needs to be a clear and solid underlying basis for the framework. There is an established set of criteria for assessing what a good ontology is, based upon what makes a good scientific theory – listed in Appendix H. One of these criteria, simplicity, provides us with a good basis for a broad assessment of the architecture and is broadly outlined below, with more technical detail in Appendix I.

Simplicity can be thought of as having two aspects; structural and ontological. Where structural or syntactic simplicity is roughly concerned with the shape of the organising structure, ontological simplicity is roughly concerned with the number of objects.

For structural (syntactic) simplicity we look at the characteristic ways the ontological commitments shape the organising structure (see section 4). For ontological simplicity we do some more analysis to establish broad brush accounting principles (as set out in 3.2.1.1 in 3.2.1.3 below).

3.2.1.1 Accounting for ontological simplicity

Ontological simplicity is usually associated with a number of characteristics including parsimony, explanatory sufficiency and fruitfulness. Parsimony is usually characterised as Ockham's Razor – objects are not to be multiplied beyond necessity. There is less well-known, but with an equally long history, principle of explanatory sufficiency – "the variety of entities should not be rashly diminished" (Kant, 1964). Parsimony and explanatory sufficiency, taken together, imply a kind of ontological economy, which aims for explanatory sufficiency with the minimum number of entities. We look at how to account for this first and then return to fruitfulness (see 3.2.1.3).

Simple counting of objects does not seem intuitively correct way of accounting. If Ann claims that the damage to my carrot patch was caused by exactly 100 rabbits, and Ben claims it was done by 101 rabbits, then it is hard to feel that Ben's theory is any way less economical than Ann's, or that Ben has multiplied entities in a way that calls for concern. If Ann now claims the damage to my carrot patch was caused by 5 rabbits (one type and five individuals) and Ben claims it was caused by 1 deer and 3 rabbits (two types and four individuals) – then despite both claims involving six objects, Ben's seems more complex. In the literature, this is seen as arising from a distinction between qualitative parsimony (roughly, the number of types of object) and quantitative parsimony (roughly, the number of individual objects). In the research, most people claim qualitative parsimony matters and quantitative parsimony is less relevant – making a distinction between the making of the commitment and its cost. If Ann now claims the damage was caused by 12 rabbits (one type and 12 individuals) and Ben claims it was caused by 1 deer and 3 rabbits (two types and four individuals), then Ann's claim is more qualitatively parsimonious (one versus two types) but less quantitatively parsimonious (twelve versus four individuals) than Ben's. The suggestion is that Ann's claim is more relevantly economic than Ben's.

The claim that qualitative parsimony matters more than quantitative parsimony resonates for the design and maintenance information systems. For example, function or object point analysis (e.g. FiSMA: ISO/IEC 29881 or IFPUG: ISO/IEC 20926:2009) measures are based upon qualitative (type) rather than quantitative (individual) counts. However, as has been noted, this qualitative-quantitative distinction seems too simplistic – for example, not taking account of algorithmic complexity.

3.2.1.2 The laser

There is a revised approach that seems to capture some of the relevant complexity. This uses a distinction between fundamental and derived objects and updates Occam's Razor with what Jonathan Schaffer (Schaffer, 2015) calls the laser – "do not multiply fundamental objects without necessity". He illustrates the difference between the razor and the laser with this example. Imagine Esther posits a fundamental theory with 100 types of fundamental particle. Her theory is predictively excellent and is adopted by the scientific community. Then Feng comes along and—in a moment of genius—builds on Esther's work to discover a deeper fundamental theory with 10 types of fundamental string, which in varying combinations make up Esther's 100 types of particle. This looks like a paradigm case of scientific progress in which a deeper, more unified, and more elegant theory replaces a shallower, less unified, and less elegant theory. However, under razor accounting, both the number of particles and strings are counted and therefore Feng's theory has 10 more objects and so should be replaced with Esther's. Under laser accounting though, 100 fundamental objects have been replaced by 10 – so Feng has made an improvement. Here again we have a distinction between making a commitment and its cost. We make a commitment to both fundamental and derived objects, but the cost of derived objects is significantly less than that of fundamental objects.

As Schaffer notes, what emerges from this approach is a general pressure towards a permissive and abundant view of what there is, coupled with a restrictive and sparse view of what is fundamental. As he notes, classical mereology (the relations of parts to wholes) and pure set theory (where the only sets, well-determined collections of objects, under consideration are those whose members are also sets) come out as paradigms of methodological virtue, for making so much from so little. This suggests a preference for, what has been called, plenitude – not placing unnecessary constraints on what can exist; if it is possible for something to exist, then it does. Both classical mereology and (impure) set theory exhibit this. Simplifying a little, in classical mereology, given any two objects, their fusion exists – in set theory, their set exists. Where many of the candidate TLOs make explicit their mereological position, they chose classical mereology. However, where they make explicit their position on types, only a significant minority adopt a position of plenitude.
Schaffer suggests a principle to capture this, the Ontological Bang for the Buck principle: optimally balance minimization of fundamental objects with maximization of derivative objects, especially useful ones.

3.2.1.3 Fruitfulness

In 3.2.1.1 above, fruitfulness was mentioned as being associated with simplicity. The examples provided above show, derivative objects are part of what makes a package of fundamental objects fruitful. In other words, they show that these fundamental objects can be used to produce something useful. However, as discussed, there is a need to be sensitive to both cost and benefits. If two very similar theories had roughly the same cost in terms of fundamental objects, but one had a large commitment to many useless entities but the other did not – and they were similar in all other relevant respects, this seems like overgeneration. The additional useless plenitude is more like profligacy or promiscuity – it is not fruitfulness. This gives us a 'useful' basis for assessing the TLOs.

4 Ontological commitment overview

Our overview of the framework for ontological commitments is divided into three parts:

1. Section 4.1 looks at the general choices TLOs make on whether and what kind of overall ontological commitment to make;
2. Section 4.2 looks at the overall formal structure; and
3. Section 4.3 considers the individual core commitments that lead to that structure.

More detailed technical notes are given in Appendix J.

4.1 General choices

The general choices track the ontological approach chosen by the top-level ontologies. They firstly note whether the TLO has chosen to make ontological commitments or not. They then note whether the ontological commitment is lightweight or heavyweight (see 4.1.2). Finally, they note what they have chosen to make the subject of their ontological commitments; natural language or the (foundational) real world (this is discussed in 4.1.3). Our survey includes examples of TLOs making all these choices and this range provides useful examples to compare and contrast as well as a comprehensive range of components that could be useful in developing a TLO.

4.1.1 Ontologically committed: ontological or generic

The top-level ontologies longlist was compiled to include any data models that might have content useful for the construction of a top-level ontology. Hence, one of the key conditions for inclusion is that the model must be sufficiently general to include content that might be useful.

There are cases where the TLO specifies a data structure with no intended ontological commitment; to deliberately leave open how the data is modelled. A classic indication of this is where the modeller can validly choose which data type to use in a model (whether something is modelled as an entity or attribute) based upon, typically, performance requirements. These TLOs are classified as generic; Topic Maps and Schema.org are examples of this. One consequence of this choice is that these top-level 'ontologies' are not able to harness the interoperability benefits of adopting ontological commitments mapping to the real world (discussed above).

4.1.2 High or low ontological commitment

Where TLOs are ontologically committed, the analysis reveals that some have explicitly committed to most, if not all, of the choices whereas others have only committed to a few. This gives us a good basis for distinguishing between the heavyweight TLOs that are highly committed and the lightweight TLOs that are only committed to a few.

4.1.3 Subject: appearance or reality: natural language or foundational ontology

One can broadly classify top-level ontologies into two kinds by their subject matter. The subject matter can be what a community implicitly accepts when using a language – a natural language ontology. This will take the surface structure of the language, for example the distinction between nouns and verbs (or the words it uses), as a window on the ontology. Or it can be an ontology of what 'really' exists according to science (and philosophy) – a foundational ontology. This is suspicious of the surface structure of the language as it has often turned out to be a false friend. For example, the English language classes tomatoes as a vegetable, but this has not persuaded botanists to stop classifying them as 'really' a fruit.

The natural language ontology may include merely conceived objects as well as those that happen to be actual – whereas the foundational ontology should only include actual ones, as well as some infrastructure to help assure that they are actual. The natural language ontology may focus on the linguistic structure of the language or on the concepts implied by the language. See Appendix J for a more detailed background.

Some TLOs on our longlist have explicitly stated their aspirations to one or other kind of subject matter – and provided the appropriate infrastructure. Clear examples are DOLCE as a natural language ontology; BFO and BORO as foundational ontologies. Some TLOs with no stated aspirations are clearly focussed on language and its linguistic infrastructure (nouns, verbs, etc.) and thereby categorised natural language ontologies: Wordnet and FrameNet are good examples.

Sometimes it is difficult to make the case for a classification; where there is neither a clear statement of intent nor clear infrastructure for one or other approach. These have not been classified.

Where one's focus is on the language used in a community, then, other things being equal, a natural language ontology makes a better fit. If the focus is on reality, then a model of what really exists makes more sense. However, both kinds of ontology should be considered as useful sources for components for a TLO.

4.1.4 Categorical

There is a long tradition of categorical ontologies, where the types of the ontology are meant to be comprehensive, covering all types of thing that can exist – or, at the very least, a broad swathe; Aristotle and Kant's Categories are historic examples of this. In principle, one would expect a TLO to be categorical. However, in practice, the comprehensiveness is often limited. In some cases, the scope is limited to a broad family of domains – such as MIMOSA's focus on asset information for machinery and systems. In other cases, the TLO adopts a cautious position and its ontology is explicitly left open-ended allowing for extensions that involve new top-level categories, so making the TLO technically non-categorical – BFO is an example of this.

4.1.5 General classifications

In summary, the general level has the following classifications:

category	type	choice
general	ontologically committed	ontological or generic
general	commitment level	high or low (heavyweight or lightweight)
general	subject	foundational or natural language
general	categorical	yes or no

Figure 1 provides a visual summary of this.

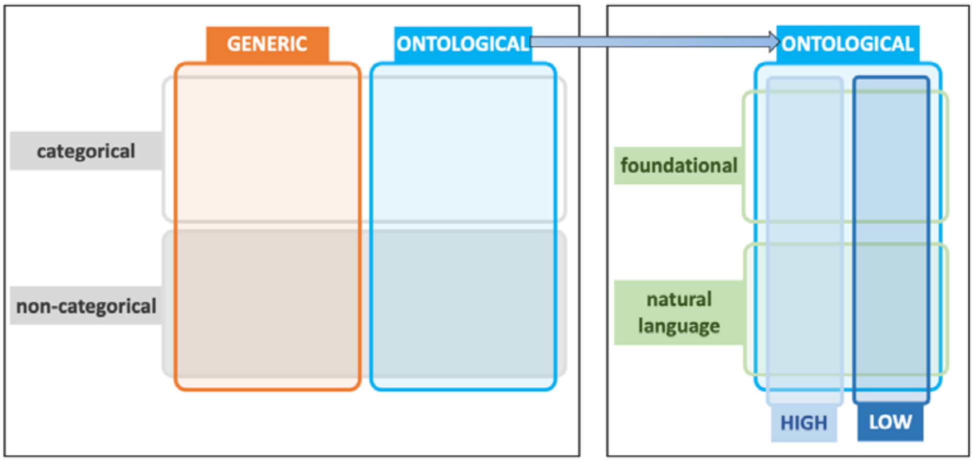


Figure 1 – General classification of the TLOs

4.2 Formal structure – horizontal and vertical

Many, if not most, of the ontological choices leave their mark on the formal structure, the ontological architecture, of the TLOs in characteristic ways; this section is about two ways we use these marks to classify them – which we tag vertical and horizontal aspects.

Three core hierarchical relations – each usually visualised upwards – provide a backbone to the TLOs. The ontological choices shape these upwards (vertical) structures in various ways – and we use these ways to characterise the impact of the choices on the TLOs on the ontological architecture.

If one looks in more detail at one of these hierarchies – the super-sub-type hierarchy – then one can see a repeating pattern of stratification across the hierarchy (horizontal) that mark particular choices. We use these to determine whether the TLO has made a particular choice.

These two ways of looking at the formal structure (the ontological architecture) – tagged vertical and horizontal aspects – map neatly onto the simplicity basis introduced above. There we divided simplicity broadly into structural and ontological – roughly the shape of the organising structure and the number of objects. The vertical aspect deals with the structure, and so structural simplicity, of the various hierarchies; roughly their shape up and down. The horizontal aspect deals with the broad ontological choices that can introduce a division across the hierarchy (horizontal stratification) – which impacts the ontological simplicity – as they increase the number of objects. Taking a broad-brush view, this section of the framework separates the vertical and horizontal aspects of the formal hierarchies.

In practical terms, these characterise the formal structures that arise in the ontological architecture from the various ontological choices. Together these form the backbone of the ontological architecture upon which the flesh of the ontology is built. Here we identify the broad structures and outline how their component ontological commitments fit into this structure. In the next section, we look inwards into the specific details of the commitments – rather than outwards at their impact of the structure.

4.2.1 Vertical aspect – varieties of hierarchies

There is a core of basic ontological hierarchical relations that are typically found in top-level ontologies; whole-part, type-instance and super-sub-type (they go by various names, these are the ones we adopt in this paper – Table 1 lists some alternatives with examples).

Table 1 – Hierarchal relations – terms

Adopted term	Alternative terms	Examples
whole-part	part of	This building has a whole-part relation to my front door (my front door is part of this building)
type-instance	instantiation, class-member, member, instance	Building has a type-instance relation to this building (this building is an instance/member of the type building – this building is a building)
super-sub-type	generalisation, subsumption, super-type, sub-type	Opening has a super-sub-type relation to door and window (door and window are sub-types of opening – doors and windows are openings)

There is debate about whether some of these are fundamental (for example, super-sub-type can be defined in terms of type-instance). There is also debate whether these all belong to the same family of relations or are distinct types. Whatever the outcome of these debates, as noted earlier, in practice these hierarchies are a key part of the backbone of the ontological architecture. One powerful way they do this is through their formal structure. Here we look at the formal properties of hierarchies and how these apply to the three relations – to see how they, together, help shape the ontological architecture. We outline the properties below and consider their relevance to the three relations.

These three relations normally manifest as hierarchies; in other words, they have the structure of a partially ordered set (or in the case of type-instance, it's cover relation, as it is not transitive). They are standardly represented in an obvious way in Hasse diagrams (sometimes known as upward diagrams) as a directed acyclic graph – nodes connected by arrows that have no cycles – see Figures 1 to 6 below. We use these diagrams to show the formal structures we are examining. The relations have a conventional direction, given in Table 2.

Table 2 – Conventional directions

relation	upwards direction
whole-part	part-to-whole
type-instance	instance-to-type
super-sub-type	subtype-to-supertype

There are some TLOs – such as Entity-Attribute-Relation (the original Chen version) – where there are no super-sub-type relations. This is often found in the physical implementation – SQL being a clear example. Further, it is often the case that whole-part relations are not explicitly marked, in other words, separated out from other relations.

Typically, TLOs will have a range of choices on how they constrain the three hierarchies – 4.2.1.1 to 4.2.1.8 identify the relevant choices which we use below to analyse the TLOs. Often, as noted earlier, the underlying question is whether these constraints breach an explanatory sufficiency (plenitude) principle and so unnecessarily limit expressiveness. As always, there is often various factors in play, so the decision is not clear cut.

4.2.1.1 Parent-child-arity

In hierarchies, the number of parents a node can have is the parent-arity, the number of children the child-arity (note that the number of parents may differ from the number of ancestors). In some TLOs some of the relations have their parent-arity or child-arity limited to one. This changes the structure from lattice-like to tree-like (see Figure 2).

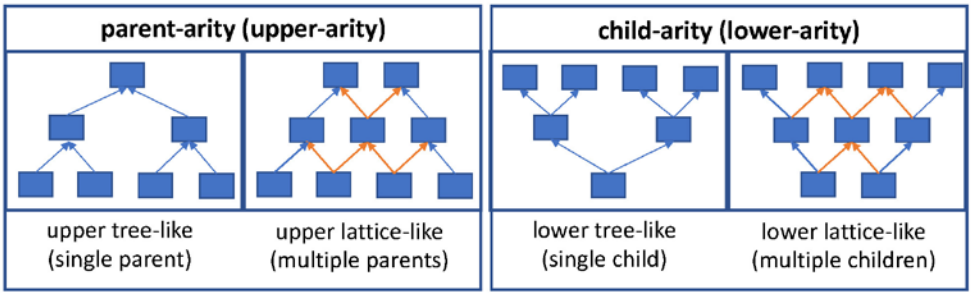


Figure 2 – Parent-child-arity structures in Hasse diagram format

The way these constraints are typically applied to the three relations is outlined in Table 3; which shows that the relevant choices are for type-instance and super-sub-type. In the object-oriented modelling community, these are known respectively as single or multiple classification and single or multiple inheritance.

Table 3 – General parent-child-arity

relation	**general direction-arity**	
	general parent-arity	general child-arity
whole-part	always unconstrained	always unconstrained
type-instance	single or unconstrained	always unconstrained
super-sub-type	single or unconstrained	always unconstrained

Constraining the hierarchy to a single parent is prima facie parsimonious. However, it also seems prima facie explanatorily insufficient. Why should a type such as mare not have female and horse as its supertypes? Why should an individual such as Donald Trump not be an instance of the types 'human being' and 'biologically male'? These choices for single parent structures look likely to be less than optimal unless there are other factors counting in their favour.

4.2.1.2 Super-sub-type – transitivity

We focus here on the transitivity of the super-sub-type relation. Type-instance is generally considered not to be transitive and whole-part to be transitive.

The super-sub-type relation can be found in the early logic, in, for example, Aristotle's syllogisms – in the assertion that 'Every S is P' (Every Human is an Animal). This can be translated into 'S is a sub-type of P' (Human is a sub-type of Animal). Its explicit recognition as a relation came with the nineteenth century mathematization of logic by Boole and others. Implicit visualisations of it as containment appear in Euler and Venn circles. However, it is more commonly visualised now as a hierarchy diagram – where the links represent instances of the sub-type relation. The majority of the graphic representations of the TLOs (in Appendix F) are super-sub-type hierarchy diagrams.

However, some care needs to be taken when interpreting these diagrams – as they only show the cover relation (parents and children with no ancestors or descendants). The traditional semantics of super-sub-type is transitive: if every B is A and every C is B, then it seems clear that every C is A. So, ancestors or descendants are automatically included (though not shown in the diagrams).

In some TLOs, their version of super-sub-type is not transitive – ancestors or descendants are not automatically included. Typically, only the links shown in the hierarchy are deemed to exist. The UML TLO is an example of this, which is most likely driven by implementation rather than semantic concerns. This choice should be made explicit.

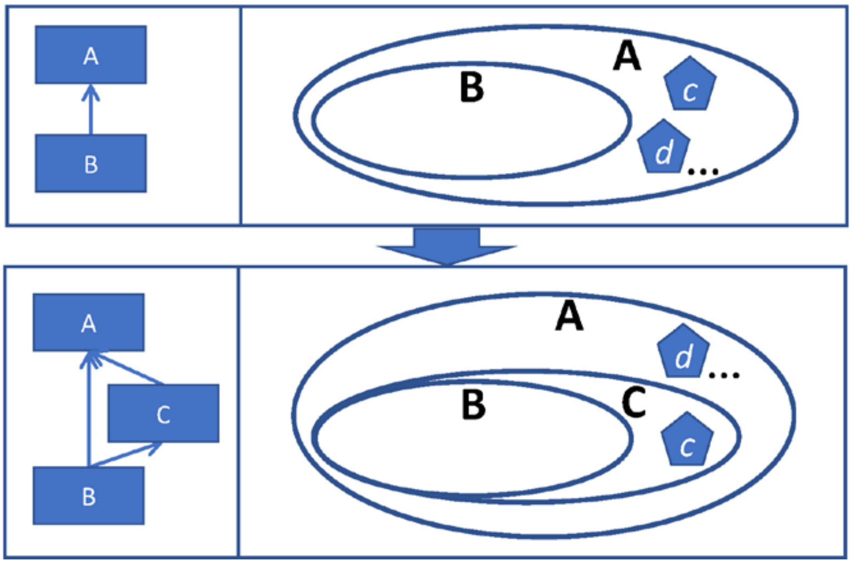


Figure 3 – Intervening subtypes

A similar kind of interpretation situation occurs when the super-sub-type hierarchy diagram only shows the relevant types. This is commonplace where the TLO supports extensional types. In this case, where a super-sub-type hierarchy diagram shows B as a sub-type of A, one cannot automatically infer that B is a child sub-type of A – as there may be intervening sub-types. An extensional TLO allows any collection of objects to be a type. If there is more than one instance of *A* that is not an instance of B – for example *c* and *d* in Figure 3, then there is a type C which has the members of B plus *c* as its members. C is not identical to either A or B as it has c but does not have d as a member. C is however a super-type of B and a sub-type of A This hopefully illustrates how ontological commitments impact upon the interpretation of these diagrams. This choice is dealt with under the formal generation section.

4.2.1.3 Boundedness

Hierarchies as ordered relations might or might not have a top or bottom. As shown in Figure 2, the hierarchy is bounded if all of the maximal paths terminate, and unbounded if any maximal path does not terminate, though, as the diagram also shows, some may. The hierarchy is upwards bounded if all the maximal paths terminate upwards, and the set of terminating nodes are the top elements of the hierarchy. The hierarchy is downwards bounded if all the maximal paths terminate downwards, and the set of terminating nodes are the bottom elements of the hierarchy.

These four options permute into four possible configurations. One of them, upwards and downwards bounded, opens up the possibility of further constraining the hierarchy to a finite number of levels.

The interesting hierarchical relation for us, in the top-level ontologies we have reviewed, is type-instances. The first interesting case for us is firstly, whether type-instances is downwards bounded – the left-most case in Figure 4.

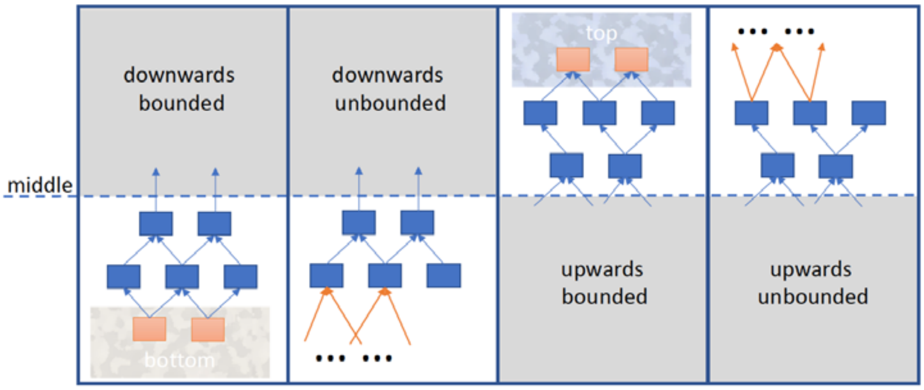


Figure 4 – Possible boundedness options in Hasse diagram format

Type-instance downwards boundedness is associated with the universals-particulars division that goes back to the Ancient Greek Aristotle, and beyond; where universals have instances, but particulars do not (another way of defining bottom). All seriously ontologically committed top-level ontologies make this division. Some of the generic ontologies have meta-models that place no constraints on the hierarchy. An example would be OWL's use of punning; it does not identify a bottom level, so it is always possible to extend downwards. The Topic Map Reference Model is similarly unconstrained.

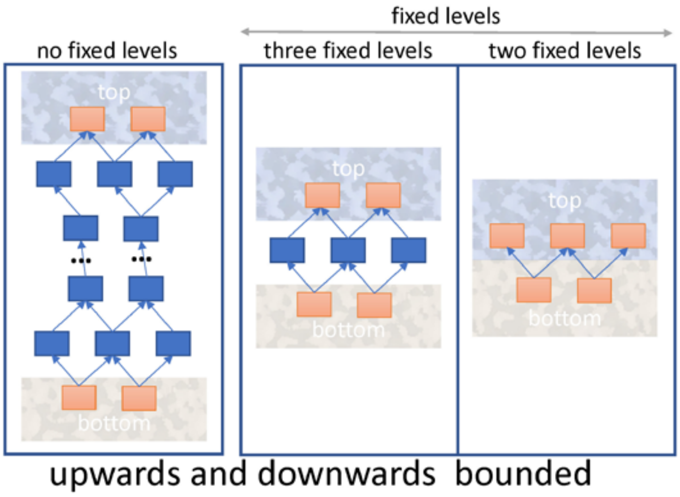


Figure 5 – Fixed level boundedness

A pragmatic argument for this lack of constraint is that it is too onerous to build the bound into the model at design time, and that it is more useful to let the users at runtime decide on whether or not to extend the hierarchy down a level. This then places the onus on the users to ensure the quality of the boundary, to ensure, for example, that something that is clearly an individual, such as Donald Trump, has no instances.

Then if type-instance is also, upwards bounded – the left-most case in Figure 5; there is a choice as to whether it is finitely bounded to a fixed number of levels – and if so, to how many levels. If TLOs are so constrained, they are often fixed to either three or two levels. Figure 5 has examples of two and three levels. OMG's Meta Object Facility (MOF) is a case whether there are four.

The way these constraints are typically applied to the type-instance relations is outlined in Table 4; the boundedness choices for the other two relations (whole-part and super-sub-type) are not sufficiently interesting to make it to the framework.

Table 4 – Type-instance boundedness options

relation	downwards bounded	fixed finitely bounded	fixed number of levels
type-instance	can be unbounded or bounded	fixed finitely bounded or not	often two or three, but can be more

4.2.1.4 Intransitive vertical stratification

Type-instance is intransitive – for example, if a is an instance of type b and b is an instance of type c – it does not follow that a is an instance of type c. This allows us to distinguish between cases where a node's descendants could have its ancestors as parents – and where it does not (for transitive relations, it is always the case). This affects the structure and is visible in the hierarchy's Hasse diagram as illustrated by Figure 6. Also, stratified hierarchies can be ranked – also illustrated in Figure 6.

Another way of characterising this is as a distinction between hierarchies where each new rank can only be constructed, or based upon, the components of the previous rank (stratified) – and ones that can be constructed from all earlier ranks (unstratified). Of course, in cases of two levels, the previous rank is all earlier ranks, so it is a limit case of stratified. The standard technical terms for these are stratified and unstratified respectively – so we use them. This vertical stratification is a different sense of stratified from the one used in horizontal stratification; it is worth paying attention to the different senses.

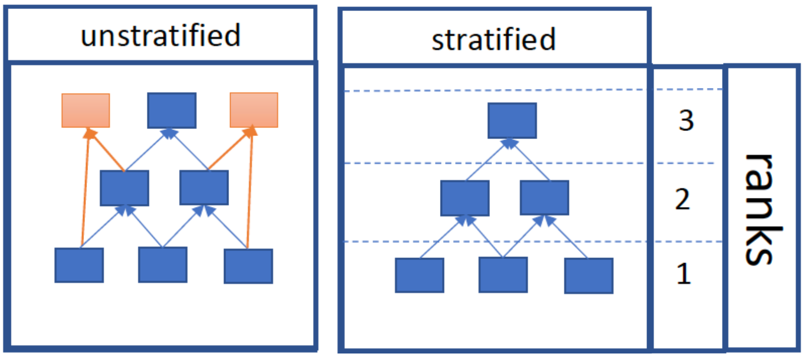


Figure 6 – Ranks: vertically stratified and unstratified

Ontologies that are defined using meta-models and meta-meta-models, such as UML and MOF, are typically stratified. Extensional ontologies, such as those based upon BORO and IDEAS, are usually unstratified. This distinction is much discussed in the mathematics of set and type theory, where sets are unstratified and types stratified.

The stratified approach, by some measures, is less structurally simple as it involves more restrictions. Also, as one can see from the Figure 6, the stratified approach is ontologically parsimonious and the unstratified approach plenitudinous. The key question is which provides more relevant expressiveness. This naturally leads to questions about what motivates the stratification restriction – there does not seem to be a good ontological answer for this.

4.2.1.5 Formal generation

Ontology models are typically built through the careful manual addition of references to objects in the model. However, this is not the only way the structure in the model is created. Some top-level ontologies include algorithms for automatically adding new objects to the model. This is known as formal generation, as there are formal (algorithmic) rules for the generation. If we want a rounded picture of the structure, we need to consider this formal generation as well.

There are a variety of types of algorithm that can be adopted. For our broad-brush picture, we just consider two core cases here: fusion and complement, the first an upwards (parent) generation the second a downwards (child) generation. This is enough to give a measure of the generative approach. Fusion is where one is given two objects of the right kind and then one can infer the existence of their parent fusion. Classic cases are mereological fusion and the pairing axiom in set theory – a whole that consists exactly of two or more particulars. Complement is where, when one is given a parent and one of its children, one can infer the existence of another child that is the rest of the parent. As Figure 7 shows, the formal generation produces both the object and its hierarchical relation(s).

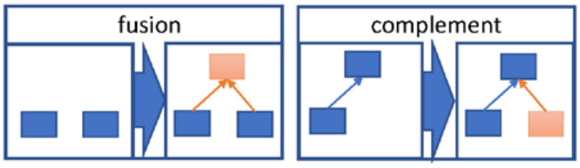


Figure 7 – Two kinds of formal generation

As Table 5 – formally generative options shows, there are choices for both these modes of generation for all three relations except for type-instance and complement. To see why type-instance is an exception, consider a singleton type A = {a}. It has the single instance *a*, but there is no complement of *a*.

Table 5 – formally generative options

relations	formally generative	
	fusion	**complement**
whole-part	yes or no	yes or no
type-instance	yes or no	typically, no
super-sub-type	yes or no	yes or no

Adopting formal generation for each of the three relations can be seen as examples of plenitude; all possible applications of the rule are automatically allowed. But this raises questions of whether there is overgeneration (see discussion on Basis in 3.2.1 above). Could these be examples of profligacy and promiscuity? Given the inter-related nature of TLOs, this assessment needs to be done in the context of all the choices made by the TLO. However, a prima facie case for them can be made on the basis of their close association with the two standard examples of ontological economy, classical mereology and set theory.

4.2.1.6 Relation class-ness

The concept of first- and second-class objects was introduced by Christopher Strachey in the 1960s (Strachey, 2000). A second-class object is one that is not given the same 'rights' as other objects, a first-class object has the same rights as other objects. The particular right we are considering here is whether an object can be an instance of a type. If the three core relations are first-class objects, then they can be. Though the details differ slightly for the three relations, in each case, this allows for a (type-instance) link in our Hasse diagrams that starts with a link – as shown in Figure 8. The resultant graph structure is known as a hyper-graph.

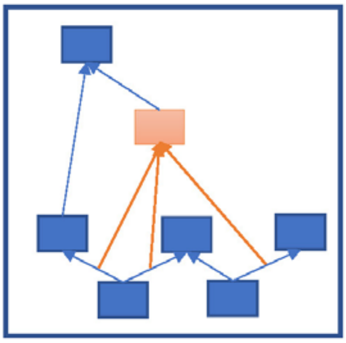


Figure 8 – First-class relations Hasse diagram

Whole-part relations are usually first-class. Type-instance and super-sub-type are often second-class but can be first-class

Table 6 – Relation class

relation	class
whole-part	usually first
type-instance	often second, but can be first
super-sub-type	often second, but can be first

Imposing second-class-ness on these relations is a restriction on plenitude – as it introduces a block on the existence of possible objects. Hence, recognising all three relations as first-class would be an example of plenitude; all possible applications of the type-building rule are automatically allowed. The question then arises whether this plenitude veers into profligacy and promiscuity. It turns out that the standard pattern for classification, as used by Linnaeus in his taxonomy, needs these first-class objects (see *Formalization of the classification pattern* (Partridge, 2016) and Business Objects (Partridge, 1996).

4.2.1.7 Vertical structures

These vertical classifications are shown below.

type	relation	characteristic	choice
parent-arity	type-instance		single or unconstrained
parent-arity	super-sub-type		single or unconstrained
boundedness	type-instance	downwards	bounded or unbounded
boundedness	type-instance	fixed finite levels	fixed or not fixed
boundedness	type-instance	number of fixed levels	[a number]
(vertical) stratification	type-instance		stratified or unstratified
formal generation	whole-part	fusion	yes or no
formal generation	whole-part	complement	yes or no
formal generation	type-instance	fusion	yes or no
formal generation	super-sub-type	fusion	yes or no
formal generation	super-sub-type	complement	yes or no
relation class-ness	type-instance		first- or second-class
relation class-ness	super-sub-type		first- or second-class

Figure 9 provides a visual summary of this.

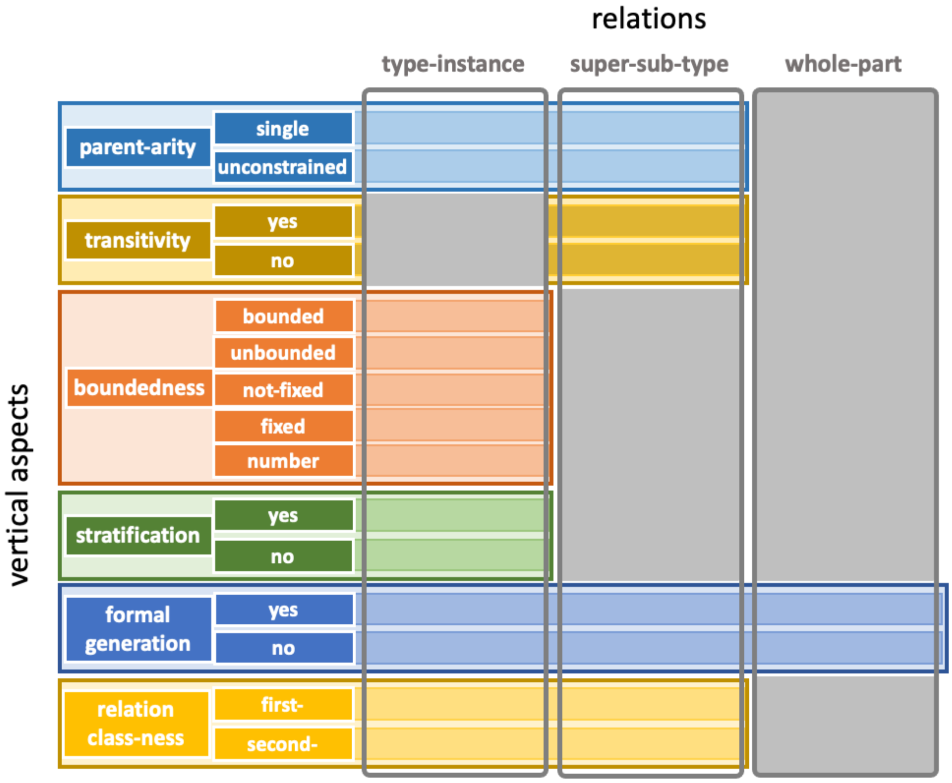


Figure 9 – Visual summary of the vertical aspects

4.2.1.8 Other vertical structures

There are a number of other vertical structures that have not been included as they are less relevant. These include:

- connectedness
- restricted single type-instance parent-arity – single classification.

Connectiveness

It is not necessarily the case that any two nodes in the graph are connected. If some nodes are not connected, then the graph is disconnected. In this case, the disconnected graph can be divided into connected graphs. See the section below on possibilia for an application. While the connectedness of the structure is important, there is no overall pattern in these core relations that allows us to broadly characterise the owning TLO.

Single classification

There are some groups of types where one would expect the instances to belong to only one type – in other words, the types partition their instances. Quantities are an example. We would not expect something to have two masses, to both weigh 5 kg and 10 kg – it weighs one or the other. Similarly, for qualities, we would not expect an object to be both coloured and transparent at the same time, it has to be one or the other. This kind of restriction is common in TLOs, but again there is no overall pattern that allows us to broadly characterise the owning TLO.

4.2.2 Horizontal aspects: stratification versus unification

There is a group of fundamental choices that impact the ontological architecture which involves whether or not to make a distinction. If one chooses not to make the distinction, one only introduces a single type. If one chooses to make the distinction, one introduces two types; one for each alternative. The choice boils down to whether to horizontally stratify or unify. One can describe choosing to make the distinction as 'separating one potentially unified type into two', creating a horizontal stratification in the hierarchy – and not making the distinction, 'unifying the potentially separated two types into one'.

These choices are perhaps best explained by looking at the specific cases (see 4.2.2.1 to 4.2.2.8). We only consider the major cases relevant to our review of the TLO candidates. We focused on identifying the formal choice – whether to horizontally stratify or unify – and leave the other aspects driving the decision to the more detailed description later in the report (see 4.3). We have mostly described the choices from a unifying perspective, they could equally well have been described from a stratifying perspective.

4.2.2.1 Spacetime

We start with one familiar from 20th century physics. Prior to then, it was assumed that the spatial geometry of the universe was independent of one-dimensional time. There were two related but independent types, spatial regions (regions of space) and temporal regions (regions of time). The work of Einstein and Minkowski introduced the idea – which became accepted – that space and time could be fused into spacetime. Ontologically, this can be seen as unifying spatial regions and temporal regions into spatio-temporal regions whose instances are regions of spacetime. Figure 10 provides some examples from the TLOs. BFO is an interesting case as it hyper-separates, it separates but keeps the unifying type. This raises interesting ontological accounting questions about whether this is overgeneration (as so profligate) or interesting plenitude.

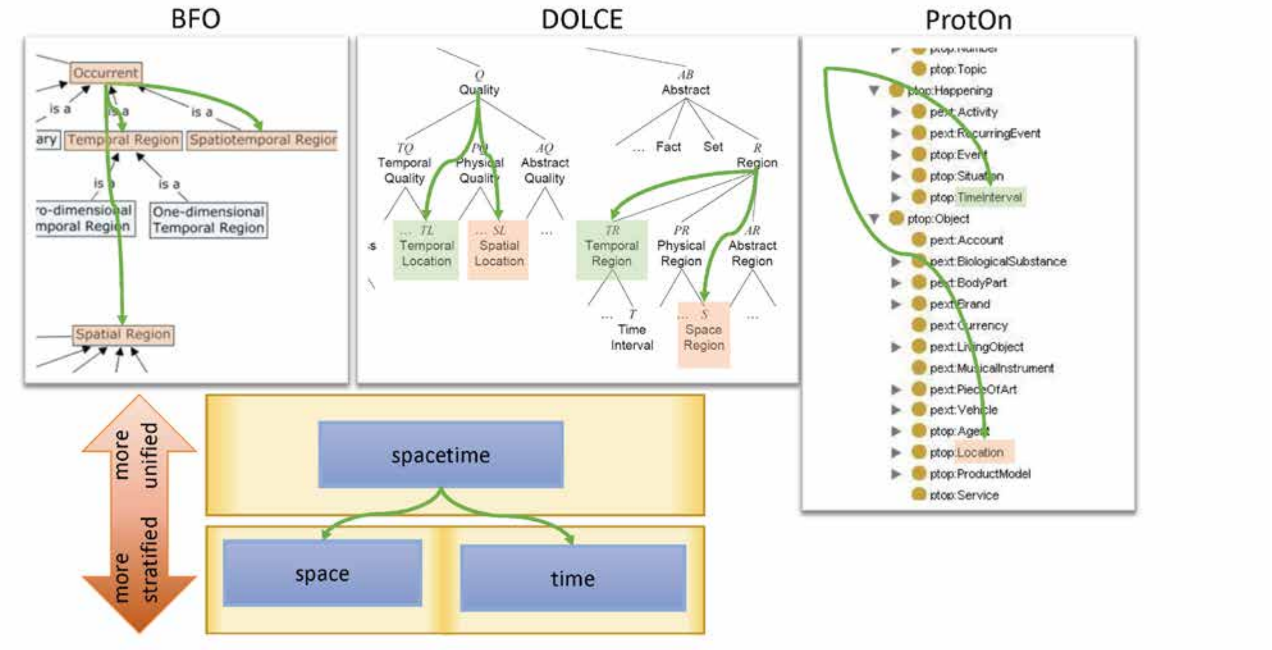


Figure 10 – Separating spacetime – TLO examples

While space, time and spacetime may appear to be familiar notions for interpreting data, it turns out to be a tricky area to tie down formally. It takes some study to develop a clear idea of what the spacetime stratification choice here implies.

We can illustrate the choice simply as between a 1D time plus 3D space or a 4D spacetime – as shown in Figure 11 (based upon Figure 1 in (Gilmore, 2016))

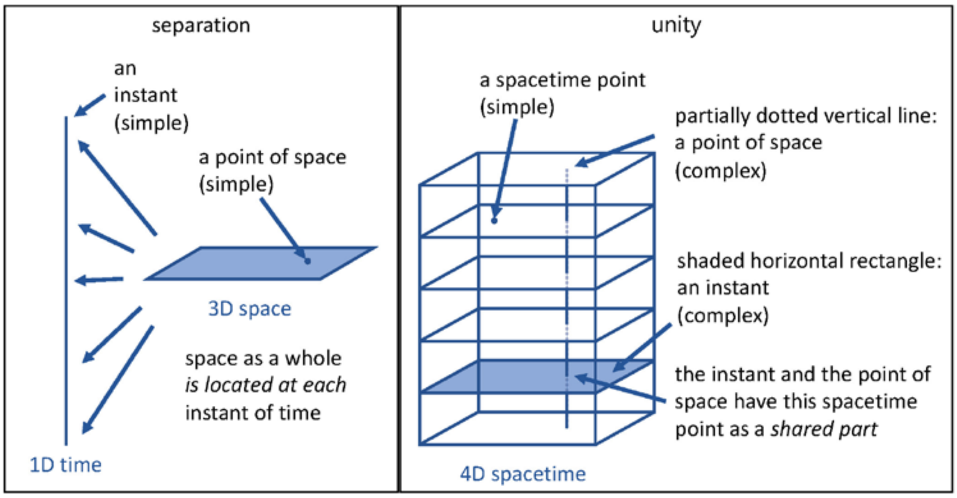


Figure 11 – Separation and unity of instances

In many cases, as here, the stratification (or unification) of the types also implies a separation or unity of their instances. To appreciate the consequences of the choices, Figure 11 shows how three (simple) instances of locations end up under the two regimes – this is recapitulated in Table 7.

Table 7 – Three locations

location	1D time plus 3D space	4D spacetime
instant (in time)	a (simple) point on the 1D timeline	a (complex) horizontal slice in spacetime
a point of space	a (simple) point in 3D space	a (complex) vertical line in spacetime
a spacetime point	A point in 3D space located at a point in 1D time	a (simple) point in 4D space

An under-appreciated consequence of the separation is that 3D space is multiply located – it is located as a whole at each instant of 1D time.

Typically, a TLO will choose to stratify or unify the types and so separate or unify the instances. However, it can attempt to do both. As noted earlier, the TLO BFO provides us with an example. It opts to include both spacetime as well as space and time. This raises interesting semantic redundancies analogous to data redundancy. And so, a requirement that the spaces and times need to be coordinated with spacetimes – one can regard this as a kind of semantic or ontological denormalization analogous to database denormalization.

4.2.2.2 Locations

People often talk of physical objects and their locations, where physical objects occupy their locations, suggesting two related types; objects and locations (let's leave the decision whether the location is spatial, temporal or spatiotemporal to the previous choice). For example, "today your car is parked in the same place as mine was yesterday" could be regarded as a location which was occupied by my car (a physical object) yesterday and your car today. There is a debate going back to Newton and Leibnitz in the 17th century as to whether location is absolute or relative. If it is relative, then location is clearly fundamentally different from physical objects – which aren't. However, if it is absolute, a kind of substance, then this opens the possibility that one could unify objects and their locations as fundamentally the same, technically known as *supersubstantivalism*. If one does not have cases of interpenetration (see 4.3.3) then this resolves the oddity where physical objects exactly occupy a single location throughout their life – unifying eliminates this double-counting. If there is interpenetration, then two objects may collapse to the same location – which may have unintended consequences. After the unification, the physical object and its location, two kinds of substance, are replaced by a single supersubstantival object. One has a broad choice between separating or unifying physical objects and locations.

4.2.2.3 Properties

In language, there is a distinction between nouns and adjectives; between rose and red. This has been taken as an indication of a more fundamental distinction, between what are known as substances and the properties or qualities they bear, for example, where a red rose would have a rose substance that bears a red property/quality. However, in other contexts, such as a Venn or Euler diagrams, we would be happy to have overlapping circles for roses and red objects – with a single icon for each red rose in the overlap. This implies there are instances of the type object that belonged to both the lower level types rose and red. So, here is a choice between stratifying to substances as the bearers of properties or unifying to objects.

4.2.2.4 Endurants

Philosophers have noted that people say some types of object, such as stones and chairs, exist; whereas other types, events, occur or happen or take place. It is suggested that this marks a fundamental distinction between continuants and occurrents – where occurrents (that occur) are the events that happen to continuants (that exist). There is a competing view that there is no fundamental difference between the two, rather a different perspective on the same object – this has been labelled perdurantist. A classic example (for perdurantists) is glaciers. From a day to day perspective they are solid, unmoving material objects – they exist and so could be classified as continuants. From a geological perspective, glaciers flow – they are events like the flowing of a river – so could be classified as occurrents. A common continuant/occurrent stance, is that there are two glaciers; the existing continuant and the flowing occurrent. A perdurantist stance would be there is a single perdurants object which can be looked at from two perspectives.

4.2.2.5 Immaterial

Philosophers have suggested that the hole inside a doughnut is different from the doughnut. The doughnut is composed of stuff whereas the hole is not composed anything – it is defined by the doughnut. They suggest making a stratification where the doughnut is a material object and its hole is an immaterial object dependent upon the material doughnut. A unifying stance would not regard this distinction as fundamental and not recognise material and immaterial are fundamental types in its ontology. The hole has a spatial extent that contains matter, though this matter may well change over time. In a sense it is 'immaterial' what matter is in the hole, but this does not, by itself, make it either immaterial or a fundamentally different kind of object.

4.2.2.6 Summary

Table 8 lists these specific cases, and the stratifying relations that typically relate the separated objects. All of these choices are present in some of the TLOs we have surveyed. The list is not intended to be complete but gives an indicative picture of the range of stratifications. For example, there is a further major choice, whether to make a distinction between form and matter (often known as hylomorphism – for more details see "Form vs. Matter" (Ainsworth, 2020)). Even though this is currently an active area of research, we have found no examples of this among the candidate TLOs. So, we merely note the choice here and omit it from the list.

Table 8 – Summary of the horizontal stratification choices introduced

Label	Unified type	Separate types	Stratifying relation
spacetime	spatio-temporal objects	spatial objects, temporal objects	spaces are multiply located at times (though this is often a derived relation – from an occupying object's links to both space and time)
locations	supersubstantival objects	(physical) objects, locations	objects are (exactly) located at their locations
properties	objects	substances, properties	substances are bearers of properties
endurants	perdurants	continuants, occurrents	occurrent is dependent upon continuant
immaterial	(physical) objects	material objects, immaterial objects	immaterial objects are part of material objects

4.2.2.7 Stratification journey

There is limited inter-dependence between the choices meaning that a range of permutations are possible. For a top-level ontology, one can visualise the architectural stratification choices being adopted in a sequence, starting with no stratifications and introducing the choices one or two at a time – as illustrated in the figures below. This sequence or journey is a rational reconstruction – the original development of the top-level ontology is most likely ad hoc and bottom up. However, this reconstruction gives us a good picture of the underlying architecture.

Figure 12 and Figure 13 show a four stage example top-level ontology stratification journey (or sequence) for BFO, which includes multiple stratifications– where five choices result in six strata. Figure 14, shows a journey with no choices resulting a single stratum based upon the IDEAS TLO. Finally, Figure 15 is the legend for these stratification journey diagrams.

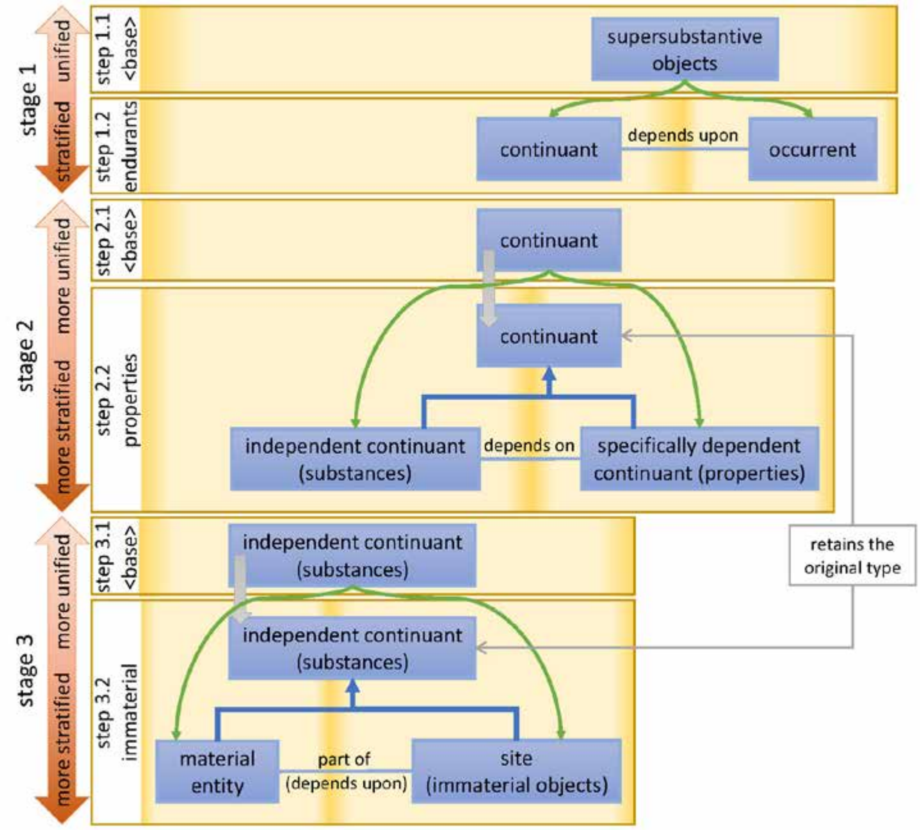


Figure 12 – An example top-level ontology stratification journey – BFO stages 1 to 3

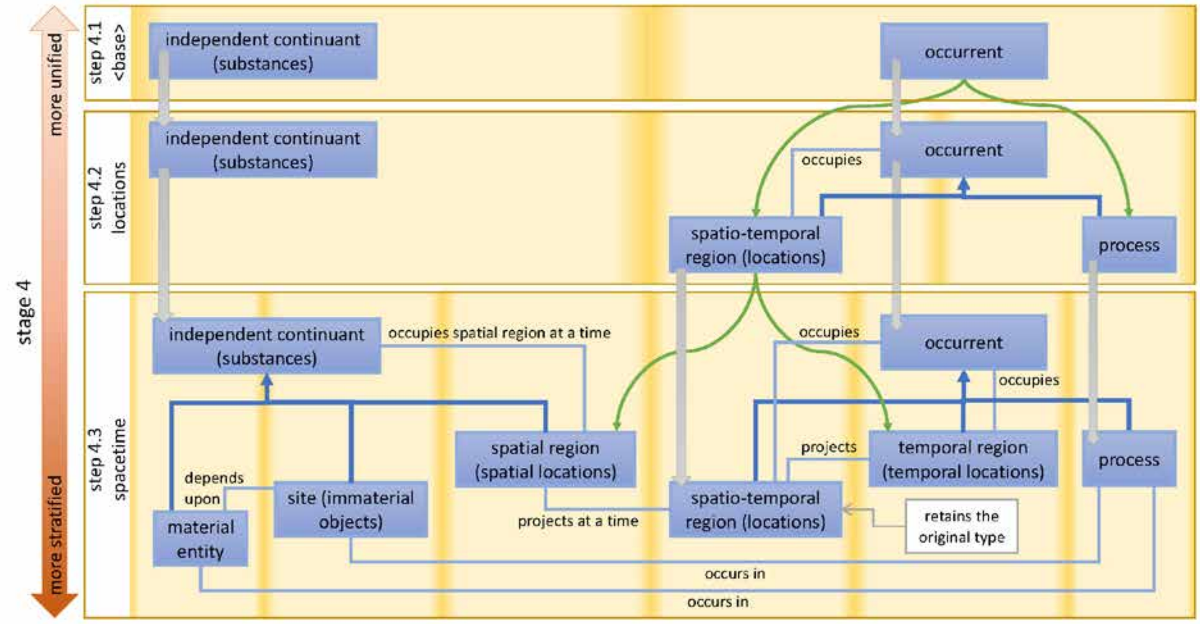


Figure 13 – An example top-level ontology stratification journey – BFO stage 4

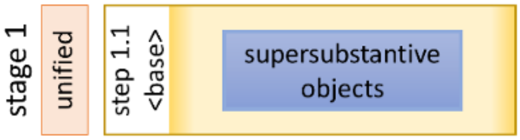


Figure 14 – An example top-level ontology stratification journey with no stratification - IDEAS

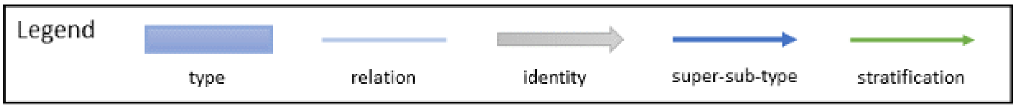


Figure 15 – Legend for stratification journey diagrams

As these examples show, choosing to make the distinction stratifies the architecture – choosing to not make it, correspondingly, unifies it. From the perspective of fundamental parsimony, making the distinction is always multiplicative, simpliciter – in the sense that it results in two fundamental types rather than one. It is also, in many cases, even more multiplicative in that it introduces a new type of fundamental relation between the separated types. There can be a variety of reasons for making the distinction. The interesting question for us is whether the cost accounting shows this worthwhile. It is often difficult to do this for the individual stratifications as useful derived objects – including fruitfulness ones – often emerge from the way the choices interact.

4.2.2.8 Other divisions

A common division, often mentioned, is the division into universals and particulars. However, unlike the stratification choices we have looked at above, this is not really a stratification. As we noted when looking at boundedness above, this is better seen as a choice as to whether the type-instance relation is downwards bounded.

4.3 Universal commitments

The section focuses on the details of the commitments themselves. It focuses on universal commitments; ones that one would expect to be exercised in all (or almost all) domains. Some of these commitments have appeared in the previous section – as they impact directly upon the ontological architecture of the hierarchies. We deal with the details of these first (see 4.3.1 and 4.3.2).

4.3.1 Type-instance

It is relatively easy to find examples of this relation. The cat called Holly is an instance of the type cat. My hand is an instance of the type hand. However, it has a variety of potential ontological explanations depending upon the details of the ontological commitment. The vertical and horizontal aspects help to characterise what has been chosen, but heavyweight TLOs will typically fill in more detail. This can vary substantially – as the following simplified examples show. One form of explanation is that types are the collection of their instances – often labelled extensional. In this case, the type-instance relation boils down to being a member of that collection. Another, form of explanation is that the instance exemplifies the type and so is in a way partially identical to it. So, Holly exemplifies cat-ness, in the sense that she is partially identical to it. Similarly, my arm exemplifies arm-ness. This topic is subsumed under the more general topic criteria of identity and dealt with in 4.3.6.

4.3.2 Whole-part – mereology

Mereology (from the Greek μερος, 'part') is the theory of the whole-part relation. This relation is typically transitive, where if A is a part of B and B is a part of C then A is a part of C. This relation has been extensively formalised where a number of decomposition principles have been identified of various strengths from which one can build the overall theory. These start with the weakest, core mereology (CM) though minimal mereology (MM) and extensional mereology (EM) to general extensional mereology (GEM) (The logical formulation of these can be found in (Varzi, 2019)). If a TLO makes its mereology explicit, one would expect it to adopt a selection of the identified principles, typically one of these variants above. Indeed, a number of the TLOs adopt GEM.

Mereology is closely related to topology, in the sense of the ways in which objects overlap and connect. This is why mereology is often extended to mereotopology. This has also been extensively formalised giving rise to a range of theories, with the strongest theory being General Extensional Mereotopology with Closure conditions (GEMTC). A detailed description of this can be found in (for example) Chapter 4 of Casati and Varzi's *Parts and places: the structures of spatial representation* (Casati, 1999). Again, if a TLO makes its mereotopology explicit, one would expect it to adopt a selection of the identified principles – or be able to provide its own.

4.3.3 Interpenetration: location and mereology (supersubstantivalism+)

The separation of location and objects provides a simpler background for explaining how mereology and interpenetration interact. Consider the standard example of a statue and the clay it is made of. Under one view, the statue and the clay are different objects and they share no parts. However, they do share a location. We typically introduce a direct relation to capture this, saying the statue is composed of clay. We also assume that the statue and the clay have separate parts and parthood relations, but that the parts of the statue align with the parts of the clay. So, if an arm (a part of the statue) breaks off, the corresponding part of the clay does too. There is some kind of mereological divorce and associated harmony or coordination. There are other kinds of similar cases. Consider a road that marks an administrative boundary. We could say the road and boundary share a location, but do not overlap – they share no common parts. In general, we say that two objects interpenetrate when they do not share parts, but their locations do – and the two examples above are cases of interpenetration.

We are faced with here a choice about how we want to deal with these cases. We can adopt a position, known as supersubstantivalism+, which does not allow the mereological divorce and coordination needed for interpenetration. Instead, it assumes there is no divorce, the objects share parts. At the times when the clay composes the statue, that part of the clay is part of the statue. At the times when the road marks the administrative boundary, they overlap, sharing common parts.

At a general level, supersubstantivalism+ is prima facie more qualitatively parsimonious as there is no need for the occupies relations. It is more quantitively parsimonious as there is no need for the separate parts – and their parthood relation. However, to make this position work it helps if one also has a unifying view of spacetime together with objects and their locations.

4.3.4 Materialism: abstract particulars and non-materialism

In modern thought, there is an attachment to the idea that the world is composed of only material objects that exist in space and time – and so one's ontology is built from these material objects – (roughly) materialism. The idea behind this is that it is difficult to see how objects that exist outside space and time could affect the world and, in particular, could affect our minds such that we could know them. An alternative view is that there are also abstract particulars – individuals that have no existence or dependence on space or time. Numbers are often given as an example of these abstract objects. TLOs often explicitly commit to one or other view.

It is difficult to see how these abstract particulars could have an extension in the normal sense. As we discuss later in 4.3.6, extension can be used as a basis for a view of identity. If one adopted this position, then one will probably favour the materialism choice and avoid the difficulty of developing an extensional criterion of identity for abstract particulars.

4.3.5 Possibilia: actual or possible worlds

Most information processing systems seem to involve possible objects that sometimes are not actual. When we arrange a meeting in the future that eventually does not take place, this is possible but not actual. When we draw up plans for a building that is not built, this is possible but not actual. The TLO needs to provide an account of such objects. In this regard, there is traditionally one major choice to be made. This is whether to limit existence to the actual world or allow it to range over (all) possible worlds.

If one adopts a possible worlds approach, then talk of possible objects becomes talk about objects in possible worlds. The possible meeting that did not happen in this world happened in some other possible world. The planned building that was not built in this actual world was built in some other possible world. If one restricts oneself to the actual world, one needs to develop alternative explanations for these objects. One example, called encoding, suggests talk about possible non-actual objects encodes the object – and that encoding works with a different logic that does not imply actual existence. There are a range of these alternatives in the literature, but none of these are as comprehensive or simple as possible worlds, which is why it almost universally adopted in the TLOs.

4.3.6 Criterion of identity: extensional or intensional

One of the key architectural choices for a top-level ontology is the basis or strategy for its criteria of identity. This establishes a key part of the infrastructure. These criteria determine for the various types of objects whether, in principle, so not necessarily in practice, selected objects are identical or different. There are two broad choices; extensional and intensional (with an 's', not a 't').

Extensional criteria of identity are compositional (constructional); where objects are identical if they are 'constructed' from some external criteria – often involving its components. A classic example of an extensional object is sets, where sets composed (constructed) from the same members/components are identical. Another example is material objects' identity defined in terms of spatiotemporal extension – making their identity dependent upon spatio-temporal regions. These regions then, typically, have identity defined in terms of equivalent (spatio-temporal) parts – assuming some kind of extensional mereology. So, two spatio-temporal regions will differ if one has a part that is not a part of the other.

Intensional criteria of identity aim to capture the meaning or essence of the entity; in practice this is typically represented as a definition. We can construct a classic example of an intensional object using the Euclidean equilateral triangle. Given Euclid's Elements – Book I – Definition 14: "A figure is that which is contained by any boundary or boundaries" and Definition 20: "Of trilateral figures, an equilateral triangle is that which has its three sides equal." An equilateral triangle is then defined as any figure with three sides, where these three sides are equal. This equilateral triangle object referred to by the definition has an extension, the set of figures it applies to – but it is, in principle, possible that other definitions will refer to different objects with the same extension.

One can get to the heart of the distinction as follows. Under an extensional strategy, any object with this set as its extension is equivalent – defining identical objects. However, this is not the case under the intensional strategy; two different intensional objects can have exactly the same extension without being identical. The subject has some key detailed technical aspects which are briefly summarised in Appendix B.

There are clear links here to other choices. Consider the choice between spacetime and space and time. Spatial extension (at a time) is insufficient for identity. There are lots of examples of objects that occupy exactly the same space at a time – at the time of writing, Donald Trump and the President of the United States is an example. However, spacetime is sufficient for identity in these cases – Donald Trump and the President of the United States have different spatio-temporal extents. This makes it natural to choose spacetime over space and time to open up the possibility for extensional identity based upon spacetime. There is a similar situation for types and their extension. In this actual world, there may not be enough verity to capture differences in meaning in the actual extensions. For example, the types renate (having a kidney) and cordate (having a heart) could have exactly the same extension. When we extend extension over possible worlds, then this becomes impossible. If it is possible to have animals with a heart but no kidney, then they will exist on some possible world. Making a choice for possible worlds over the single actual world creates the space for types to have extensional identity.

From an ontological engineering perspective, one can summarise the choice between the two as follows. An extensional strategy allows one to have, in principle, a clear, accurate criteria of identity – the extension. However, these are thin criteria, they do not attempt to capture the characteristics that one might use to identify the object. This is done elsewhere.

An intensional strategy cannot provide clear, accurate principles as there is a difficulty giving these definitions to most common notions (person being an example). However, the definitions given are thick and typically contain the characteristics that enable easy identification.

Prima facie, an extensional strategy is useful when one needs to clearly and unambiguously identify objects. However, this choice comes entangled with other choices, so a case also needs to be made in the context of an integrated set of choices.

4.3.7 Time: presentist versus eternalist (including change)

Natural language is often tensed. We say "Jane was in Glasgow yesterday, she is in London today and she will be in Dover tomorrow" – using past, present and future tenses. We use the tenses in ways that imply existence. So, we say "there are no more dinosaurs" implying they no longer exist. Presentists believes that these tenses in ordinary language have an ontological significance. What is going on now really exists, what happens in the past has existed and what might happen in the future has yet to exist. The eternalists do not, they assume tenses have no real ontological significance.

One way to understand this position is to look at a problem it raises. Presentism prima facie imposes restrictions on cross-time relations; relations where the relata are in different times – so never present at the same time. Being a great-great-grandfather (for example) will typically involve people from different non-overlapping times. Presentism would seem to deny the existence of any of these kinds of relations. There are, as always, ways around this, but this helps to illustrate the effect this choice has on the architecture.

We mention this choice here, as it is a well-recognised choice in philosophy textbooks and also in the computer/applied ontology design. However, all the candidate top-level ontologies that make an explicit choice, choose eternalism. This is understandable from a data perspective. There would be a large processing overhead in having to switch tenses as objects 'move' from future to present to past.

4.3.8 Indexicals: here and now

If the previous choice is eternalism (which it is for all the candidate TLOs that make the choice) then this raises a new issue. How does one recapture the present? This is a key requirement. If one does not know when now is, then one loses track of all sorts of things. One can know the balance of an account at every point in time, but not be able to work out what the current balance is. Being able to represent the present is an obvious requirement for a system. One solution, if one is eternalist, is that one needs to include non-ontological epistemological (or agentological) aspects into the model for the system. For further discussion see Appendix J.

4.3.9 Relations – arity

We often think of relations as being between two objects. Examples are father-of or earlier-than. However, there are relations of higher-order arity (greater than two) – an example of a three-place relation is: X is between Y and Z. This is an example of plenitude, where it is possible to have relations of higher order – where there is no, in principle, bound. Hence, it is worth noting whether TLOs support higher-order arity, and where they restrict themselves to two objects, what the motivation is. Pragmatically, higher-order relations are more difficult to implement, especially if there is not a low finite bound. In practice, one tends not to find relations with orders of more than a single digit – so a pragmatic finite bound may be acceptable when implementing.

The formal nature of relations is an active area of research. So, when doing the more detailed analysis within the framework and when looking in more detail at the TLOs, it may be worth looking at what position is taken on this. In particular, for extensional TLOs it may be worth considering their criterion of identity for relations. We do this in Appendix F.

4.3.10 Choices

These cash out into seven choices in the assessment framework.

Choice	Alternatives
mereology	standard (e.g. GEM) or not
interpenetration	allowed or not allowed
materialism	adopted or not adopted
possibilia	possible worlds or actual world
criteria of identity	intensional or extensional
time	presentist or eternalist
indexicals: here and now	supported – not supported
higher arity	supported – not supported

5 Assessment Framework Results

The framework has three parts; general choices, formal structure and universal ontological commitments as discussed in 4.1, 4.2. and 4.3 respectively. The details of the assessment are shown in Appendix E and the results are presented in 5.1 to 5.4 below.

5.1 General choices

Figure 16 shows the distribution of TLOs across the various choices. Figure 17 shows the percentages for the main choices. It is worth noting there are more ontologically committed than generic TLOs in the candidate list – though this may be partially a result of the selection procedure, as we were not directly looking for generic TLOs. Also, there are slightly more lightweight than heavyweight TLOs.

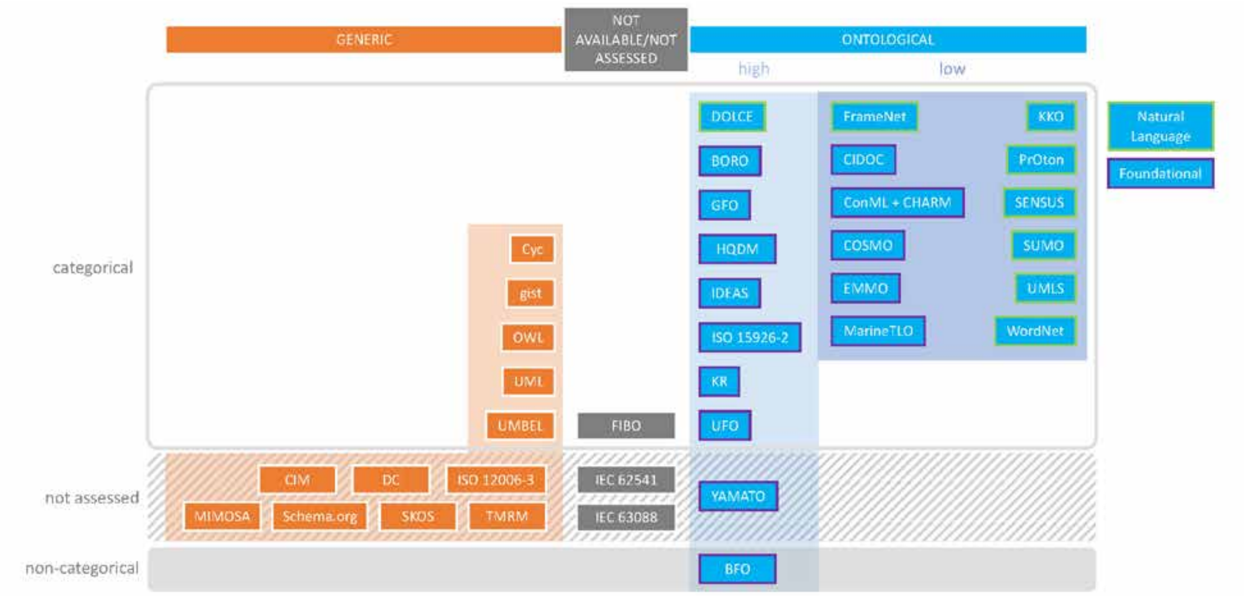


Figure 16 – General choices – framework assessment

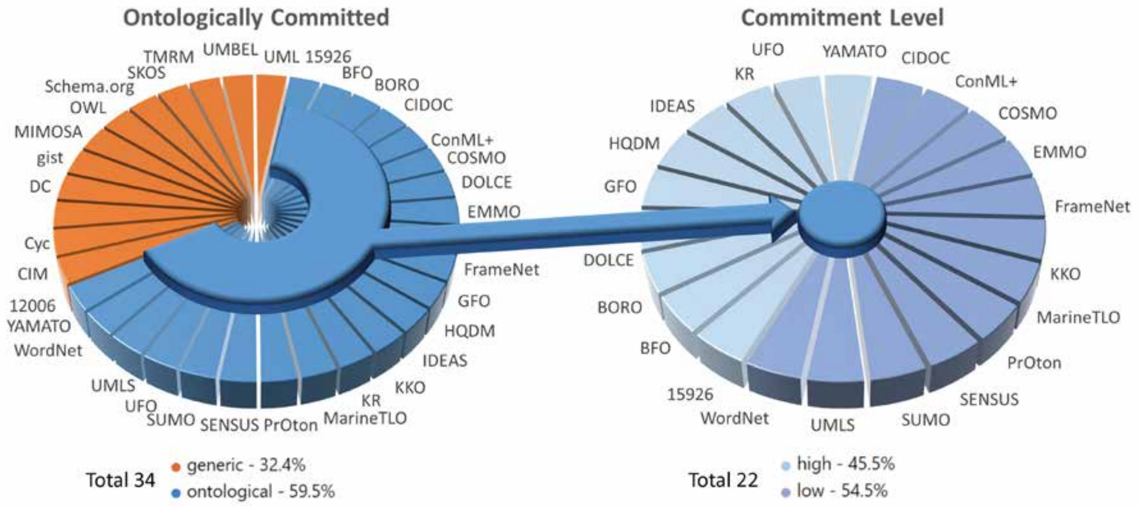
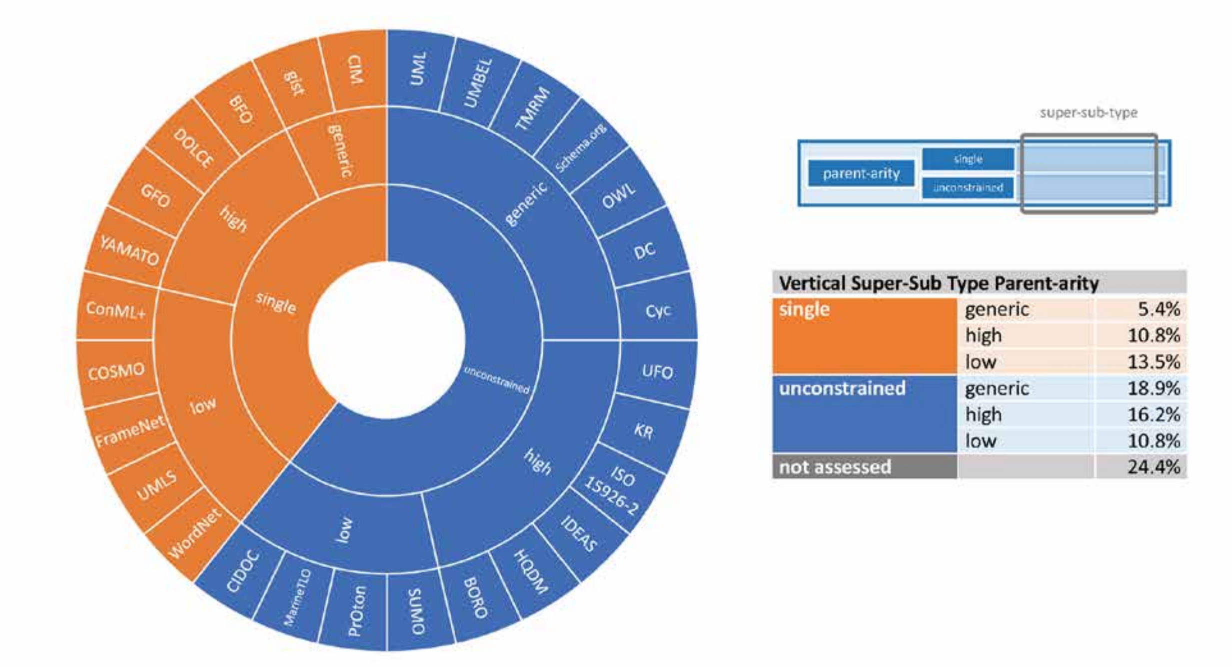


Figure 17 – Main general choices – percentages

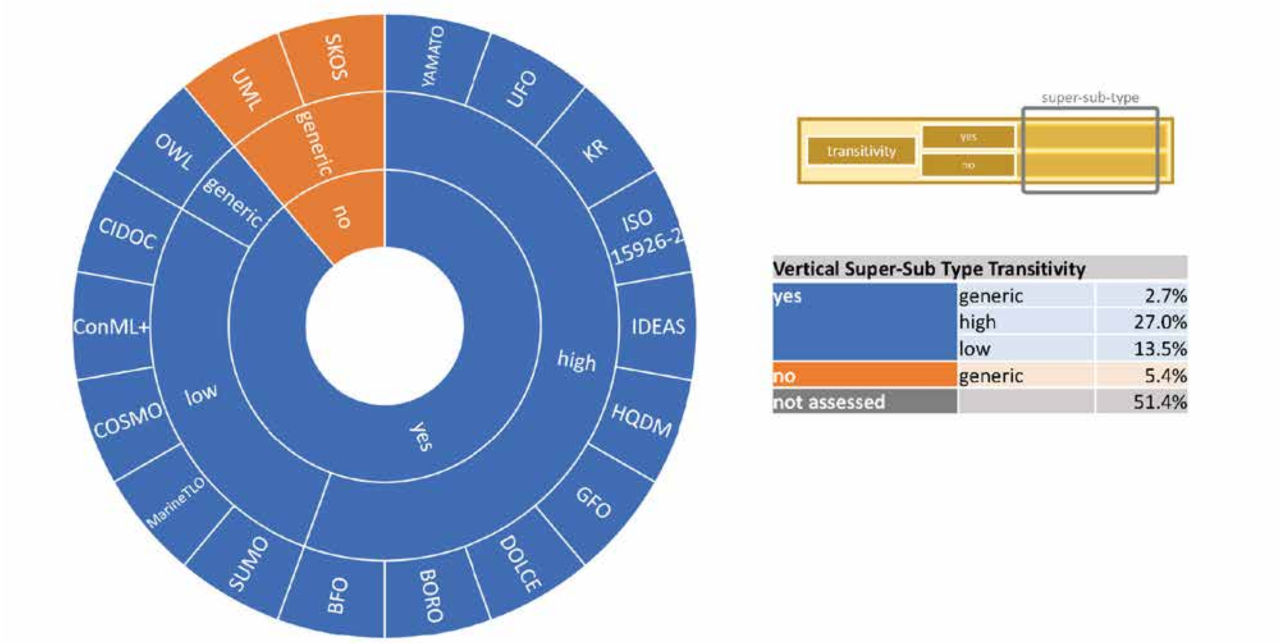
5.2 Formal structure: vertical aspects

The formal structure assessment has two parts; vertical and horizontal. This section deals with the vertical choices. The following figures give the percentages for each choice.



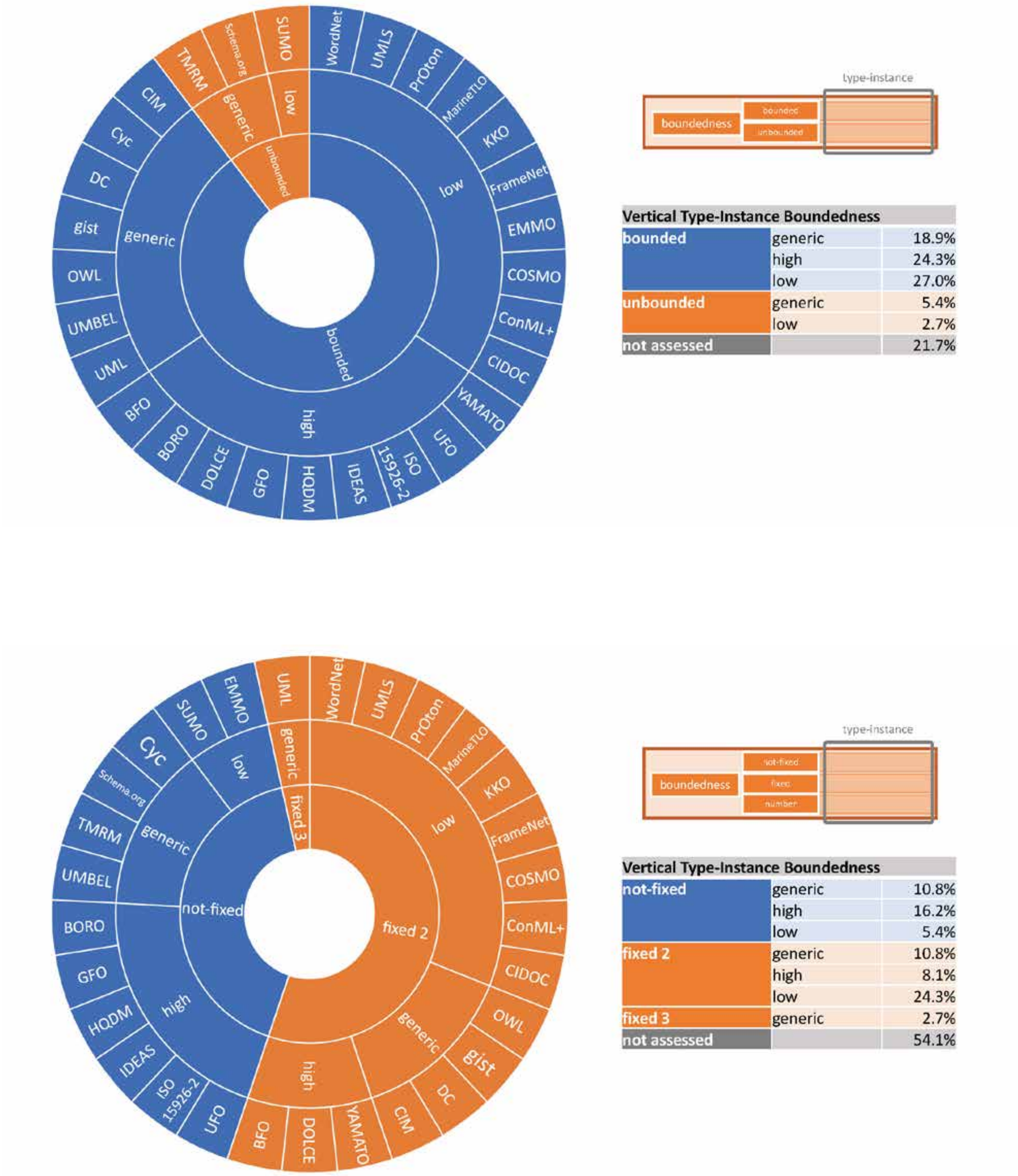
Vertical Super-Sub Type Parent-arity		
single	generic	5.4%
	high	10.8%
	low	13.5%
unconstrained	generic	18.9%
	high	16.2%
	low	10.8%
not assessed		24.4%

Figure 18 – Vertical aspects – commitment level – parent-arity



Vertical Super-Sub Type Transitivity		
yes	generic	2.7%
	high	27.0%
	low	13.5%
no	generic	5.4%
not assessed		51.4%

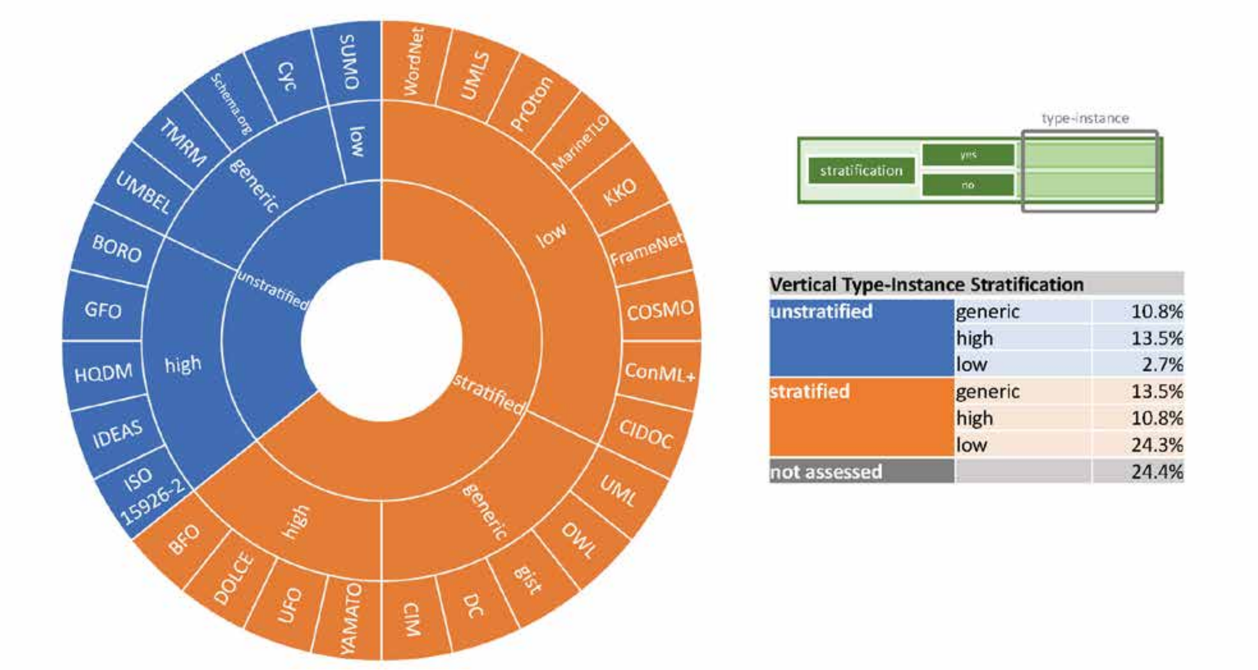
Figure 19 – Vertical aspects – commitment level – transitivity and boundedness



Vertical Type-Instance Boundedness		
bounded	generic	18.9%
	high	24.3%
	low	27.0%
unbounded	generic	5.4%
	low	2.7%
not assessed		21.7%

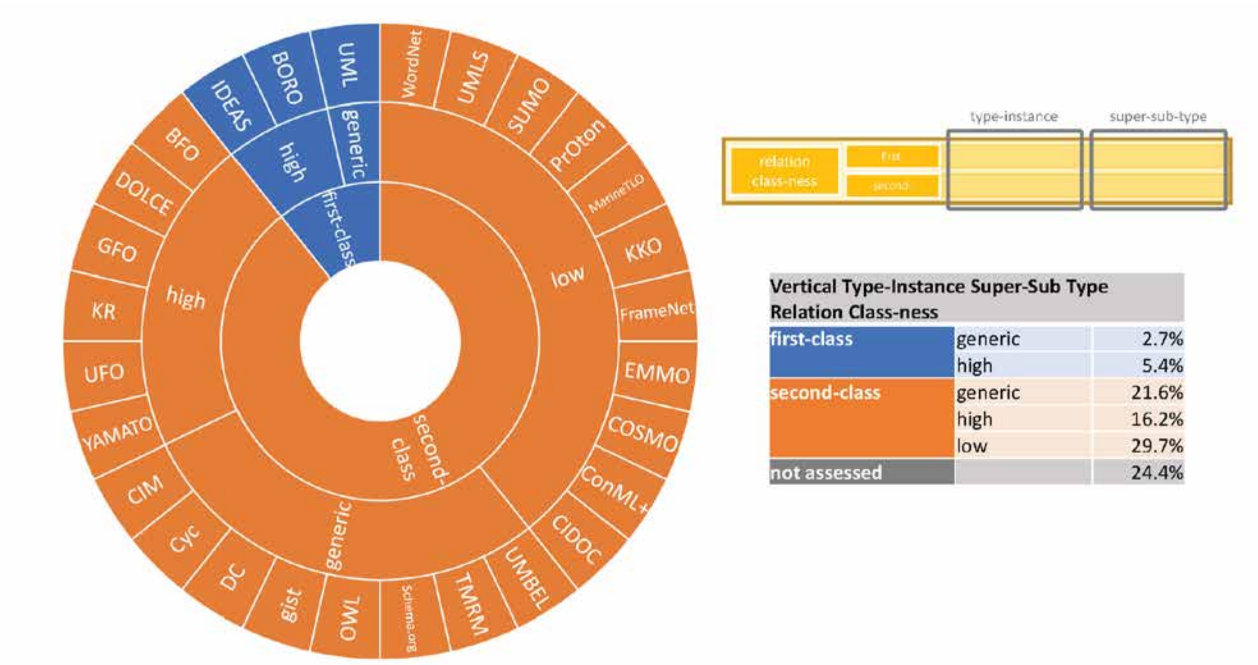
Vertical Type-Instance Boundedness		
not-fixed	generic	10.8%
	high	16.2%
	low	5.4%
fixed 2	generic	10.8%
	high	8.1%
	low	24.3%
fixed 3	generic	2.7%
not assessed		54.1%

Figure 20 – Vertical aspects – commitment level – boundedness



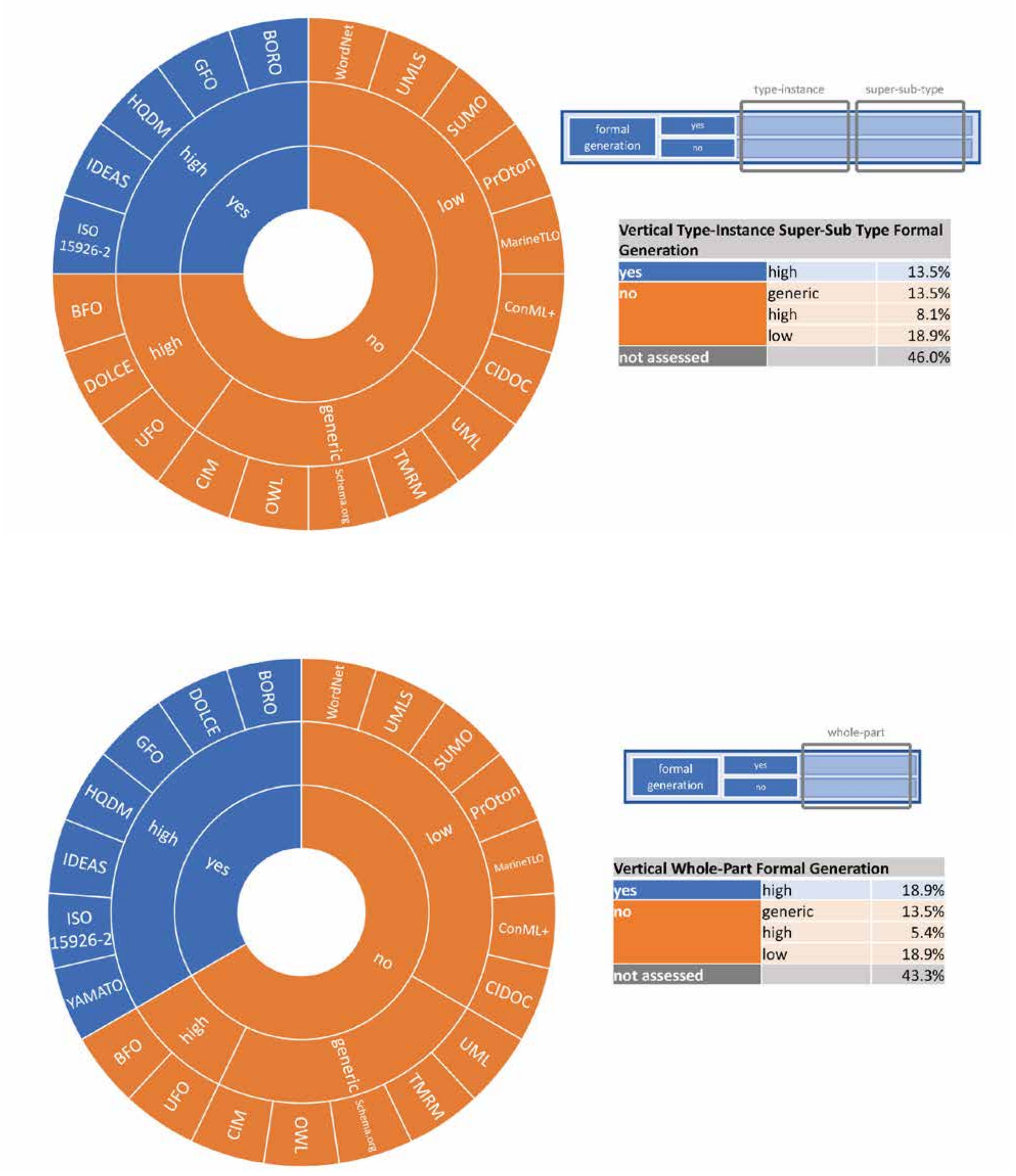
Vertical Type-Instance Stratification		
unstratified	generic	10.8%
	high	13.5%
	low	2.7%
stratified	generic	13.5%
	high	10.8%
	low	24.3%
not assessed		24.4%

Figure 21 – Vertical aspects – commitment level – by type-instance – stratification



Vertical Type-Instance Super-Sub Type Relation Class-ness		
first-class	generic	2.7%
	high	5.4%
second-class	generic	21.6%
	high	16.2%
	low	29.7%
not assessed		24.4%

Figure 22 – Vertical aspects – commitment level – relation class-ness



Vertical Type-Instance Super-Sub Type Formal Generation

yes	high	13.5%
no	generic	13.5%
	high	8.1%
	low	18.9%
not assessed		46.0%

Vertical Whole-Part Formal Generation

yes	high	18.9%
no	generic	13.5%
	high	5.4%
	low	18.9%
not assessed		43.3%

Figure 23 – Vertical aspects – commitment level – formal generation

5.3 Formal structure: horizontal aspects

Figure 24 maps the TLOs against the horizontal aspects – the TLOs cluster at either end of the range of unification-stratification. Figure 25 gives the percentages for each choice.

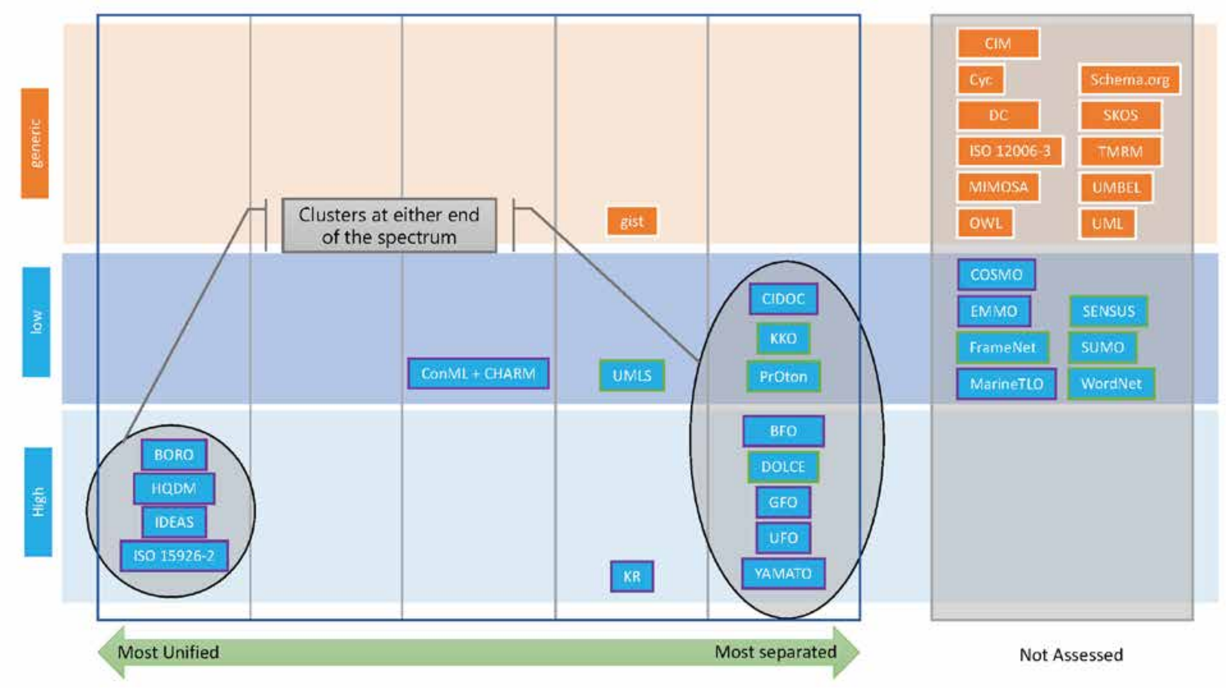


Figure 24 – Horizontal aspects – framework assessment

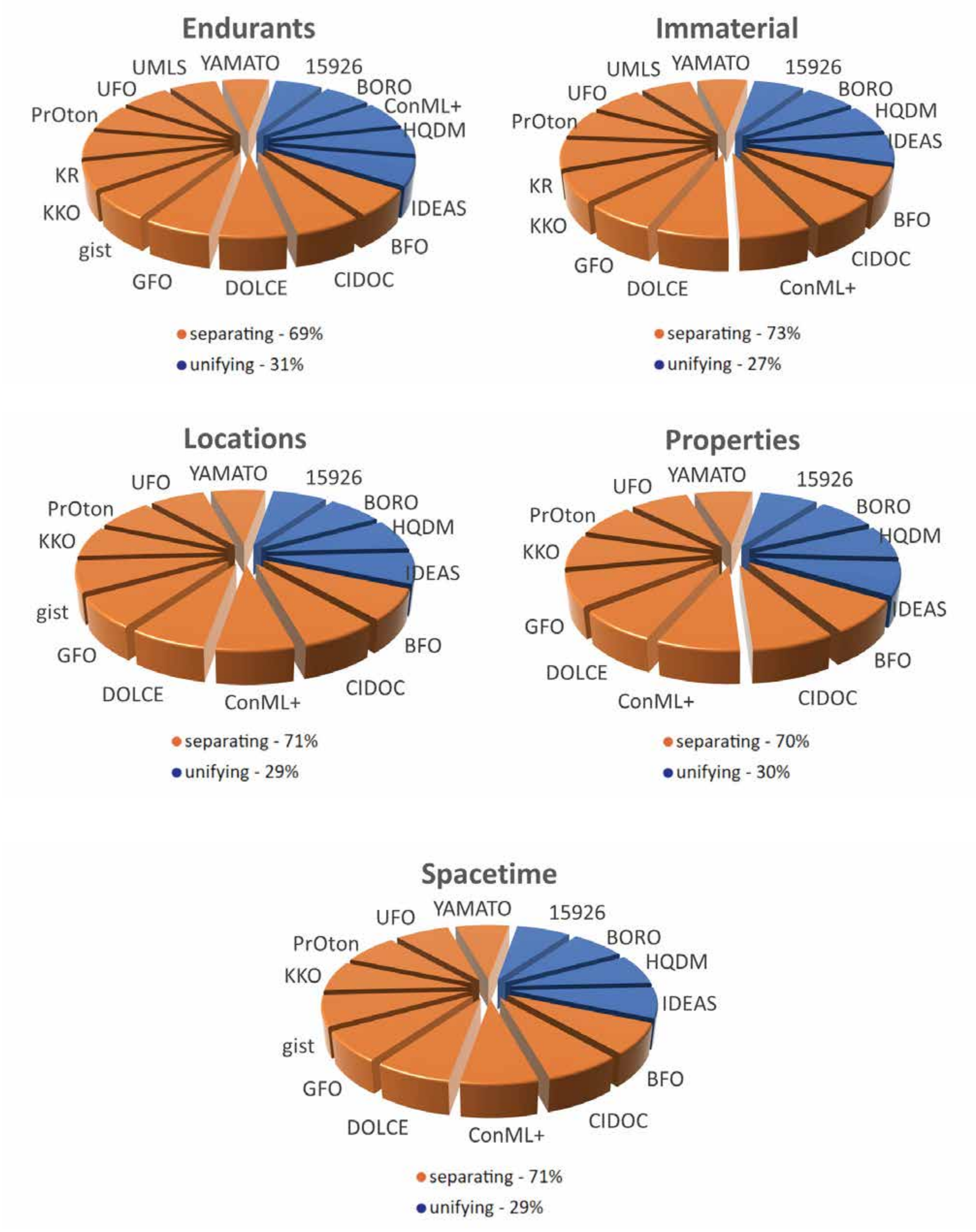
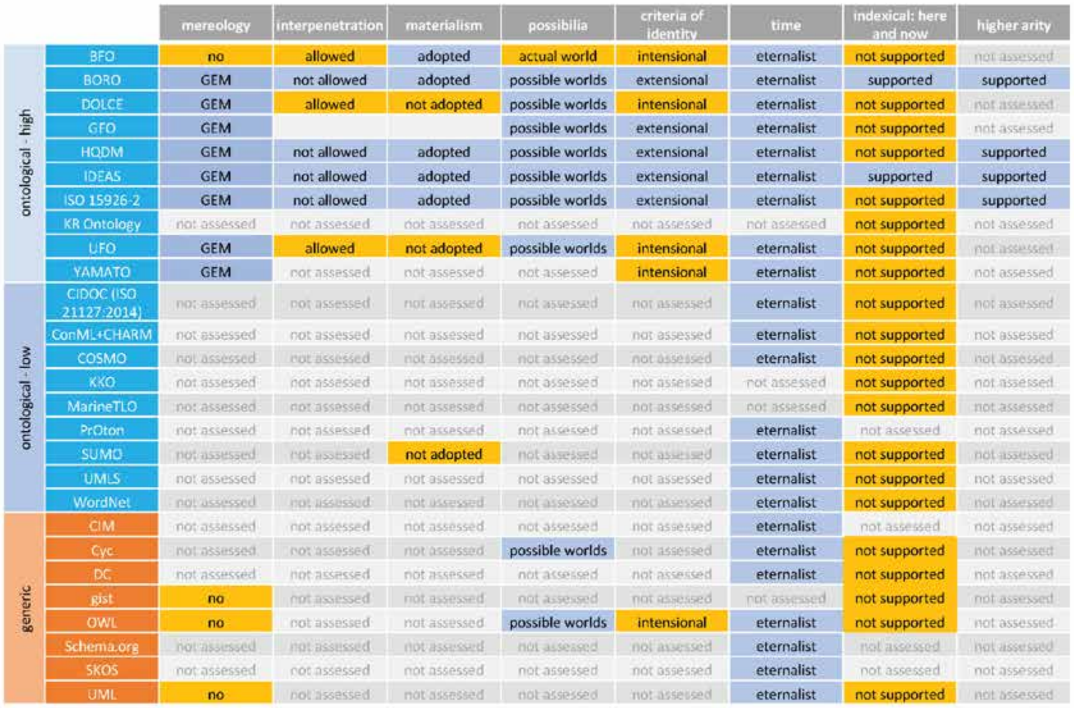


Figure 25 – Vertical aspects – percentages

5.4 Universal commitments

Figure 26 maps the TLOs against the universal commitments – this shows that the heavyweight ontologies have the most commitments. Figure 27 gives the percentages for each choice.



		mereology	interpenetration	materialism	possibilia	criteria of identity	time	indexical: here and now	higher arity
ontological - high	BFO	no	allowed	adopted	actual world	intensional	eternalist	not supported	not assessed
	BORO	GEM	not allowed	adopted	possible worlds	extensional	eternalist	supported	supported
	DOLCE	GEM	allowed	not adopted	possible worlds	intensional	eternalist	not supported	not assessed
	GFO	GEM			possible worlds	extensional	eternalist	not supported	not assessed
	HQDM	GEM	not allowed	adopted	possible worlds	extensional	eternalist	not supported	supported
	IDEAS	GEM	not allowed	adopted	possible worlds	extensional	eternalist	supported	supported
	ISO 15926-2	GEM	not allowed	adopted	possible worlds	extensional	eternalist	not supported	supported
	KR Ontology	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed	not supported	not assessed
	UFO	GEM	allowed	not adopted	possible worlds	intensional	eternalist	not supported	not assessed
	YAMATO	GEM	not assessed	not assessed	not assessed	intensional	eternalist	not supported	not assessed
ontological - low	CIDOC (ISO 21127:2014)	not assessed	not assessed	not assessed	not assessed	not assessed	eternalist	not supported	not assessed
	ConML+CHARM	not assessed	not assessed	not assessed	not assessed	not assessed	eternalist	not supported	not assessed
	COSMO	not assessed	not assessed	not assessed	not assessed	not assessed	eternalist	not supported	not assessed
	KKO	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed	not supported	not assessed
	MarineTLO	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed	not supported	not assessed
	PrOton	not assessed	not assessed	not assessed	not assessed	not assessed	eternalist	not assessed	not assessed
	SUMO	not assessed	not assessed	not adopted	not assessed	not assessed	eternalist	not supported	not assessed
	UMLS	not assessed	not assessed	not assessed	not assessed	not assessed	eternalist	not supported	not assessed
	WordNet	not assessed	not assessed	not assessed	not assessed	not assessed	eternalist	not supported	not assessed
generic	CIM	not assessed	not assessed	not assessed	not assessed	not assessed	eternalist	not assessed	not assessed
	Cyc	not assessed	not assessed	not assessed	possible worlds	not assessed	eternalist	not supported	not assessed
	DC	not assessed	not assessed	not assessed	not assessed	not assessed	eternalist	not supported	not assessed
	gist	no	not assessed	not assessed	not assessed	not assessed	not assessed	not supported	not assessed
	OWL	no	not assessed	not assessed	possible worlds	intensional	eternalist	not supported	not assessed
	Schema.org	not assessed	not assessed	not assessed	not assessed	not assessed	eternalist	not assessed	not assessed
	SKOS	not assessed	not assessed	not assessed	not assessed	not assessed	eternalist	not assessed	not assessed
	UML	no	not assessed	not assessed	not assessed	not assessed	eternalist	not supported	not assessed

Figure 26 – Universal commitments – framework assessment

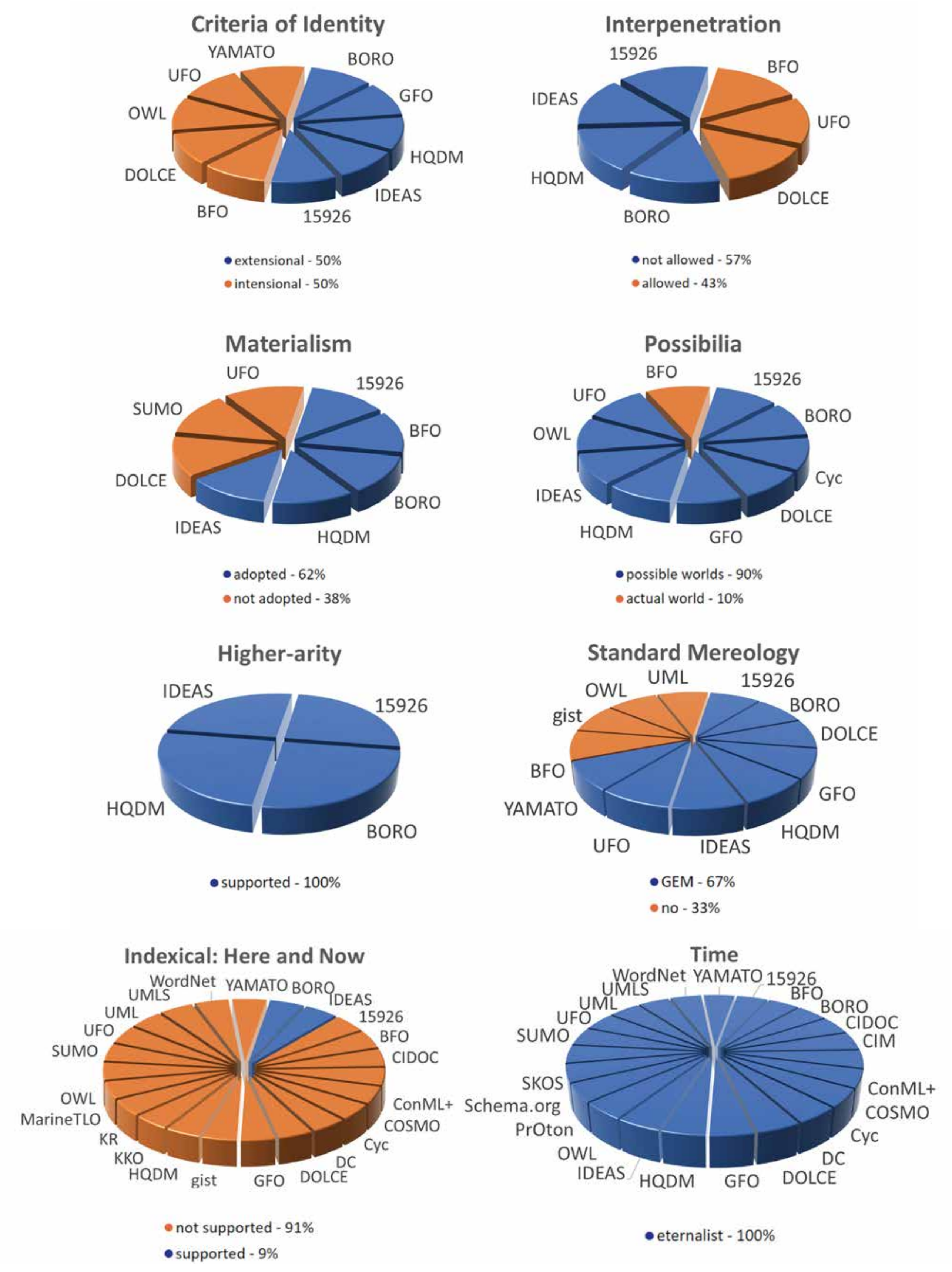


Figure 27 – Universal – percentages

6 Summary

We have developed a list of candidate TLOs – there are currently thirty seven listed in Appendix D (with ten more waiting in a queue to be analysed) and this will be updated as new TLOs are found. This provides a reasonably comprehensive picture of what is currently available.

We have developed a framework for providing an overarching top-down picture of the TLOs. This has three parts. It starts by looking at the general choices; whether to explicitly make ontological commitments and if so, whether to focus on natural language or the real world. It then looks at the formal structure. Finally, it reviews the range of choices of universal ontological commitments typically encountered in TLOs.

The formal structure is divided into two aspects. Firstly, the vertical aspects – the different kinds of formal structures that arise from the core ontological relations. Then the horizontal aspects – the different stratification and unifying choices driven by ontological commitments. The results of this assessment are in the previous section.

We have assessed the candidate TLOs based upon this framework. See Appendix E for the table of results. This reveals that:

- There are a range of levels of explicit ontological commitment, from little or none through lightweight to full-blown heavyweight commitment.
    - In some cases, there is a deliberate decision not to make explicit ontological commitments, but in most cases, the choice seems less deliberate.
    - In many cases, only a few of the choices of ontological commitment are explicitly made or are visible in the TLO's structure. In these cases, the TLO is considered to have a lightweight commitment.
    - One explanation for the lightweight explicit commitment may be the lack of a framework for making the commitments explicit.
    - There is a small, but substantial number of cases where a majority of the choices are reasonably explicitly made.
- Where an ontological commitment is made, some clearly focus on natural language, others on fundamental reality.

- Whether or not a wide range of framework choices have been made is a good indicator of the level of ontological commitment.
    - TLOs with a heavyweight ontological commitment, understandably, tend to explicitly, or otherwise, make a majority of the framework choices.
    - TLOs with a lightweight ontological commitment, understandably, tend to explicitly, or otherwise, make a minority of the framework choices.
    - Generic TLOs that have avoided explicit ontological commitment, often have clear vertical aspects in their formal structure, perhaps indicating the importance of these choices.
- For the horizontal stratifying aspects, there is a general tendency for the heavyweight TLOs to either stratify or unify. This is a similar pattern to that found in philosophy, where stratifying (or unifying) choices tend to reinforce one another.

We have included in Appendix F a brief summary of the TLOs with, where available, a hierarchical picture of their top-most level. We have also added comments on particular details. A quick review of the hierarchical pictures reveals the vast range of top-level structures in current use. A comparison of the lightweight and heavyweight TLO hierarchies, still reveals a wide range of structures. The assessment framework helps to give some idea of where these are based on similar commitments and where they have different commitments.

The results of this survey will be used to perform a structured assessment of the TLOs identified herein, with a view to selecting one or more TLOs that will form the kernel around which the FDM will evolve. A further report – The FDM TLO Selection Paper – will be issued to describe this process in late 2020.

Appendix A
Pathway requirements for a Foundation Data Model

Extract from: The pathway towards an Information Management Framework: A ‘Commons’ for Digital Built Britain (2020) (Hetherington, 2020)

3.5. A Foundation Data Model: clearing up the concepts.

Our Foundation data model will need to address the questions proper to top-level ontology, which can describe general concepts independent of a problem domain. Our FDM should be able to provide answers to:

- Time, space and place: How does the ontology deal with time and space-time? How does the ontology deal with places, locations, shape, holes and a vacuum?
- Actuality and possibility: How does the ontology deal with what could happen or what could be the case, such as where multiple data sets give conflicting stories on the behaviour of a network?
- Classes and types: How does the ontology deal with issues of classification?
- Time and change: How does the ontology deal with time and change?
- Parts, wholes, unity and boundaries: How does the ontology deal with relations of parthood?
- Scale and granularity: How does the ontology deal with scale, resolution and granularity?
- Qualities and other attributes: How does the ontology deal with qualities and other qualitative attributes?
- Quantities and mathematical entities: How does the ontology deal with quantitative data and with mathematical data and theories?
- Processes and events: How does the ontology deal with processes?
- Constitution: How does the ontology deal with the relation – sometimes referred to as a relation of “constitution” – between material entities and the material of which, at any given time, they are made?
- Causality: How does the ontology deal with causality?
- Information and reference: How does the ontology deal with information entities?
- Artefacts and socially constructed entities: How does the ontology deal with artefacts (e.g. engineered items) and socially constructed items like money and laws?

Our FDM will need to select from the available ontological approaches to provide answers to these questions that are as comprehensive and rigorous as possible. These choices will need to be guided by the general requirements of the engineering domain and the domains engineered systems support. This document does not attempt to make these choices now.

- A digital twin is more than just a collection of pieces of data that describes the world. How do we describe the relationship between a digital twin and the corresponding elementary pieces of data?
- How do we describe the domain of validity of a digital twin, including any assumptions or simplifications of real-world behaviour? This may include describing assumptions about the underpinning physics, engineering, biology or sociology that influence the way that the asset operates. How do we define the boundaries of validity of models to ensure that models are used and composed only in ways that are meaningful? Especially for “black box” models, how do we ensure the validity of inputs and outputs and mitigate the risk/damage of erroneous outputs, especially when users will not be close to the development decisions made in the creation of the models.

- How will the data models used by an existing digital twin be validated and interpreted so that mappings and transformations can be established prior to seeking to integrate the twins?
- What is the relationship between a twin and the physical, mathematical or computational model(s) that underpins it? How do we define the non-physical parameters describing the mathematical model used, such as a grid resolution?
- What is the relationship between a digital twin, and the kind of things it describes? Is this a twin of the make and model of my car, or of my specific car? If the latter, what do we call the kind of thing that is a potential twin of all such cars, before I connect it to the telemetry from my specific car? To what extent does such telemetry actually need to be real-time or would it be sufficient to periodically collect, to analyse performance and arrange maintenance? How do we aggregate models that operate at very different tempos – from seconds, to hours to days?
- How do we make statements about time? How do I talk about discrete time periods like "on Thursdays" or "in Summer" or "FY 20/21"? How do we describe and capture change over time? How do we model future operations to assess the impact of planned changes, without disturbing the current operating model?
- How do we break down the physical world into parts? What is the relationship between a twin of a component of a city, such as a building, and a twin of a city? How do we reference the "coarse-graining" that takes place when we have different models, at different resolutions, that overlap in the aspects of the physical world that they describe? How do we aggregate models, particularly in circumstances where there may be modelling gaps? How do we deal with missing information, or unconnected assets?
- How do we handle uncertainty? How should we best manage the difference between "measurement uncertainty", "variability within a class", "variation over time", "environmental noise" and so on?
- Some models will be mechanistic, based on known understanding of the physical world. Others will be purely empirical: based on maximising goodness-of-fit from models to information without incorporation of domain knowledge. Many will be a mix of these paradigms [6]. How will this aspect of the use of digital twins be reflected in the ontology? How reliable is each paradigm, and how can a lack of reliability be taken into account?

It is also worth noting that the scope of the data to be described covers more than just the digital twins themselves. Our data requirements also include the following:

- How do we handle versioning? Should version histories be curated indefinitely? How do we handle archiving and ultimately removal of out-of-date data?
- How is invalid data corrected? What audit trail is needed and how will data ownership be managed and maintained? Are statements recorded together with the identity of the person or organisation making the claim, so that the provenance of information is tracked and unreliable information can be managed? How do we correctly handle missing, invalid or inaccurate data? Outlying or erroneous data can sometimes still be useful when analysed in a different way.
- What relationships do twins have to their authors? Who owns them? Who can know what about them? What kind of roles and actors are there?
- How do we describe not just the models themselves, but the methods that are used to derive insight from them? Models live alongside visualisations, interfaces, deployments and so on, which also need to be described and exchanged.
- How do we describe the uses to which models have been put? Will we need to log each question asked of a digital twin? This will facilitate audit and meta-analysis and save on computational time lost when re-running old studies, but may have information governance and privacy implications.
- How do we model social concepts related to ownership, rights, legislation and regulation? What are the permitted uses of the model and any licence or usage constraints?

References

[6] West (2011) Developing High Quality Data Models. San Francisco, CA: Morgan Kaufmann Publishers Inc.

Appendix B
ISO IEC 21838-1:2019 – Documenting Coverage

Extract from: 2019 – ISO IEC 21838-1:2019 – Information technology – Top-Level Ontologies (TLO) – Part 1: Requirements – Section 4.4.6: Documentation demonstrating breadth of coverage (ISO, 2019, sec. 4.4.6).

4.4.6 Documentation demonstrating breadth of coverage

4.4.6.1 Overview

The ontology documentation shall provide answers to the questions listed in subclauses 4.4.6.2 –4.4.6.16. These answers shall document how the TLO would be used in managing data of the types addressed in each subclause. (Annex C provides examples of such documentation.)

In some TLOs data about entities of given classes or types would be managed by using terms included in the ontology representing those classes or types. Where a TLO does not include classes or types that cover one or more of the areas identified, it shall be documented how it will address corresponding data, for example, by specifying an additional ontology whose relation to the TLO is documented.

NOTE The rationale for requiring breadth of coverage in a TLO is as follows. When an ontology-based approach is adopted, for example, by a large organization in order to promote interoperability of the data systems within its constituent sub-organizations, the ontologies in question will be required to deal with an evolving collection of different sorts of data. These will include:

- data that is spatially and temporally referenced;
- data about entities that change over time;
- data that result from assays along multiple qualitative and quantitative dimensions;
- data reflecting mereological and other relations between such entities, including relations between entities and the material of which they are composed;
- data about data artefacts themselves (for example about designs, plans, requirements specifications).

If it is to have a high likelihood of being able to serve reliably as an over-arching framework for the management of data in such circumstances – even when new sorts of data are being brought on stream – then a TLO requires a maximal breadth of coverage in the set of terms it includes. Similarly, a TLO should include relational expressions that enable representation of a broad range of relations among entities in its chosen categories. Various candidate TLOs have made different – and sometimes incompatible – choices concerning these categories and relations. To show conformity to this document, these choices shall be documented in a way that will justify the claim that the ontology has a sufficiently broad coverage of categories and associated relations to satisfy the requirements of a TLO as defined by this document.

4.4.6.2 Space and time

How does the ontology deal with time, space and spacetime?

- Does the ontology recognize entities which persist in time?
- How does the ontology deal with entities which occur in time?
- Does the ontology recognize entities which are extended in both space and time?
- How does the ontology deal with spatial, temporal and spatiotemporal regions?

4.4.6.3 Actuality and possibility

How does the ontology deal with what could happen or what could be the case, rather than what is the case or has happened?

- How does the ontology deal with possibility?
- Does the ontology support both possible and actual entities?
- Does the ontology have a treatment of dispositions or tendencies?
- Does the ontology have a way of dealing with merely possible or potential entities as might be described in unrealized plans or designs?

4.4.6.4 Classes and types

How does the ontology deal with issues of classification?

- Does classification reflect the existence of certain relations of similarity between certain entities, or do classes or types exist as general entities in addition to particular instances?
- Are classes of classes allowed?
- Does the ontology distinguish between types and the classes of their instances?
- Are classes or types instantiated by the same particulars identical?

4.4.6.5 Time and change

How does the ontology deal with time and change?

- How does the ontology deal with the distinction between past, present and future entities? How does the ontology deal with identity and change of material objects over time? How does the ontology deal with location, and with change of location?
- Does the ontology allow for more than one material object to occupy exactly the same spatial location at the same time?
- How does the ontology deal with changeable properties, such as being a student?
- Does the ontology recognize a distinction between classes or types that apply necessarily to a particular for the whole of its existence, and classes or types that apply only temporarily?

EXAMPLES Mammal is an example of a class or type that applies to a particular for the whole of its existence. An organism is an example of an entity that can undergo change over time, such as by losing hair, without changing identity.

4.4.6.6 Parts, wholes, unity and boundaries

How does the ontology deal with relations of parthood?

- If one entity is part of a second entity, and this entity part of a third entity, does it follow that the first entity is also part of the third entity?
- If one entity is part of but not identical to a second entity, must there be a third entity which makes up the difference?
- How does the ontology deal with wholes formed through the summation of parts?
- How does the ontology deal with continuity where a material object has parts between which there is no natural boundary?
- How does it deal with the factor of unity, which obtains where the parts of a whole are joined together in a way that distinguishes it from a sum?

EXAMPLES Unity is manifested by organisms or planets through the relation of direct or indirect physical connectedness; unity is manifested by solar systems and galaxies through relations of gravity that are above certain thresholds. Unity is manifested by a married couple through the relation of married to, and by a group of siblings through the relation sibling of.

NOTE A whole manifesting the factor of unity can be defined as being such that all its parts are related to each other, and only to each other, by a single distinguished relation.

4.4.6.7 Space and place

How does the ontology deal with places and locations?

- How does the ontology deal with holes, conduits, cavities, a vacuum?
- How does the ontology deal with shape?

4.4.6.8 Scale and granularity

How does the ontology deal with scale, granularity and levels of reality?

- Does the ontology treat the material world as being made up of entities at distinguished levels?

EXAMPLES Atoms, molecules, cells, organisms, planets and galaxies are examples of entities at distinguished levels of reality.

4.4.6.9 Qualities and other attributes

How does the ontology deal with qualities and other attributes?

NOTE 'Attribute' here is meant to include what are sometimes referred to as properties, features or characteristics.

- How do attributes relate to the entities that have or bear them?
- Does the ontology distinguish between attributes and values?
- Does the ontology recognize attributes of attributes?

EXAMPLES Quantitative and qualitative are examples of attributes of attributes.

4.4.6.10 Quantities and mathematical entities

How does the ontology deal with quantitative data and with mathematical data and theories?

- How does the ontology deal with units of measure?
- How are those attributes which are represented using qualitative terms such as 'hot' or 'elevated temperature' related to attributes represented using quantity expressions such as '63 °C'?

4.4.6.11 Processes and events

How does the ontology deal with processes?

- Are processes identical to changes?
- What kinds of processes exist?
- Does the ontology allow attributes of processes?
- Does the ontology distinguish between processes and states?
- Does the ontology recognize instantaneous processes?

4.4.6.12 Constitution

- How does the ontology deal with the relation – sometimes referred to as a relation of 'constitution' – between material entities and the material of which, at any given time, they are made?
- How does the ontology deal with the relation between, for example, minds and brains, persons and organisms, or between organizations and the totality of their members?
- Is there an analogue of the relation of constitution holding between processes, or between non-material entities of other sorts?

4.4.6.13 Causality

- How does the ontology deal with causality?

4.4.6.14 Information and reference

- How does the ontology deal with information entities?

EXAMPLES Databases, symbols, text documents, emails, video files, a speech.

- Does the ontology incorporate a relation between an information entity and what the information entity is about?
- If yes, how does the ontology deal with cases where there is no actual entity which a given information entity is about? Does the ontology deal with cases of this sort by recognizing possible worlds?

EXAMPLE Cases of aboutness where there is no corresponding actual entity may arise where plans for the future are being made.

4.4.6.15 Artefacts and socially constructed entities

- How does the ontology deal with artefacts?

EXAMPLE Engineered items.

- How does the ontology deal with entities commonly viewed as socially constructed, such as money?
- How does the ontology deal with entities such as laws, agreements, duties or permissions?

4.4.6.16 Mental entities; imagined entities; fiction; mythology; religion

- How does the ontology deal with mental entities?

EXAMPLES Minds, thoughts, decisions, memories, images.

- How does the ontology deal with imagined entities?
- How does the ontology deal with entities or data in the realm of mythology?
- How does the ontology deal with entities or data in the realm of fiction?
- How does the ontology deal with entities or data in the realm of religion?

Appendix C
Coverage Mapping to Assessment Framework

The extract from the Pathway requirements for a Foundation Data Model (Hetherington, 2020) document in Appendix A contains a summary of the coverage requirements in ISO IEC 21838-1:2019 – Information technology – Top-Level Ontologies (TLO) – Part 1: Requirements (ISO, 2019) – Section 4.4.6: Documentation demonstrating breadth of coverage – extracted in Appendix B.

Table 9 maps the coverage items onto the items in the assessment framework.

The coverage items were designed for assessing the description of a single TLO rather than assessing a collection of the, so it is no surprise that the mapping is many to many.

Table 9 – Mapping from coverage into the framework assessment

Coverage Item	Description	Assessment Framework
Time, space and place	How does the ontology deal with time and space-time? How does the ontology deal with places, locations, shape, holes and a vacuum?	*horizontal aspect:* • spacetime • locations • immaterial *universal:* • interpenetration • time
Actuality and possibility	How does the ontology deal with what could happen or what could be the case, such as where multiple data sets give conflicting stories on the behaviour of a network?	*Universal:* • possibilia Conflicting datasets are per se an epistemic not an ontological matter.
Classes and types	How does the ontology deal with issues of classification?	*vertical aspect:* • parent-arity • boundedness • stratification • formal generation • relation class-ness *universal:* • criteria of identity
Time and change	How does the ontology deal with time and change?	*horizontal aspect:* • spacetime
Parts, wholes, unity and boundaries	How does the ontology deal with relations of parthood?	*vertical aspect:* • formal generation *horizontal aspect:* • spacetime • locations • immaterial *universal:* • interpenetration
Scale and granularity	How does the ontology deal with scale, resolution and granularity?	Is an epistemic concern.
Qualities and other attributes	How does the ontology deal with qualities and other qualitative attributes?	*horizontal aspect:* • properties
Quantities and mathematical entities	How does the ontology deal with quantitative data and with mathematical data and theories?	*universal:* • materialism

Processes and events	How does the ontology deal with processes?	*horizontal aspect:* - endurants
Constitution	How does the ontology deal with the relation – sometimes referred to as a relation of "constitution" – between material entities and the material of which, at any given time, they are made?	*horizontal aspect:* - spacetime - locations - immaterial *universal:* - interpenetration
Causality	How does the ontology deal with causality?	Low level
Information and reference	How does the ontology deal with information entities?	Low level
Artefacts and socially constructed entities	How does the ontology deal with artefacts (e.g. engineered items) and socially constructed items like money and laws?	Low level

Appendix D
Candidate source top-level ontologies – longlist

Acronym	Preferred Name	Full Name	Initial release	Links	Self Description
BFO	Basic Formal Ontology	Basic Formal Ontology	2002	http://basic-formal-ontology.org/, https://en.wikipedia.org/wiki/Basic_Formal_Ontology, https://en.wikipedia.org/wiki/Upper_ontology#Basic_Formal_Ontology_(BFO)	The Basic Formal Ontology (BFO) framework developed by Barry Smith and his associates consists of a series of sub-ontologies at different levels of granularity. The ontologies are divided into two varieties: relating to continuant entities such as three-dimensional enduring objects, and occurrent entities (primarily) processes conceived as unfolding in successive phases through time. BFO thus incorporates both three-dimensionalist and four-dimensionalist perspectives on reality within a single framework. Interrelations are defined between the two types of ontologies in a way which gives BFO the facility to deal with both static/spatial and dynamic/temporal features of reality. A continuant domain ontology descending from BFO can be conceived as an inventory of entities existing at a time. Each occurrent ontology can be conceived as an inventory of processes unfolding through a given interval of time. Both BFO itself and each of its extension sub-ontologies can be conceived as a window on a certain portion of reality at a given level of granularity.
BORO	BORO	Business Objects Reference Ontology	late 1980s	https://www.borosolutions.net/, https://en.wikipedia.org/wiki/BORO, https://en.wikipedia.org/wiki/Upper_ontology#BORO	Business Objects Reference Ontology is an upper ontology designed for developing ontological or semantic models for large complex operational applications that consists of a top ontology as well as a process for constructing the ontology. It is built upon a series of clear metaphysical choices to provide a solid (metaphysical) foundation. A key choice was for an extensional (and hence, four-dimensional) ontology which provides it a simple criteria of identity. Elements of it have appeared in a number of standards. For example, the ISO standard, ISO 15926 – Industrial automation systems and integration – was heavily influenced by an early version. The IDEAS (International Defence Enterprise Architecture Specification for exchange) standard is based upon BORO, which in turn was used to develop DODAF 2.0.
CIDOC (ISO 21127:2014)	CIDOC Conceptual Reference Model	CIDOC object-oriented Conceptual Reference Model	1999	http://www.cidoc-crm.org/, https://en.wikipedia.org/wiki/CIDOC_Conceptual_Reference_Model, https://en.wikipedia.org/wiki/Upper_ontology#CIDOC_Conceptual_Reference_Model	Although"CIDOC object-oriented Conceptual Reference Model"" (CRM) is a domain ontology, specialised to the purposes of representing cultural heritage, a subset called CRM Core is a generic upper ontology, including:[15][16] - Space-Time – title/identifier, place, era/period, time-span, relationship to persistent items - Events – title/identifier, beginning/ending of existence, participants (people, either individually or in groups), creation/modification of things (physical or conceptional), relationship to persistent items - Material Things – title/identifier, place, the information object the material thing carries, part-of relationships, relationship to persistent items - Immaterial Things – title/identifier, information objects (propositional or symbolic), conceptional things, part-of relationships"
CIM	Common Information Model	Common Information Model	1999	https://www.dmtf.org/standards/cim, https://en.wikipedia.org/wiki/Common_Information_Model_(computing)	The Common Information Model (CIM) is an open standard that defines how managed elements in an IT environment are represented as a common set of objects and relationships between them.
ConML+CHARM	ConML	Conceptual Modelling Language (ConML)	2011	http://www.conml.org/, http://www.conml.org/Resources/TechSpec.aspx, http://www.charminfo.org/	ConML is a conceptual modelling language that has been constructed from scratch with three major goals in mind: - Ease of use for non-experts in information technologies. - Simplicity. - Expressiveness in complex domains, such as those in the humanities. - Capturing “soft” issues such as temporality, subjectivity and vagueness. CHARM is a cultural heritage abstract reference model that extends ConML.
COSMO	COSMO	COmmon Semantic MOdel	not known - pre-2006	http://www.micra.com/, https://en.wikipedia.org/wiki/Upper_ontology#COSMO	Developed with the goal of developing a foundation ontology that can serve to enable broad general Semantic Interoperability.

Acronym	Preferred Name	Full Name	Initial release	Links	Self Description
Cyc	Cyc	Cyc	1984	https://www.cyc.com/the-cyc-platform, https://en.wikipedia.org/wiki/Cyc, https://en.wikipedia.org/wiki/Upper_ontology#Cyc	Artificial intelligence project that aims to assemble a comprehensive ontology and knowledge base that spans the basic concepts and rules about how the world works.
DC	The Dublin Core (DC) ontology	The Dublin Core ontology	1995	http://dublincore.org/, https://en.wikipedia.org/wiki/Dublin_Core	This is a light weight RDFS vocabulary for describing generic metadata.
DOLCE	DOLCE	Descriptive Ontology for Linguistic and Cognitive Engineering	2019	http://www.loa.istc.cnr.it/dolce/overview.html, https://en.wikipedia.org/wiki/Upper_ontology#DOLCE	Is oriented toward capturing the ontological categories underlying natural language and human common sense.
EMMO	EMMO	The European Materials Modelling Ontology (EMMO)	2019 (?)	https://github.com/emmo-repo/EMMO, https://materialsmodelling.com/2019/06/14/european-materials-modelling-ontology-emmo-release/	The EMMO top level is the group of fundamental axioms that constitute the philosophical foundation of the EMMO. Adopting a physicalistic/nominalistic perspective, the EMMO defines real world objects as 4D objects that are always extended in space and time (i.e. real world objects cannot be spaceless nor timeless). For this reason abstract objects, i.e. objects that does not extend in space and time, are forbidden in the EMMO. It has been instigated by materials science and provides the connection between the physical world, the experimental world (materials characterisation) and the simulation world (materials modelling).
FIBO	Financial Industry Business Ontology	Financial Industry Business Ontology	2010 (?)	https://spec.edmcouncil.org/fibo/	The Financial Industry Business Ontology (FIBO) defines the sets of things that are of interest in financial business applications and the ways that those things can relate to one another.
FrameNet	FrameNet	FrameNet	2000 (?)	https://framenet.icsi.berkeley.edu/fndrupal/, https://en.wikipedia.org/wiki/FrameNet	The FrameNet project is building a lexical database of English that is both human- and machine-readable, based on annotating examples of how words are used in actual texts.
GFO	General Formal Ontology	General Formal Ontology	2006	https://www.onto-med.de/ontologies/gfo, https://en.wikipedia.org/wiki/General_formal_ontology, https://en.wikipedia.org/wiki/Upper_ontology#General_Formal_Ontology_(GFO)	Realistic ontology integrating processes and objects. It attempts to include many aspects of recent philosophy, which is reflected both in its taxonomic tree and its axiomatizations.
gist	gist	gist	2007	https://www.semanticarts.com/gist/, https://en.wikipedia.org/wiki/Upper_ontology#gist	gist is developed and supported by Semantic Arts. gist (not an acronym – it means to get the essence of) is a “minimalist upper ontology”. gist is targeted at enterprise information systems, although it has been applied to healthcare delivery applications. The major attributes of gist are: it is small (there are 140 classes and 127 properties) it is comprehensive (most enterprises will not find the need to create additional primitive classes, but will find that most of their classes can be defined and derived from gist) it is robust – all the classes descend from 12 primitive classes, which are mostly mutually disjoint. This aids a great deal in subsequent error detection. There are 1342 axioms, and it uses almost all of the DL constructs (it is SROIQ(D)) it is concrete – most upper ontologies start with abstract philosophical concepts that users must commit to in order to use the ontology. Gist starts with concrete classes that most people already do, or reasonably could agree with, such as Person, Organization, Document, Time, UnitOfMeasure and the like) it is unambiguous – ambiguous terms (such as “term”) have been removed as they are often overloaded and confused. Also terms that frequently have different definitions at different enterprises (such as customer and order) have been removed, also to reduce ambiguity. it is understandable – in addition to being built on concrete, generally understood primitives, it is extremely modular. The 140 classes are implemented in 18 modular ontologies, each can easily be understood in its entirety, and each imports only the other modules that it needs.

Acronym	Preferred Name	Full Name	Initial release	Links	Self Description
HQDM	HQDM	High Quality Data Models	2011	http://www.informationjunction.co.uk/hqdm_framework/	The High Quality Data Models (HQDM) Framework is a 4 dimensionalist top level ontology with extensional identity criteria that aims to support large scale data integration. As such it aims to ensure there is consistency among data created using the framework. The HQDM Framework is based on work developing and using ISO 15926 and lessons learnt from BORO, which influenced ISO 19526.
IDEAS	IDEAS	International Defence Enterprise Architecture Specification	2006	https://en.wikipedia.org/wiki/IDEAS_Group, https://en.wikipedia.org/wiki/Upper_ontology#IDEAS	The upper ontology developed by the IDEAS Group is higher-order, extensional and 4D. It was developed using the BORO Method. The IDEAS ontology is not intended for reasoning and inference purposes; its purpose is to be a precise model of business.
IEC 62541	IEC 62541 - OPC Unified Architecture	IEC 62541 - OPC Unified Architecture	2006	https://opcfoundation.org/developer-tools/specifications-unified-architecture, https://en.wikipedia.org/wiki/OPC_Unified_Architecture	OPC Unified Architecture (OPC UA) is a machine to machine communication protocol for industrial automation developed by the OPC Foundation.
IEC 63088	IEC PAS 63088:2017	Smart manufacturing - Reference architecture model industry 4.0 (RAMI4.0)	2017	https://webstore.iec.ch/publication/30082	IEC PAS 63088:2017(E) describes a reference architecture model in the form of a cubic layer model, which shows technical objects (assets) in the form of layers, and allows them to be described, tracked over their entire lifetime (or “vita”) and assigned to technical and/or organizational hierarchies. It also describes the structure and function of Industry 4.0 components as essential parts of the virtual representation of assets.
ISO 12006-3	ISO 12006-3:2007	ISO 12006-3:2007 - Building construction — Organization of information about construction works — Part 3: Framework for object-oriented information	2007	https://www.iso.org/standard/38706.html, https://en.wikipedia.org/wiki/ISO_12006	ISO 12006-3:2007 specifies a language-independent information model which can be used for the development of dictionaries used to store or provide information about construction works. It enables classification systems, information models, object models and process models to be referenced from within a common framework.
ISO 15926-2	ISO 15926	Industrial automation systems and integration— Integration of life-cycle data for process plants including oil and gas production facilities	2003	https://www.iso.org/standard/29557.html, https://en.wikipedia.org/wiki/ISO_15926, https://en.wikipedia.org/wiki/Upper_ontology#ISO_15926	The ISO 15926 is a standard for data integration, sharing, exchange, and hand-over between computer systems.

Acronym	Preferred Name	Full Name	Initial release	Links	Self Description
KKO	KKO	KBpedia Knowledge Ontology (KKO)	not known	https://kbpedia.org/docs/kko-upper-structure/	KBpedia is a comprehensive knowledge structure for promoting data interoperability and knowledge-based artificial intelligence, or KBAI. The KBpedia knowledge structure combines seven 'core' public knowledge bases — Wikipedia, Wikidata, schema.org, DBpedia, GeoNames, OpenCyc, and standard UNSPSC products and services — into an integrated whole. KBpedia's upper structure, or knowledge graph, is the KBpedia Knowledge Ontology. We base KKO on the universal categories and knowledge representation insights of the great 19th century American logician, polymath and scientist, Charles Sanders Peirce. The upper structure of the KBpedia Knowledge Ontology (KKO) is informed by the triadic logic and universal categories of Charles Sanders Peirce. This trichotomy, also the basis for his views on semiosis (or the nature of signs), was in Peirce's view the most primitive or reduced manner by which to understand and categorize things, concepts and ideas.
KR Ontology	KR Ontology	KR Ontology	1999	http://www.jfsowa.com/ontology/toplevel.htm	The KR Ontology is defined in the book Knowledge Representation by John F. Sowa. Its categories have been derived from a synthesis of various sources, but the two major influences are the semiotics of Charles Sanders Peirce and the categories of existence of Alfred North Whitehead. The primitive categories are: Independent, Relative, or Mediating; Physical or Abstract; Continuant or Occurrent.
MarineTLO	MarineTLO	Marine Top Level Ontology	2013 (?)	https://projects.ics.forth.gr/isl/MarineTLO/, https://en.wikipedia.org/wiki/Upper_ontology#MarineTLO	Is a top level ontology, generic enough to provide consistent abstractions or specifications of concepts included in all data models or ontologies of marine data sources and provide the necessary properties to make this distributed knowledge base a coherent source of facts relating observational data with the respective spatiotemporal context and categorical (systematic) domain knowledge.
MIMOSA CCOM	MIMOSA CCOM	MIMOSA CCOM (Machinery Information Management Open Systems Alliance - Common Conceptual Object Model)	not known	https://www.mimosa.org/mimosa-ccom/ , https://en.wikipedia.org/wiki/OpenO%26M	MIMOSA CCOM (Common Conceptual Object Model) serves as an information model for the exchange of asset information. Its core mission is to facilitate standards-based interoperability between systems: providing an XML model to allow systems to electronically exchange data.
OWL	Web Ontology Language	OWL	2004	https://www.w3.org/OWL/, https://en.wikipedia.org/wiki/Web_Ontology_Language	The Web Ontology Language (OWL) is a family of knowledge representation languages for authoring ontologies. Ontologies are a formal way to describe taxonomies and classification networks, essentially defining the structure of knowledge for various domains: the nouns representing classes of objects and the verbs representing relations between the objects.
PROTON	PROTON	PROTo ONtology	2005 (?)	https://ontotext.com/documents/proton/Proton-Ver3.0B.pdf, https://en.wikipedia.org/wiki/Upper_ontology#PROTON	Is designed as a lightweight upper-level ontology for use in Knowledge Management and Semantic Web applications.
Schema.org	Schema.org	Schema.org	2011	https://schema.org/, https://en.wikipedia.org/wiki/Schema.org	Schema.org is a collaborative, community activity with a mission to create, maintain, and promote schemas for structured data on the Internet, on web pages, in email messages, and beyond.
SENSUS	The SENSUS ontology	The SENSUS ontology	2001	https://www.isi.edu/natural-language/projects/ONTOLOGIES.html	We have constructed SENSUS, a 70,000-node terminology taxonomy, as a framework into which additional knowledge can be placed. SENSUS is an extension and reorganization of WordNet.
SKOS	SKOS	Simple Knowledge Organization System	2009	https://www.w3.org/2004/02/skos/, https://en.wikipedia.org/wiki/Simple_Knowledge_Organization_System	SKOS is an area of work developing specifications and standards to support the use of knowledge organization systems (KOS) such as thesauri, classification schemes, subject heading systems and taxonomies within the framework of the Semantic Web.

Acronym	Preferred Name	Full Name	Initial release	Links	Self Description
SUMO	SUMO	Suggested Upper Merged Ontology	2000	http://www.adampease.org/OP/, https://en.wikipedia.org/wiki/Suggested_Upper_Merged_Ontology, https://en.wikipedia.org/wiki/Upper_ontology#SUMO_(Suggested_Upper_Merged_Ontology)	Is an upper ontology intended as a foundation ontology for a variety of computer information processing systems. SUMO is organized for interoperability of automated reasoning engines. It isbeing used for research and applications in search, linguistics and reasoning.
TMRM	The Topic Maps Reference Model	The Topic Maps Reference Model	late 1990s	https://www.isotopicmaps.org/tmrm/, https://en.wikipedia.org/wiki/Topic_map	A topic map is a standard for the representation and interchange of knowledge, with an emphasis on the findability of information.
UFO	UFO	Unified Foundational Ontology	2005	https://nemo.inf.ufes.br/en/projetos/ufo/, https://en.wikipedia.org/wiki/OntoUML, https://en.wikipedia.org/wiki/Upper_ontology#UFO_(Unified_Foundational_Ontology)	Incorporates developments from GFO, DOLCE and the Ontology of Universals underlying OntoClean in a single coherent foundational ontology.
UMBEL	UMBEL	Upper Mapping and Binding Exchange Layer	2008	https://en.wikipedia.org/wiki/UMBEL, https://en.wikipedia.org/wiki/Upper_ontology#UMBEL	Is a logically organized knowledge graph of 34,000 concepts and entity types that can be used in information science for relating information from disparate sources to one another. Since UMBEL is an open-source extract of the OpenCyc knowledge base, it can also take advantage of the reasoning capabilities within Cyc.
UML	UML	Unified Modeling Language (UML)	1994	http://uml.org/, https://en.wikipedia.org/wiki/Unified_Modeling_Language	The Unified Modeling Language (UML) is a general-purpose, developmental, modeling language in the field of software engineering that is intended to provide a standard way to visualize the design of a system.
UMLS	Unified Medical Language System	UMLS	1986	https://www.nlm.nih.gov/research/umls/index.html, https://en.wikipedia.org/wiki/Unified_Medical_Language_System	The Unified Medical Language System (UMLS) is a compendium of many controlled vocabularies in the biomedical sciences (created 1986).[1] It provides a mapping structure among these vocabularies and thus allows one to translate among the various terminology systems; it may also be viewed as a comprehensive thesaurus and ontology of biomedical concepts. UMLS further provides facilities for natural language processing. It is intended to be used mainly by developers of systems in medical informatics.
WordNet	WordNet	WordNet	1985	https://wordnet.princeton.edu/, https://en.wikipedia.org/wiki/WordNet, https://en.wikipedia.org/wiki/Upper_ontology#WordNet	WordNet® is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept.
YAMATO	YAMATO	Yet Another More Advanced Top Ontology	1999	https://en.wikipedia.org/wiki/Upper_ontology#YAMATO_(Yet_Another_More_Advanced_Top_Ontology)	YAMATO: Yet Another More Advanced Top-level Ontology which has been developed intended to cover three features in Quality description, Representation and Process/Event, respectively, in a better way than existing ontologies. It has been extensively used for developing other, more applied, ontologies.

Appendix E
Summary of Framework Assessment Matrix Results

category	general			
type	ontologically committed	commitment level	subject	categorical
relation				
characteristic				
choice	ontological or generic	high or low	foundational or natural language	yes or no
BFO	ontological	high	foundational	no
BORO	ontological	high	foundational	yes
YAMATO	ontological	high	foundational	not assessed
HQDM	ontological	high	foundational	yes
IDEAS	ontological	high	foundational	yes
ISO 15926-2	ontological	high	foundational	yes
UFO	ontological	high	foundational	yes
GFO	ontological	high	foundational	yes
KR Ontology	ontological	high	foundational	yes
DOLCE	ontological	high	natural language	yes
ConML+CHARM	ontological	low	foundational	yes
CIDOC (ISO 21127:2014)	ontological	low	foundational	yes
MarineTLO	ontological	low	foundational	yes
COSMO	ontological	low	foundational ??	yes
EMMO	ontological	low	foundational ??	yes
SENSUS	ontological	low	natural language	yes
FrameNet	ontological	low	natural language	yes
PrOton	ontological	low	natural language	yes
WordNet	ontological	low	natural language	yes
UMLS	ontological	low	natural language	yes
KKO	ontological	low	natural language	yes
SUMO	ontological	low	natural language	yes
FIBO	not yet assessed	not yet assessed	not yet assessed	yes
IEC 62541	not available	not available	not available	not assessed
IEC 63088	not available	not available	not available	not assessed
ISO 12006-3	generic			not assessed
SKOS	generic			not assessed
MIMOSA CCOM	generic			not assessed
CIM	generic			not assessed
DC	generic			not assessed
Schema.org	generic			not assessed
TMRM	generic			not assessed
UML	generic			yes
gist	generic			yes
OWL	generic			yes
UMBEL	generic			yes
Cyc	generic			yes

category	vertical aspect													
type	parent-arity		transitivity	boundedness			stratification	formal generation					relation class-ness	
relation	type-instance	super-sub-type	super-sub-type	type-instance			type-instance	whole-part		type-instance	super-sub-type		type-instance	super-sub-type
characteristic				downwards	fixed finite levels	number of fixed levels		fusion	complement	fusion	fusion	complement		
choice	single or unconstrained	single or unconstrained	yes or no	bounded or unbounded	fixed or not-fixed	[a number]	stratified or unstratified	yes or no	yes or no	yes or no	yes or no	yes or no	first- or second-class	first- or second-class
BFO	unconstrained	single	yes	bounded	fixed	2	stratified	no	no	no	no	no	second-class	second-class
BORO	unconstrained	unconstrained	yes	bounded	not-fixed	not applicable	unstratified	yes	yes	yes	yes	yes	first-class	first-class
YAMATO	not assessed	single	yes	bounded	fixed	2	stratified	yes	yes	not assessed	not assessed	not assessed	second-class	second-class
HQDM	unconstrained	unconstrained	yes	bounded	not-fixed	not applicable	unstratified	yes	yes	yes	yes	yes	first-class	first-class
IDEAS	unconstrained	unconstrained	yes	bounded	not-fixed	not applicable	unstratified	yes	yes	yes	yes	yes	first-class	first-class
ISO 15926-2	unconstrained	unconstrained	yes	bounded	not-fixed	not applicable	unstratified	yes	yes	yes	yes	yes	first-class	first-class
UFO	unconstrained	unconstrained	yes	bounded	not-fixed	not applicable	stratified	no	no	no	no	no	second-class	second-class
GFO	unconstrained	single	yes	bounded	not-fixed	not applicable	unstratified	yes	yes	yes	yes	yes	second-class	second-class
KR Ontology	not yet assessed	unconstrained	yes	not yet assessed	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed	second-class	second-class
DOLCE	unconstrained	single	yes	bounded	fixed	2	stratified	yes	yes	no	no	no	second-class	second-class
ConML+CHARM	unconstrained	single	yes	bounded	fixed	2	stratified	no	no	no	no	no	second-class	second-class
CIDOC (ISO 21127:2014)	unconstrained ?	unconstrained	yes	bounded	fixed	2	stratified	no	no	no	no	no	second-class	second-class
MarineTLO	unconstrained ?	unconstrained	yes	bounded	fixed	2	stratified	no	no	no	no	no	second-class	second-class
COSMO	unclear	single	yes	bounded	fixed	2 ??	stratified	not assessed	not assessed	not assessed	not assessed	not assessed	second-class	second-class
EMMO	not assessed	not assessed	not assessed	bounded	not-fixed	not applicable	unclear	unclear	unclear	unclear	unclear	unclear	second-class	second-class
SENSUS	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed
FrameNet	unconstrained	single	not assessed	bounded	fixed	2	stratified	not assessed	not assessed	not assessed	not assessed	not assessed	second-class	second-class
PrOton	unconstrained	unconstrained	not assessed	bounded	fixed	2	stratified	no	no	no	no	no	second-class	second-class
WordNet	unconstrained	single	not assessed	bounded	fixed	2	stratified	no	no	no	no	no	second-class	second-class

category	vertical aspect													
type	parent-arity		transitivity	boundedness			stratification	formal generation					relation class-ness	
relation	type-instance	super-sub-type	super-sub-type	type-instance			type-instance	whole-part		type-instance	super-sub-type		type-instance	super-sub-type
characteristic				downwards	fixed finite levels	number of fixed levels		fusion	complement	fusion	fusion	complement		
choice	single or unconstrained	single or unconstrained	yes or no	bounded or unbounded	fixed or not-fixed	[a number]	stratified or unstratified	yes or no	yes or no	yes or no	yes or no	yes or no	first- or second-class	first- or second-class
UMLS	unconstrained	single	not assessed	bounded	fixed	2	stratified	no	no	no	no	no	second-class	second-class
KKO	not assessed	not assessed	not assessed	bounded	fixed	2	stratified	not assessed	not assessed	not assessed	not assessed	not assessed	second-class	second-class
SUMO	unconstrained	unconstrained	yes	unbounded	not-fixed	not applicable	unstratified	no	no	no	no	no	second-class	second-class
FIBO	not yet assessed	not yet assessed	not assessed	not yet assessed	not yet assessed	not yet assessed	not yet assessed	not yet assessed	not yet assessed	not yet assessed	not yet assessed	not yet assessed	not yet assessed	not yet assessed
IEC 62541	not available	not available	not assessed	not available	not available	not available	not available	not available	not available	not available	not available	not available	not available	not available
IEC 63088	not available	not available	not assessed	not available	not available	not available	not available	not available	not available	not available	not available	not available	not available	not available
ISO 12006-3	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed
SKOS	not assessed	not assessed	no	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed
MIMOSA CCOM	not available	not available	not assessed	not available	not available	not available	not available	not available	not available	not available	not available	not available	not available	not available
CIM	not assessed	single	not assessed	bounded	fixed	2	stratified	no	no	no	no	no	second-class	second-class
DC	unconstrained	unconstrained	not assessed	bounded	fixed	2	stratified	not assessed	not assessed	not assessed	not assessed	not assessed	second-class	second-class
Schema.org	unconstrained	unconstrained	not assessed	unbounded	not-fixed	not applicable	unstratified	no	no	no	no	no	second-class	second-class
TMRM	unconstrained	unconstrained	not assessed	unbounded	not-fixed	not applicable	unstratified	no	no	no	no	no	second-class	second-class
UML	unconstrained	unconstrained	no	bounded	fixed	3	stratified	no	no	no	no	no	first-class	first-class
gist	unclear	single (mostly)	not assessed	bounded	fixed	2	stratified	not assessed	not assessed	not assessed	not assessed	not assessed	second-class	second-class
OWL	unconstrained	unconstrained	yes	bounded	fixed	2	stratified	no	no	no	no	no	second-class	second-class
UMBEL	unconstrained	unconstrained	not assessed	bounded	not-fixed	not applicable	unstratified	unclear	unclear	unclear	unclear	unclear	second-class	second-class
Cyc	unconstrained	unconstrained	not assessed	bounded	not-fixed	not applicable	unstratified	not assessed	not assessed	not assessed	not assessed	not assessed	second-class	second-class

category	horizontal aspect				
type	spacetime	locations	properties	endurants	immaterial
relation					
characteristic					
choice	unifying or separating	unifying or separating	unifying or separating	unifying or separating	unifying or separating
BFO	separating	separating	separating	separating	separating
BORO	unifying	unifying	unifying	unifying	unifying
YAMATO	separating	separating	separating	separating	separating
HQDM	unifying	unifying	unifying	unifying	unifying
IDEAS	unifying	unifying	unifying	unifying	unifying
ISO 15926-2	unifying	unifying	unifying	unifying	unifying
UFO	separating	separating	separating	separating	separating
GFO	separating	separating	separating	separating	separating
KR Ontology	not assessed	not assessed	not assessed	separating	separating
DOLCE	separating	separating	separating	separating	separating
ConML+CHARM	separating	separating	separating	unifying	separating
CIDOC (ISO 21127:2014)	separating	separating	separating	separating	separating
MarineTLO	not assessed	not assessed	not assessed	not assessed	not assessed
COSMO	not assessed	not assessed	not assessed	not assessed	not assessed
EMMO	not assessed	not assessed	not assessed	not assessed	not assessed
SENSUS	not assessed	not assessed	not assessed	not assessed	not assessed
FrameNet	not assessed	not assessed	not assessed	not assessed	not assessed
PrOton	separating	separating	separating	separating	separating
WordNet	not assessed	not assessed	not assessed	not assessed	not assessed
UMLS	not assessed	not assessed	not assessed	separating	separating
KKO	separating	separating	separating	separating	separating
SUMO	not assessed	not assessed	not assessed	not assessed	not assessed
FIBO	not yet assessed	not yet assessed	not yet assessed	not yet assessed	not yet assessed
IEC 62541	not available	not available	not available	not available	not available
IEC 63088	not available	not available	not available	not available	not available
ISO 12006-3	not assessed	not assessed	not assessed	not assessed	not assessed
SKOS	not assessed	not assessed	not assessed	not assessed	not assessed
MIMOSA CCOM	not assessed	not assessed	not assessed	not assessed	not assessed
CIM	not assessed	not assessed	not assessed	not assessed	not assessed
DC	not assessed	not assessed	not assessed	not assessed	not assessed
Schema.org	not assessed	not assessed	not assessed	not assessed	not assessed
TMRM	not assessed	not assessed	not assessed	not assessed	not assessed
UML	not assessed	not assessed	not assessed	not assessed	not assessed
gist	separating	separating	not assessed	separating	not assessed
OWL	not assessed	not assessed	not assessed	not assessed	not assessed
UMBEL	not assessed	not assessed	not assessed	not assessed	not assessed
Cyc	not assessed	not assessed	not assessed	not assessed	not assessed

category	universal							
type	mereology	interpenetration	materialism	possibilia	criteria of identity	time	indexicals: here and now	higher arity
relation								
characteristic								
choice	standard	allowed or not allowed	adopted or not adopted	possible worlds or actual world	intensional or extensional	presentist or eternalist	supported - not supported	supported - not supported
BFO	no	allowed	adopted	actual world	intensional	eternalist	not supported	not assessed
BORO	GEM	not allowed	adopted	possible worlds	extensional	eternalist	supported	supported
YAMATO	GEM	not assessed	not assessed	not assessed	intensional	eternalist	not supported	not assessed
HQDM	GEM	not allowed	adopted	possible worlds	extensional	eternalist	not supported	supported
IDEAS	GEM	not allowed	adopted	possible worlds	extensional	eternalist	supported	supported
ISO 15926-2	GEM	not allowed	adopted	possible worlds	extensional	eternalist	not supported	supported
UFO	GEM	allowed	not adopted	possible worlds	intensional	eternalist	not supported	not assessed
GFO	GEM			possible worlds	extensional	eternalist	not supported	not assessed
KR Ontology	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed	not supported	not assessed
DOLCE	GEM	allowed	not adopted	possible worlds	intensional	eternalist	not supported	not assessed
ConML+CHARM	not assessed	not assessed	not assessed	not assessed	not assessed	eternalist	not supported	not assessed
CIDOC (ISO 21127:2014)	not assessed	not assessed	not assessed	not assessed	not assessed	eternalist	not supported	not assessed
MarineTLO	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed	not supported	not assessed
COSMO	not assessed	not assessed	not assessed	not assessed	not assessed	eternalist	not supported	not assessed
EMMO	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed
SENSUS	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed
FrameNet	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed
PrOton	not assessed	not assessed	not assessed	not assessed	not assessed	eternalist	not assessed	not assessed
WordNet	not assessed	not assessed	not assessed	not assessed	not assessed	eternalist	not supported	not assessed
UMLS	not assessed	not assessed	not assessed	not assessed	not assessed	eternalist	not supported	not assessed
KKO	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed	not supported	not assessed
SUMO	not assessed	not assessed	not adopted	not assessed	not assessed	eternalist	not supported	not assessed
FIBO	not assessed	not yet assessed	not yet assessed	not yet assessed	not yet assessed	not yet assessed	not yet assessed	not assessed
IEC 62541	not assessed	not available	not available	not available	not available	not available	not available	not assessed
IEC 63088	not assessed	not available	not available	not available	not available	not available	not available	not assessed
ISO 12006-3	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed
SKOS	not assessed	not assessed	not assessed	not assessed	not assessed	eternalist	not assessed	not assessed
MIMOSA CCOM	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed
CIM	not assessed	not assessed	not assessed	not assessed	not assessed	eternalist	not assessed	not assessed
DC	not assessed	not assessed	not assessed	not assessed	not assessed	eternalist	not supported	not assessed
Schema.org	not assessed	not assessed	not assessed	not assessed	not assessed	eternalist	not assessed	not assessed
TMRM	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed
UML	no	not assessed	not assessed	not assessed	not assessed	eternalist	not supported	not assessed
gist	no	not assessed	not assessed	not assessed	not assessed	not assessed	not supported	not assessed
OWL	no	not assessed	not assessed	possible worlds	intensional	eternalist	not supported	not assessed
UMBEL	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed	not assessed
Cyc	not assessed	not assessed	not assessed	possible worlds	not assessed	eternalist	not supported	not assessed

Appendix F
Selected candidate source top-level ontologies – details

F.1 Introduction

This appendix is intended to give a feel for the range of different approaches to top-level ontologies, including those that make little or no ontological commitment.

We provide a brief overview (usually a self-description) and a picture of their top structure, where available. We also include comments on key ontological characteristics. A selection of relevant extracts is included as to give more insight into the characteristics.

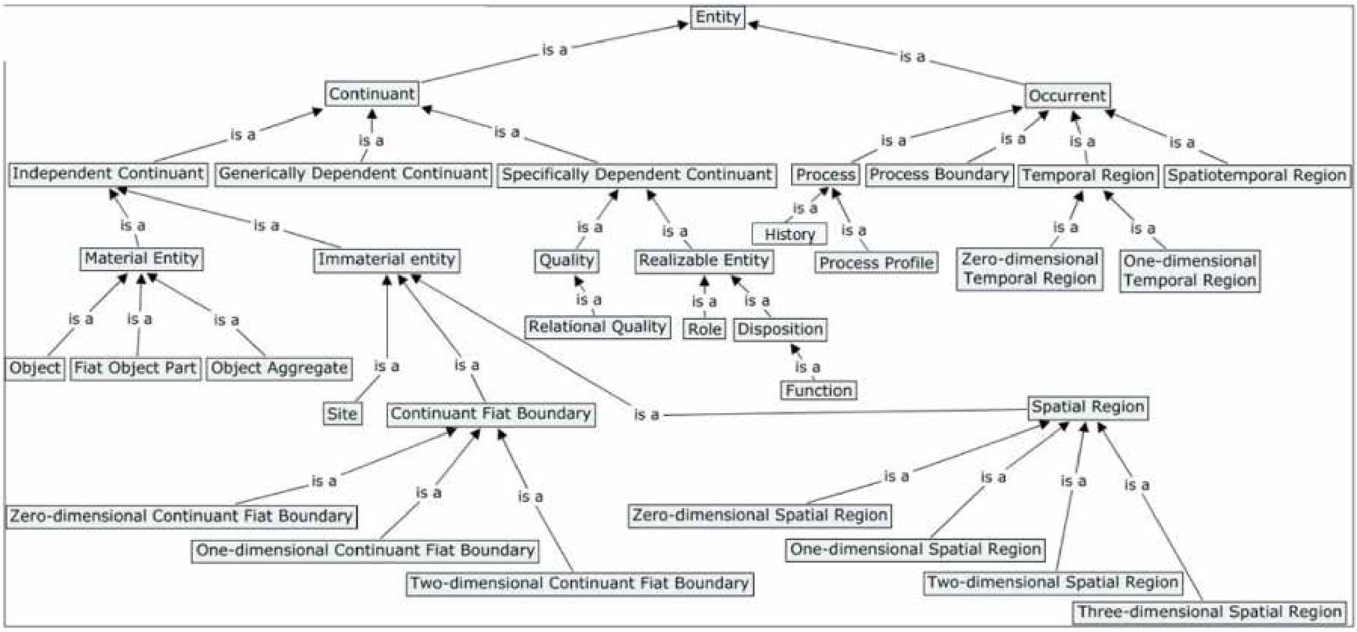
F.2 BFO – Basic Formal Ontology

F.2.1 Overview

The Basic Formal Ontology (BFO) framework developed by Barry Smith and his associates consists of a series of sub-ontologies at different levels of granularity. The ontologies are divided into two varieties: relating to continuant entities such as three-dimensional enduring objects, and occurrent entities (primarily) processes conceived as unfolding in successive phases through time. BFO thus incorporates both three-dimensionalist and four-dimensionalist perspectives on reality within a single framework. Interrelations are defined between the two types of ontologies in a way which gives BFO the facility to deal with both static/spatial and dynamic/temporal features of reality. A continuant domain ontology descending from BFO can be conceived as an inventory of entities existing at a time. Each occurrent ontology can be conceived as an inventory of processes unfolding through a given interval of time. Both BFO itself and each of its extension sub-ontologies can be conceived as a window on a certain portion of reality at a given level of granularity.

From https://en.wikipedia.org/wiki/Basic_Formal_Ontology

F.2.2 Top-level



F.2.3 Key characteristics

BFO is a well-documented heavyweight foundational ontology.

It has an interesting horizontal stratification, which is documented in the journey in Figure 28.

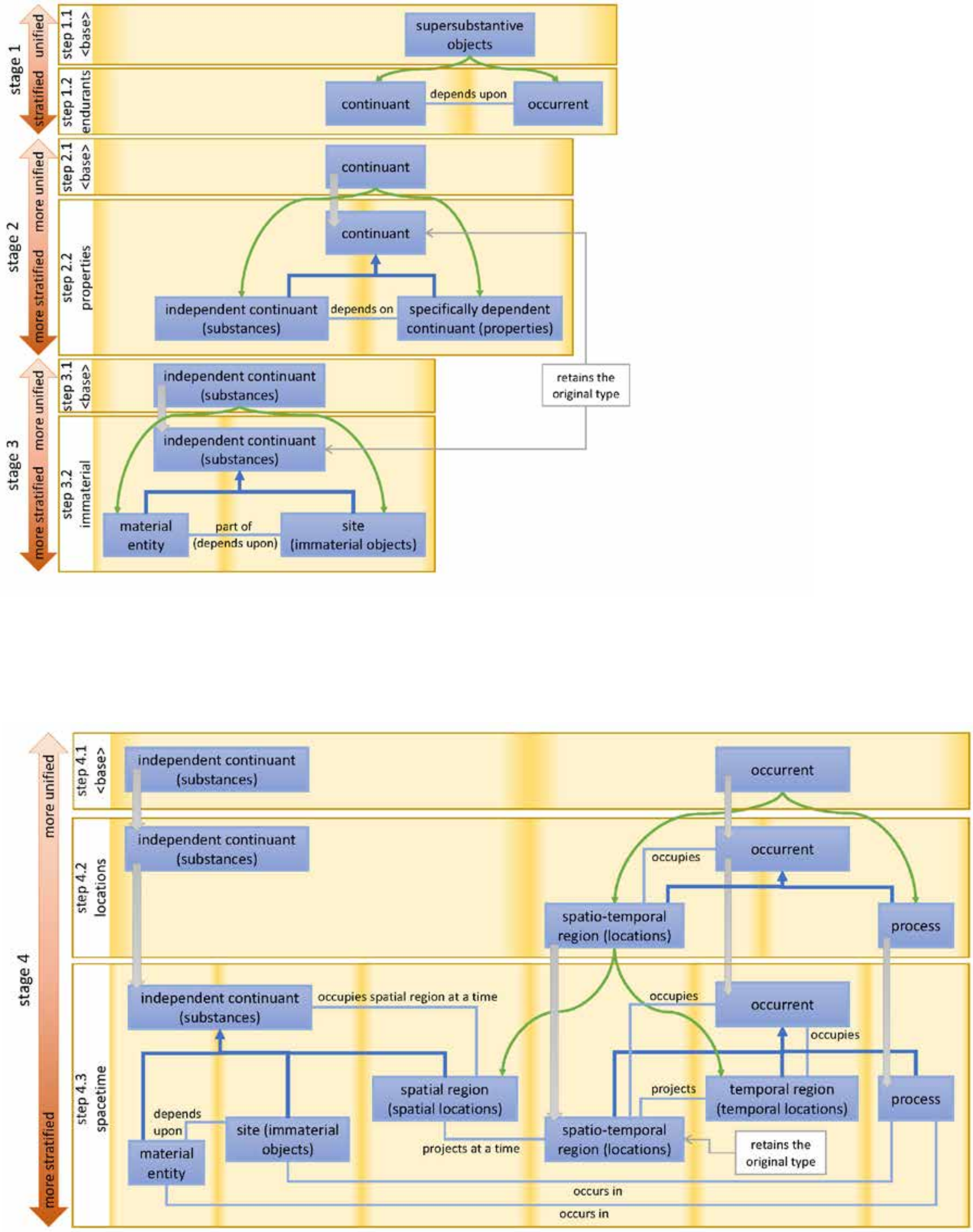


Figure 28 – BFO Stratification Journey – six strata (time indexed relations suffixed with '... at a time)

This shows an unusual architecture for spacetime. While electing to be separatist about space and time, the TLO also retains spacetime. This results in both spatial and temporal redundancy. Another is the single spatio-temporal reference frame (currently) – see 3.2.1 below.

Single super-sub-type parent-arity for universals – The monohierarchy principle see 2.7 below.

Possibilia: Actualist about worlds – see 3.14 below – that is, no possible worlds. Uses dispositions for some aspects of modality. It is unclear whether it supports a full-blown ontology of modality.

Interpenetration – for example, material and immaterial entities can interpenetrate – a person (material) can stand in a doorway (immaterial) – they are related by having overlapping spatial regions not sharing parts.

Mereology – own version. For example – an immaterial object – the hold of the ship – is a part of the ship, but the material objects in the hold are not part of it, though they are situated in it. Based upon Minimal Extensional Mereology (see below).

Extensible – currently excludes numbers.

F.2.4 Relevant extracts

These extracts from: Basic Formal Ontology 2.0 – SPECIFICATION AND USER'S GUIDE (https://github.com/BFO-ontology/BFO/raw/master/docs/bfo2-reference/BFO2-Reference.pdf).

Extract 1 – Single Inheritance

2.7 The monohierarchy principle

BFO rests on a number of heuristic principles that are designed to advance its utility to formal reasoning. These take the form of simple rules – analogous to the rules of the road – that are designed to promote consistency in the making of both domain-neutral and domain-specific choices in ontology construction. [19] One heuristic principle of this kind – expressing what we can think of as a principle of good behavior in the realm of universals – asserts that the asserted taxonomies of types and subtypes in BFO-conformant ontologies should be genuine trees (in the graph-theoretic sense), so that each node in the graph of universals should have at most one asserted is_a parent. (On the use of 'asserted' here, see [19].) This principle is of value not only because it supports a simple strategy for the formulation of definitions and thereby helps to prevent certain common kinds of error in ontology construction, but also because it brings technical benefits when ontologies are implemented computationally.

[19] Barry Smith and Werner Ceusters, "Ontological Realism as a Methodology for Coordinated Evolution of Scientific Ontologies", Applied Ontology, 5 (2010), 139–188. PMC3104413

Extract 2 – Modality – Actualist

3.7.8 Material basis

Dispositions (and thus also functions) are introduced into BFO in order to provide a means for referring to what we can think of as the potentials or powers of things in the world without the need to quantify over putative 'possible worlds' or 'possible objects'.

...

Extract 3 – Location – Separatist and Unitist

3.14 Spatiotemporal region

> ELUCIDATION: A spatiotemporal region is an occurrent entity that is part of spacetime. [095-001]

'Spacetime' here refers to the maximal instance of the universal spatiotemporal region.

...

3.15 Temporal region

Given a temporal reference frame R, we can define 'time_R' as the maximal **instance** of the universal *temporal region*.

> ELUCIDATION: A *temporal region* is an *occurrent* entity that is **part** of time as defined relative to some reference frame. [100-001]

AXIOM: Every *temporal region t* is such that *t* **occupies_temporal_region** *t*. [119-002]

AXIOM: All parts of temporal regions are temporal regions. [101-001]

zero-dimensional temporal region

ELUCIDATION: A *zero-dimensional temporal* region is a temporal region that is without extent. [102-001]

EXAMPLES: a temporal region that is occupied by a process boundary; right now; the moment at which a finger is detached in an industrial accident; the moment at which a child is born, the moment of death.

SYNONYM: temporal instant.

…

Extract 4 – Location – Reference frames

3.2.1 Excursus on frames

The four dimensions of the spacetime continuum are not homogeneous. Rather there is one time-like and three space-like dimensions. This heterogeneity is sufficient, for the purposes of BFO, to justify our division of reality in a way that distinguishes spatial and temporal regions. In a future version, however, we will need to do justice to the fact that there are multiple ways of dividing up the spacetime continuum into spatial and temporal regions, corresponding to multiple frames that might be used by different observers.

…

3.6.3 Spatial region

We recommend that users of BFO region terms specify the coordinate frame in terms of which their spatial and temporal data are represented. When dealing with spatial regions on the surface of the Earth, for example, this will be the coordinate frame of latitude and longitude, potentially supplemented by the dimension of altitude.

…

Extract 5 – Endurantist – Occurrent dependence on Continuants

3.7.2 No s-dependence of higher order

BFO does not recognize universals of higher order (for example, the universal universal). All universals are instantiated by instance entities which are not universals.

Extract 6 – Mereology – Minimal Extensional Mereology

3 Specification

3.1 Relations of parthood

As our starting point in understanding the parthood relation, we take the axioms of Minimal Extensional Mereology as defined by Simons [46, pp. 26-31], assuming, with Simons, the axioms of first order predicate calculus.

F.3 BORO

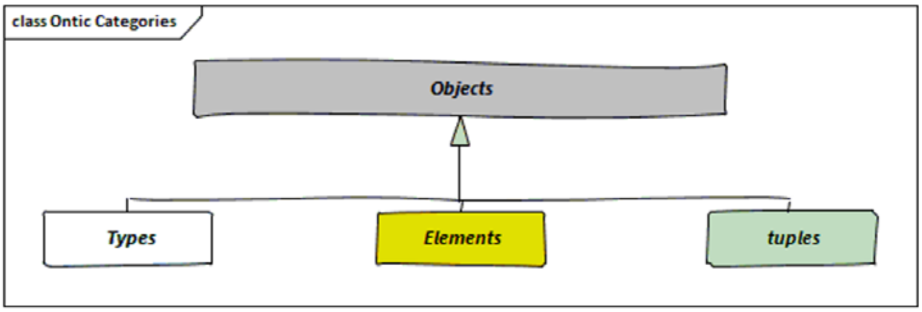
F.3.1 Overview

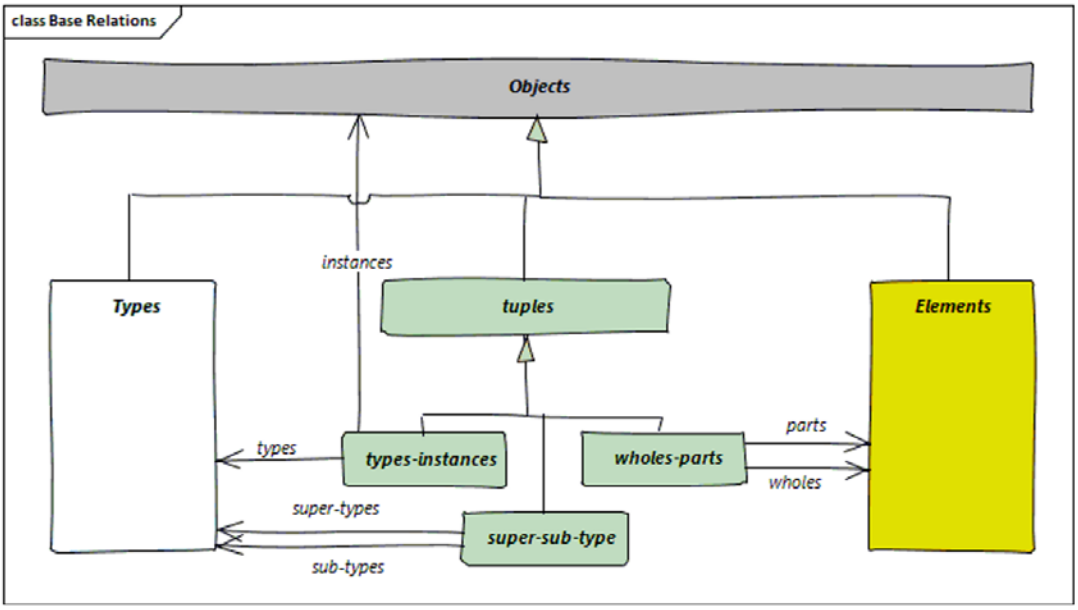
Business Objects Reference Ontology is an upper ontology designed for developing ontological or semantic models for large complex operational applications that consists of a top-level ontology as well as a process for constructing the ontology. It is built upon a series of clear metaphysical choices to provide a solid (metaphysical) foundation. A key choice was for an extensional (and hence, four-dimensional) ontology which provides it with a simple criteria of identity. Elements of it have appeared in a number of standards. For example, the ISO standard, ISO 15926 – Industrial automation systems and integration – was heavily influenced by an early version. The IDEAS (International Defence Enterprise Architecture Specification for exchange) standard is based upon BORO, which in turn was used to develop DODAF 2.0.

From https://en.wikipedia.org/wiki/BORO

See also: https://www.borosolutions.net/

F.3.2 Top-level





F.3.3 Key characteristics

BORO is a well-documented heavyweight foundational ontology. It is an extensional ontology with a general unifying approach – illustrated in the journey in Figure 29.

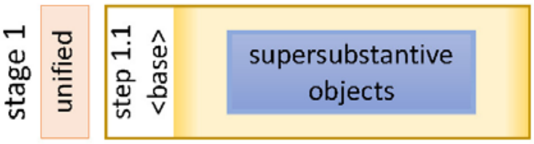


Figure 29 – BORO Stratification Journey – one stratum

Unlike most other TLOs BORO has a position on the Indexicals for 'here' and 'now'. See Business objects: re-engineering for re-use. Chapter 8 – Section 4 (Partridge, 1996) – The time-based 'consciousness' of information systems – which discusses a 'now' and 'here' object. In a later paper, a more sophisticated way of handling indexicality using agentology is described (Partridge, 2018). The paper suggests that there is an agentology layer indexed to the agent/system under the ontology.

Unlike most other TLOs it has a clearly documented position on the unstratified type-instance hierarchy. See Developing an Ontological Sandbox: Investigating Multi-Level Modelling's Possible Metaphysical Structures (Partridge, 2017) and Coordinate Systems: Level Ascending Ontological Options for a detailed discussion (Partridge, 2019).

F.3.4 Relevant extracts

These extracts from: BORO as a Foundation to Enterprise Ontology - https://www.academia.edu/33717627/ – for references see the original document.

BORO includes a foundational (or upper) ontology and a closely intertwined methodology for information systems (IS) re-engineering (Partridge, 1996), hence the term BORO refers to both the ontology and the methodology. BORO was originally conceived in the late 1980s to address a particular need for a solid legacy re-engineering process and then evolved to address a wider need for developing enterprise systems in a 'better way'; in other words in a way that was less cumbersome, compared to the heavyweight methodologies of the time, enabling higher levels of reuse and, as a consequence, capable of reducing the effort and cost of (re-)developing, maintaining and interoperating enterprise systems. It was eventually publicly documented in (Partridge, 1996).

The BORO Foundational Ontology is strongly rooted in philosophical ontology. Ontology is defined by Jonathan Lowe as "the set of things whose existence is acknowledged by a particular theory or system of thought" (Honderich, 2006, 670). This definition is particularly relevant in the context of enterprise modeling and systems development since it grounds ontology in reality (i.e. "the things whose existence is acknowledged") rather than one's subjective conception of what constitutes the real world. As such BORO is a realist ontology, one that recognizes the existence of an objective reality.

…

In the BUML model of Figure 1 Objects represents the three top level BORO categories: Elements, Types and tuples. Every object belongs to one and only one of the three categories which are framed, as mentioned earlier, by a range of metaphysical choices. These choices mean that, within BORO, each category has its own identity criteria.

- Elements are individual objects whose identity is given by the element's spatiotemporal extent (or extension); i.e. the space and time it occupies. BORO simplifies things by assuming that matter and space-time are identical (this is a metaphysical stance that has been called super-substantivalism (Sklar, 1974; Schaffer, 2009). An example of an element would be the person *John*.
- *Types* are collections of any type of object (in other words, objects of any of the three categories). The identity of a type is determined by its extension, the collection of its instances (i.e. members). For example, the extension of the type *Persons* is the set of all people. In BORO, *Types* play a similar role to universals in other foundational ontologies.
- *Tuples* are relationships between objects. The identity of a tuple is defined by the places in the tuple. An example is (*Mary*, *John*) in which the elements Mary and John occupy places 1 and 2 in the tuple respectively. Tuples can be collected into types, called tuple types. An example is *parentOf*, which is the collection of all relationships between parents and their children. Section 2.3 will describe tuple types and their top level patterns in more detail.

There is a system of ontological dependence relations between these categories. One rather abstract way of developing an understanding of these, and so developing a better understanding of the categories is through grounding (Fine, 2010), which provides a kind of ontogenesis narrative for the objects in the ontology. The grounding (ontogenesis) narrative starts with a single element, the pluriverse of all possible worlds (a position Schaffer (2010) calls ‘priority monism’). Consider the generative operation of decomposition that divides an element into all its parts. If we apply this to the pluriverse we then have all the elements.

This operation exhausts all elements as the pluriverse and its parts are all the elements. Then consider the generative type-builder operation; we can then apply this to the (previously generated) elements to build the type Elements; this is the ontological category of Elements. Then consider the generative operation power-type-builder (power-types are described in more detail below). Apply the powertype-builder operation to the set Elements – this builds the type that has all the subsets of Elements as its members. Applying the power type-builder operation repeatedly builds a type hierarchy. Finally, consider the generative tuple-builder operation, this takes a number of any type of object, including tuples, and organizes them into a tuple. This grounding approach is reflected in the BORO methodology. An example is provided in Partridge (2002a).

F.4 CIDOC

F.4.1 Overview

Although “CIDOC object-oriented Conceptual Reference Model” (CRM) is a domain ontology, specialised to the purposes of representing cultural heritage, a subset called CRM Core is a generic upper ontology, including:

- Space-Time – title/identifier, place, era/period, time-span, relationship to persistent items
- Events – title/identifier, beginning/ending of existence, participants (people, either individually or in groups), creation/modification of things (physical or conceptional), relationship to persistent items
- Material Things – title/identifier, place, the information object the material thing carries, part-of relationships, relationship to persistent items
- Immaterial Things – title/identifier, information objects (propositional or symbolic), conceptional things, part-of relationships

A persistent item is a physical or conceptional item that has a persistent identity recognized within the duration of its existence by its identification rather than by its continuity or by observation. A persistent item is comparable to an endurant.

A propositional object is a set of statements about real or imaginary things.

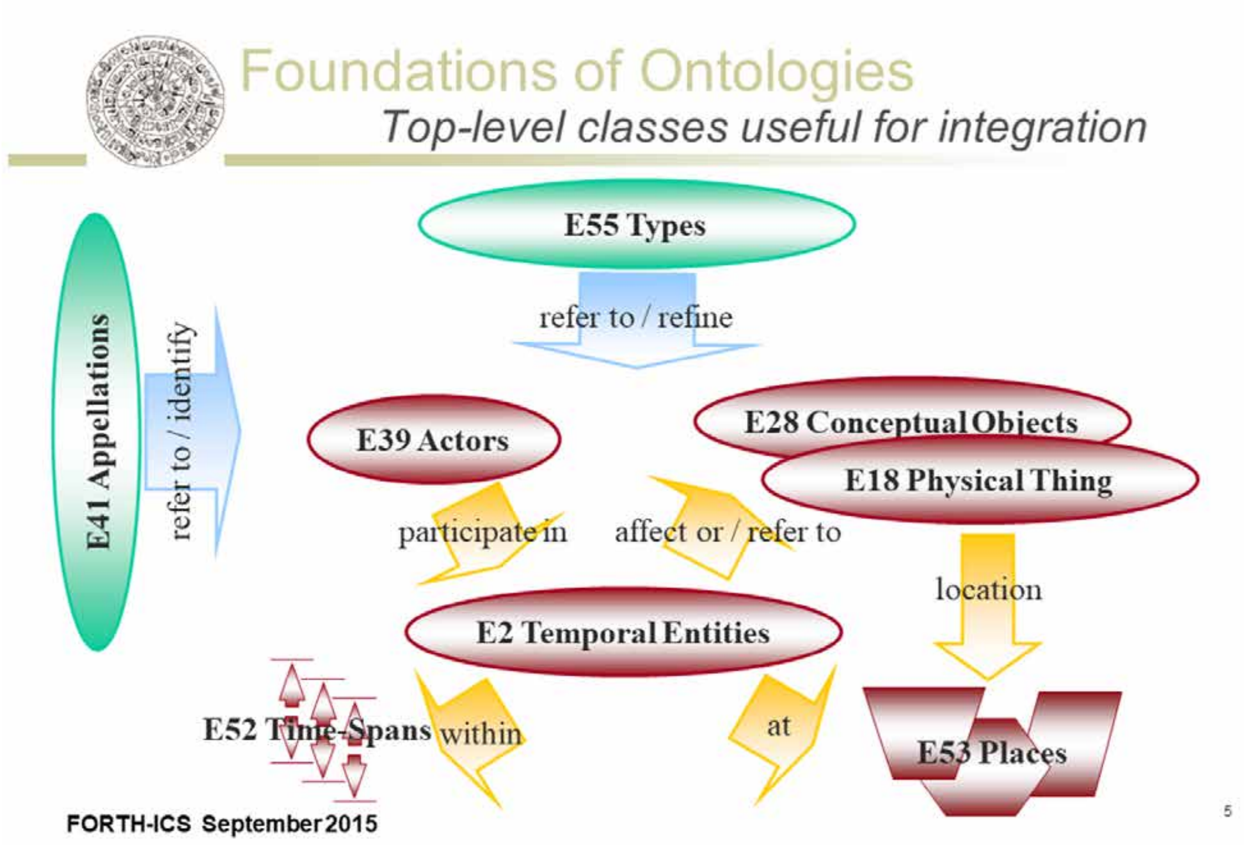
A symbolic object is a sign/symbol or an aggregation of signs or symbols.

From https://en.wikipedia.org/wiki/Upper_ontology#CIDOC_Conceptual_Reference_Model

See also: http://www.cidoc-crm.org/

Also ISO 21127:2014 Information and documentation – A reference ontology for the interchange of cultural heritage information – https://www.iso.org/standard/57832.html

F.4.2 Top-level



Foundations of Ontologies

Top-level classes useful for integration

E55 Types

refer to / refine

E41 Appellations

refer to / identify

E39 Actors

E28 Conceptual Objects

E18 Physical Thing

participate in

affect or / refer to

location

E2 Temporal Entities

E52 Time-Spans

within

at

E53 Places

FORTH-ICS September 2015

5

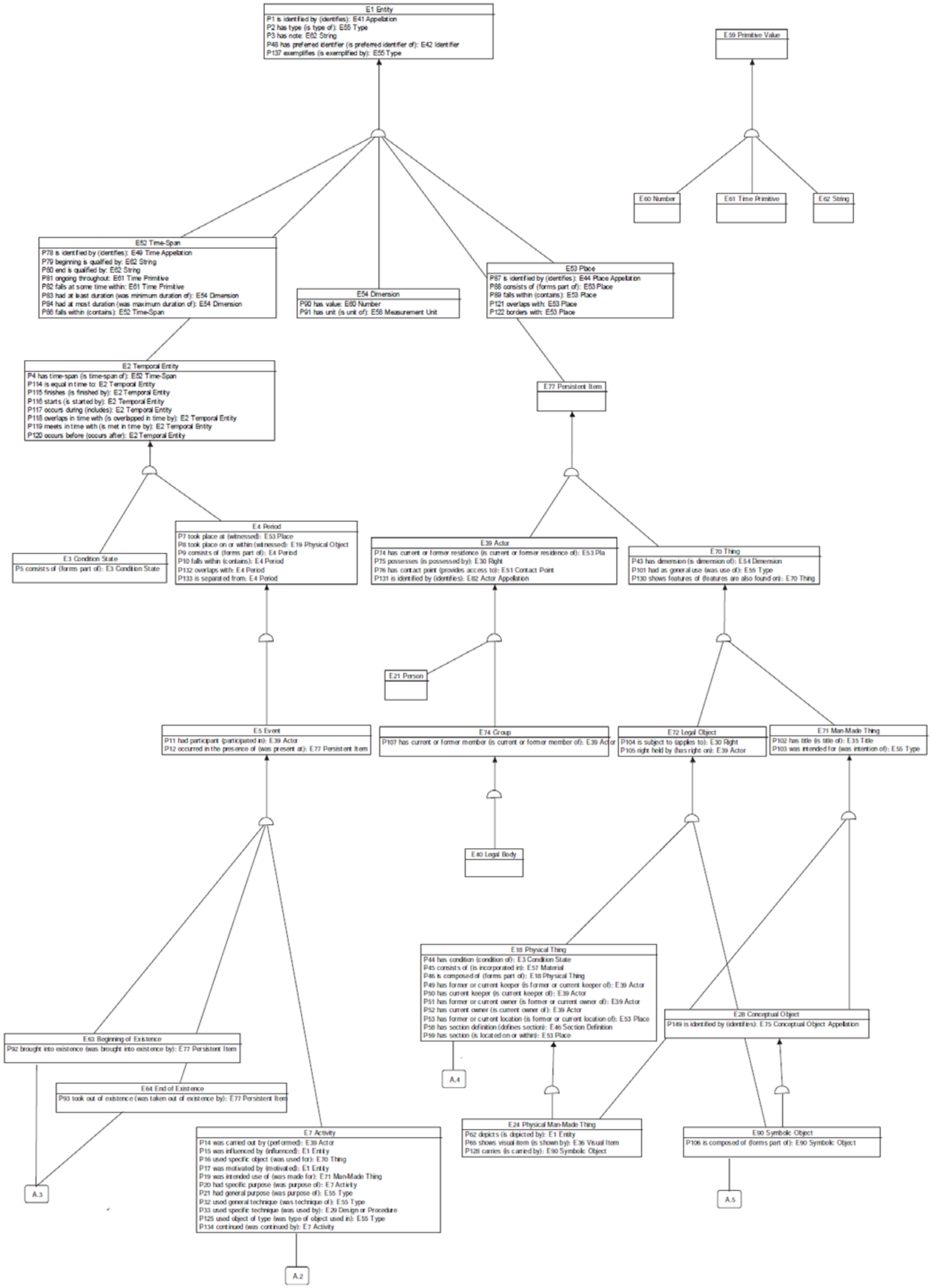


Figure A.1 — E1 Entity

F.4.3 Key characteristics

CIDOC is a lightweight foundational ontology. It does not have much documentation of its ontological commitments.

F.4.4 Relevant extracts

These extracts from: ISO 21127:2014 Information and documentation – A reference ontology for the interchange of cultural heritage information – https://www.iso.org/standard/57832.html.

3.1 – class – category of items that share one or more common traits.

(Hence, intensional criterion of identity)

3.11 – multiple inheritance – possibility for a class to have more than one immediate superclass

E77 Persistent Item – Scope note: This class comprises items that have a persistent identity, sometimes known as “endurants” in philosophy. They can be repeatedly recognized within the duration of their existence by identity criteria rather than by continuity or observation. Persistent Items can be either physical entities, such as people, animals, or things; or conceptual entities, such as ideas, concepts, products of the imagination, or common names. ... The main classes of objects that fall outside the scope of the E77 Persistent Item class are temporal objects such as periods, events and acts, and descriptive properties.

(Hence, endurant stratification)

F.5 CIM

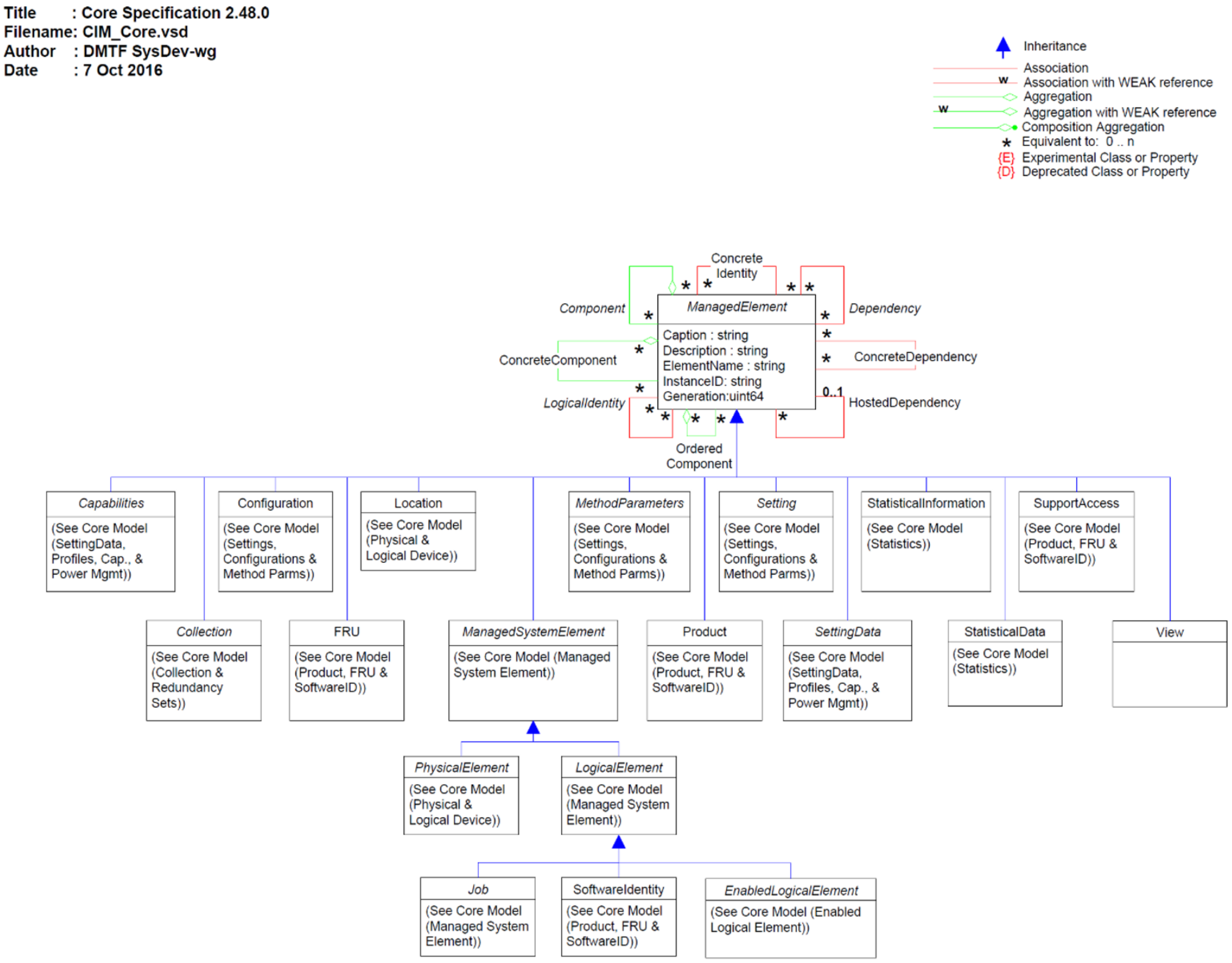
F.5.1 Overview

The Common Information Model (CIM) is an open standard that defines how managed elements in an IT environment are represented as a common set of objects and relationships between them. The Distributed Management Task Force maintains the CIM to allow consistent management of these managed elements, independent of their manufacturer or provider.

From https://en.wikipedia.org/wiki/Common_Information_Model_(computing)

See also:
https://www.dmtf.org/standards/cim%20

F.5.2 Top-level



F.5.3 Key characteristics

CIM is a generic top-level data model. It has few, if any, foundational ontological commitments). It is understandably domain focussed.

F.6 ConML + CHARM – Conceptual Modelling Language and Cultural Heritage Abstract Reference Model

F.6.1 Overview

ConML is a conceptual modelling language that has been constructed from scratch with three major goals in mind:

- Ease of use for non-experts in information technologies.
- Simplicity.
- Expressiveness in complex domains, such as those in the humanities
- Capturing "soft" issues such as temporality, subjectivity and vagueness.

CHARM is a cultural heritage abstract reference model that extends ConML.

From http://www.conml.org/ and http://www.charminfo.org/.

See also: http://www.conml.org/Resources/TechSpec.aspx

F.6.2 Top-level

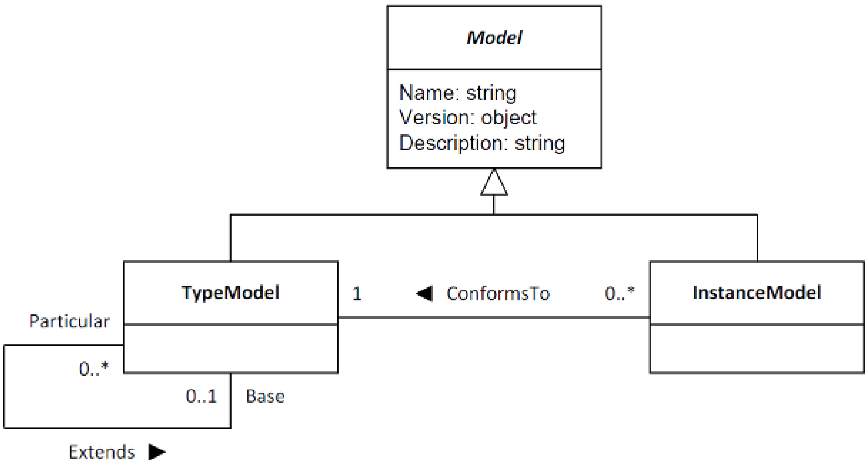


Figure 2. Model-related classes in the ConML metamodel.

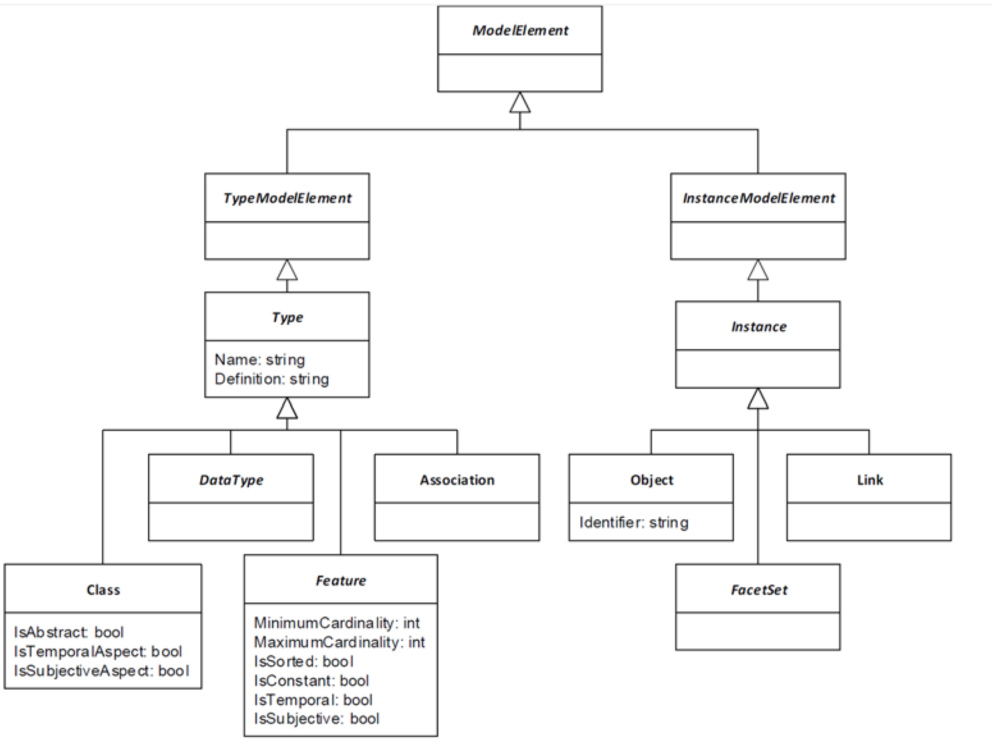


Figure 1. Overview of the major model element types in ConML.

F.6.3 Key characteristics

ConML(+CHARM) is a lightweight foundational ontology. It has few foundational ontological commitments.

The combined ontology is focussed on its domain.

F.7 COSMO – COmmon Semantic MOdel

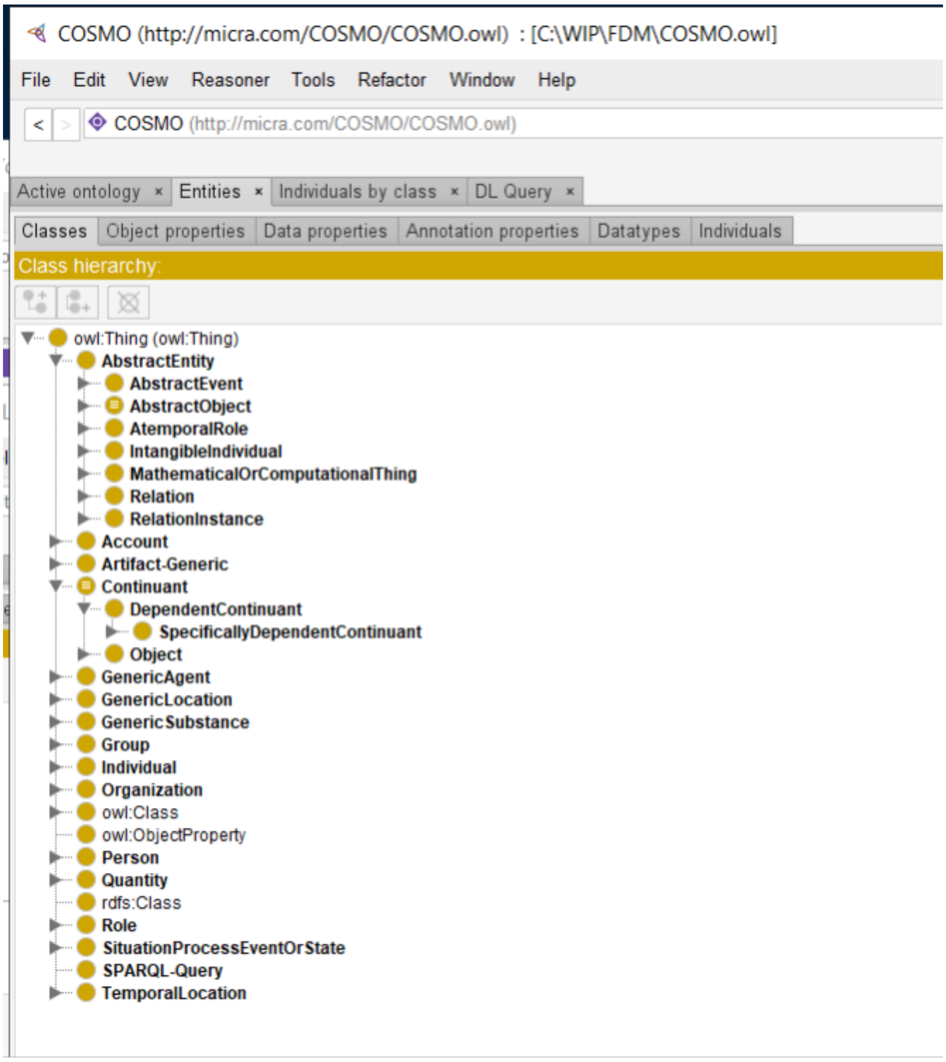
F.7.1 Overview

Developed with the goal of developing a foundation ontology that can serve to enable broad general Semantic Interoperability.

From https://en.wikipedia.org/wiki/Upper_ontology#COSMO.

See also: http://www.micra.com/

F.7.2 Top-level



Taken from http://micra.com/COSMO/COSMO.owl.

F.8 Cyc

F.8.1 Overview

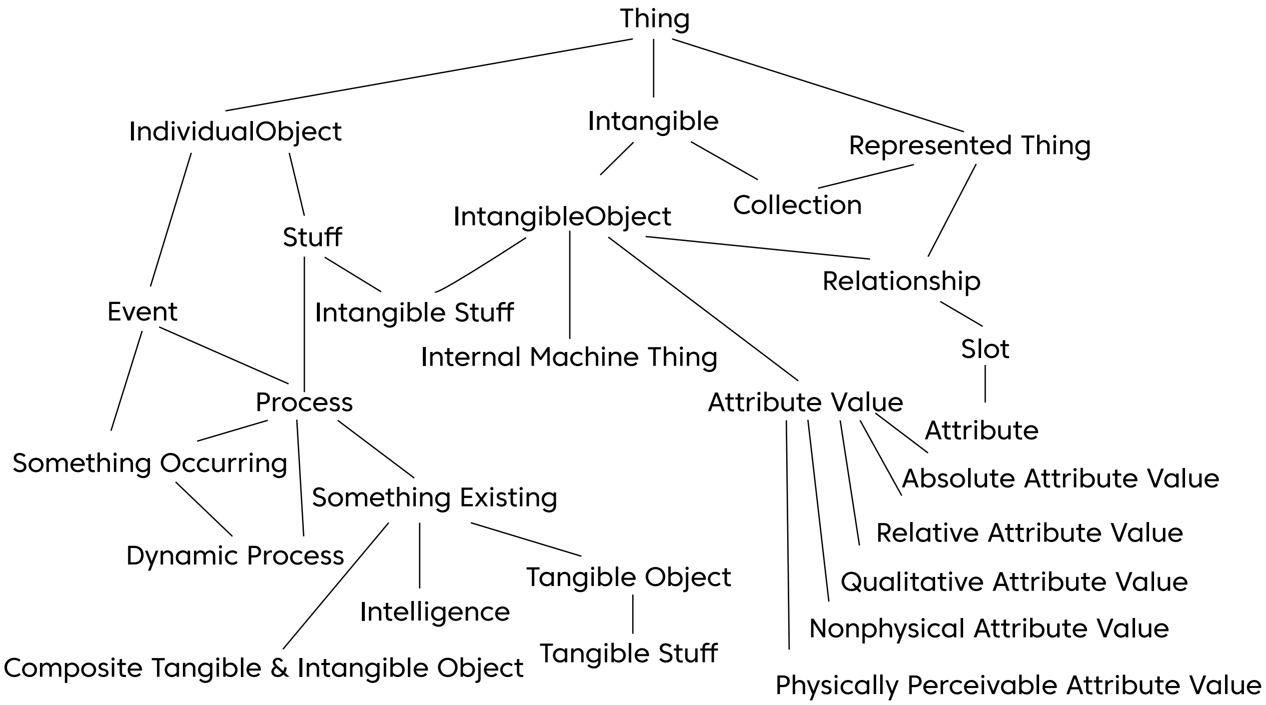
Cyc is a long-term artificial intelligence project that aims to assemble a comprehensive ontology and knowledge base that spans the basic concepts and rules about how the world works. Hoping to capture common sense knowledge, Cyc focuses on implicit knowledge that other AI platforms may take for granted. This is contrasted with facts one might find somewhere on the internet or retrieve via a search engine or Wikipedia. Cyc enables AI applications to perform human-like reasoning and be less “brittle” when confronted with novel situations.

The first version of OpenCyc was released in spring 2002 and contained only 6,000 concepts and 60,000 facts. The knowledge base was released under the Apache License.

From https://en.wikipedia.org/wiki/Cyc

See also: https://www.cyc.com/the-cyc-platform

F.8.2 Top-level



The developers of Cyc did not believe that the top-most levels of the 'ontology' mattered a great deal. They thought the hard work is done lower down. And so, their top-level is very simple – with no real explicit foundational ontological commitments.

Cyc allows multiple inheritance (multiple is-a parents): for example, Intangible Stuff has Intangible Object and Stuff as parents. It uses Collection for higher order types.

F.8.3 Key characteristics

Cyc is a generic TLO. It intentionally has few ontological commitments.

F.9 DC – Dublin Core

F.9.1 Overview

The Dublin Core schema is a small set of vocabulary terms that can be used to describe digital resources (video, images, web pages, etc.), as well as physical resources such as books or CDs, and objects like artworks.

From https://en.wikipedia.org/wiki/Dublin_Core

See also: http://dublincore.org/

F.9.2 Top-level

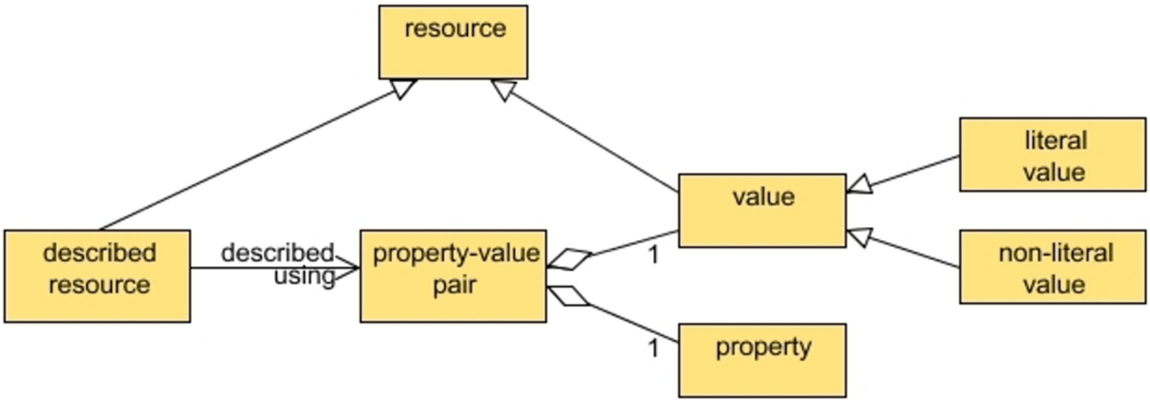


Figure 1 – the DCMI resource model

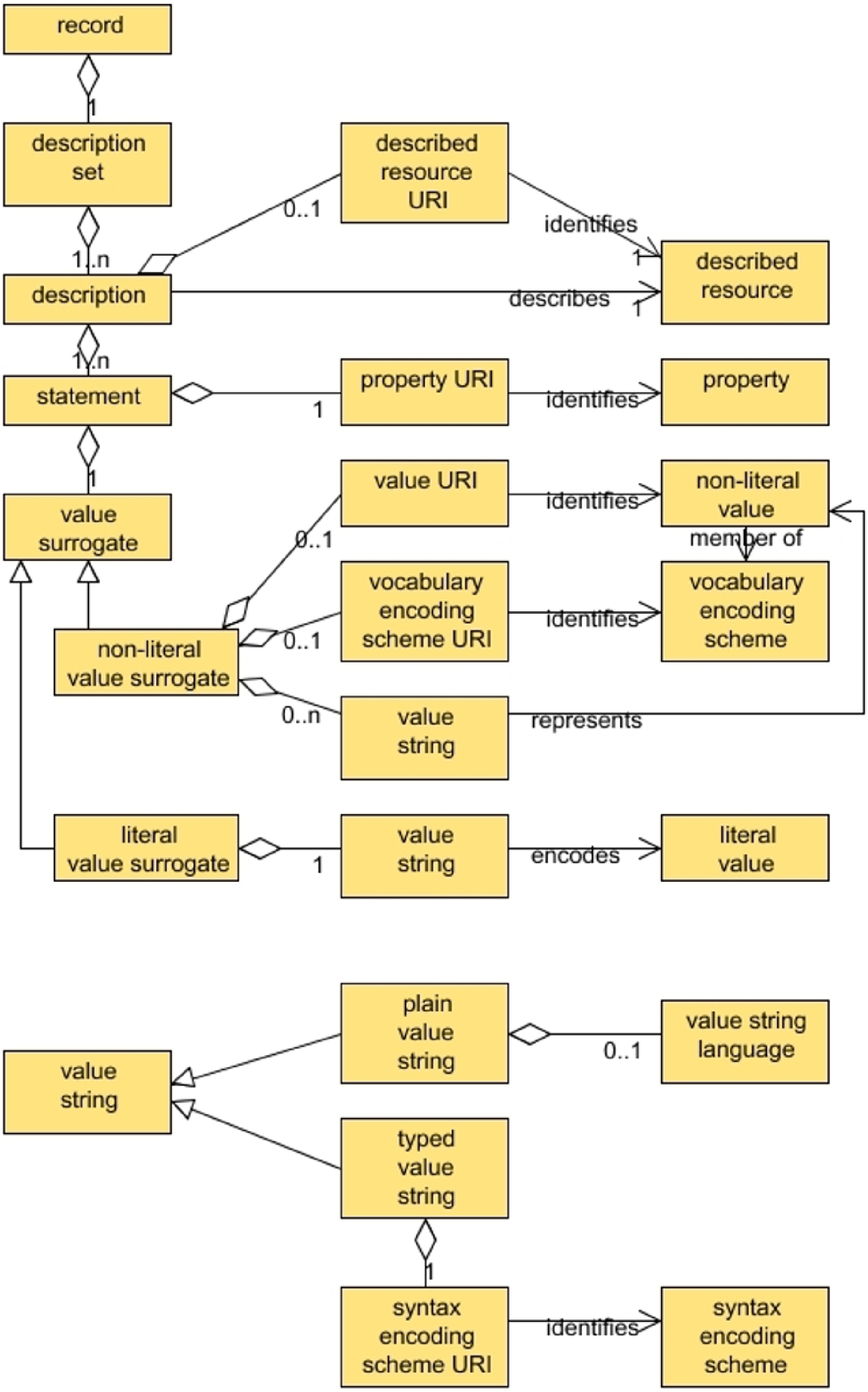


Figure 2 – the DCMI description set model
From https://www.dublincore.org/specifications/dublin-core/abstract-model/.

F.9.3 Key characteristics

Dublin Core is a generic TLO with a focus on digital resources.

F.9.4 Relevant extracts

From https://www.dublincore.org/specifications/dublin-core/abstract-model/

Extract 1 – The DCMI Vocabulary Model

2.3 The DCMI Vocabulary Model

The abstract model of the vocabularies used in DC metadata descriptions is as follows:

- A vocabulary is a set of one or more terms. Each term is a member of one or more vocabularies.
- A term is a property (element), class, vocabulary encoding scheme, or syntax encoding scheme.
- Each property may be related to one or more classes by a has domain relationship. Where it is stated that a property has such a relationship with a class and the property is part of a property/value pair, it follows that the described resource is an instance of that class.
- Each property may be related to one or more classes by a has range relationship. Where it is stated that a property has such a relationship with a class and the property is part of a property/value pair, it follows that the value is an instance of that class.
- Each resource may be an instance of one or more classes.
- Each resource may be a member of one or more vocabulary encoding schemes.

Each class may be related to one or more other classes by a sub-class of relationship (where the two classes are defined such that all resources that are instances of the sub-class are also instances of the related class).

Each property may be related to one or more other properties by a sub-property of relationship. Where it is stated that such a relationship exists, the two properties are defined such that whenever the sub-property is part of a property/value pair describing a resource, it follows that the resource is also described using a second property/value pair made up of the property and the value.

Each syntax encoding scheme is a class (of literals).

Note that the word “vocabulary” is used here to refer specifically to a set of terms, a set in which the members are properties (elements), classes, vocabulary encoding schemes, and/or syntax encoding schemes.

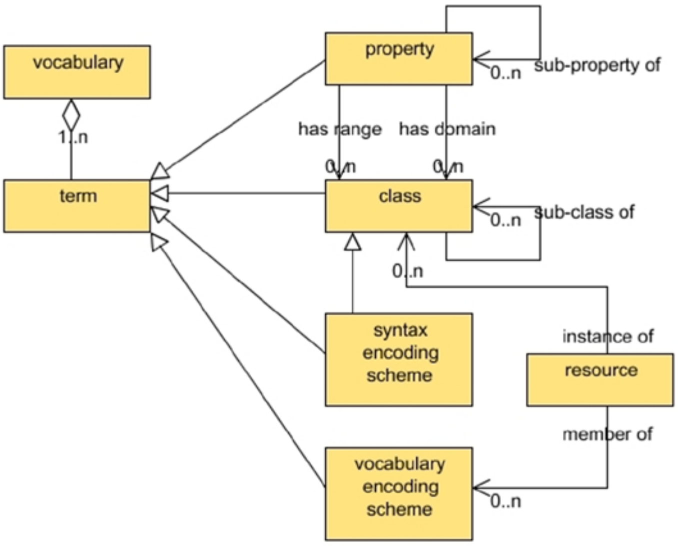


Figure 3 – the DCMI vocabulary model

F.10 DOLCE – Descriptive Ontology for Linguistic and Cognitive Engineering

F.10.1 Overview

Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) is a foundational ontology designed in 2002 in the context of the WonderWeb EU project, developed by Nicola Guarino and his associates at the Laboratory for Applied Ontology (LOA). As implied by its acronym, DOLCE is oriented toward capturing the ontological categories underlying natural language and human common sense. DOLCE, however, does not commit to a strictly referentialist metaphysics related to the intrinsic nature of the world. Rather, the categories it introduces are thought of as cognitive artifacts, which are ultimately depending on human perception, cultural inprints, and social conventions. In this sense, they intend to be just descriptive (vs prescriptive) notions, which support the formal specification of domain conceptualizations.

From https://en.wikipedia.org/wiki/Upper_ontology#DOLCE

See also: http://www.loa.istc.cnr.it/dolce/overview.html

F.10.2 Top-level

IST Project 2001-33052 WonderWeb:
Ontology Infrastructure for the Semantic Web

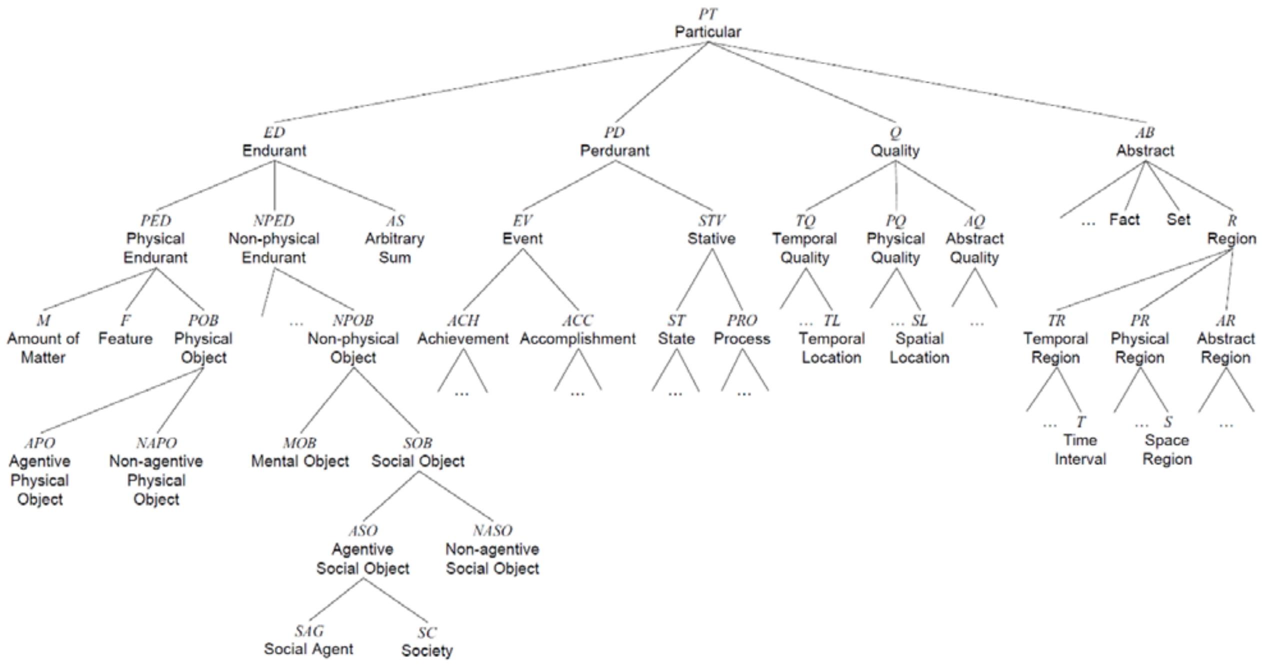


Figure 2: Taxonomy of DOLCE basic categories.

The top object is labelled 'Particular' indicating that all instances of this and its sub-types are particulars. One implication of this is that the ontology is first order – that there are no higher order ontologies.

F.10.3 Key characteristics

DOLCE is a well-documented heavyweight natural language ontology aiming to capture the ontological categories underlying natural language and human common sense.

F.11 EMMO

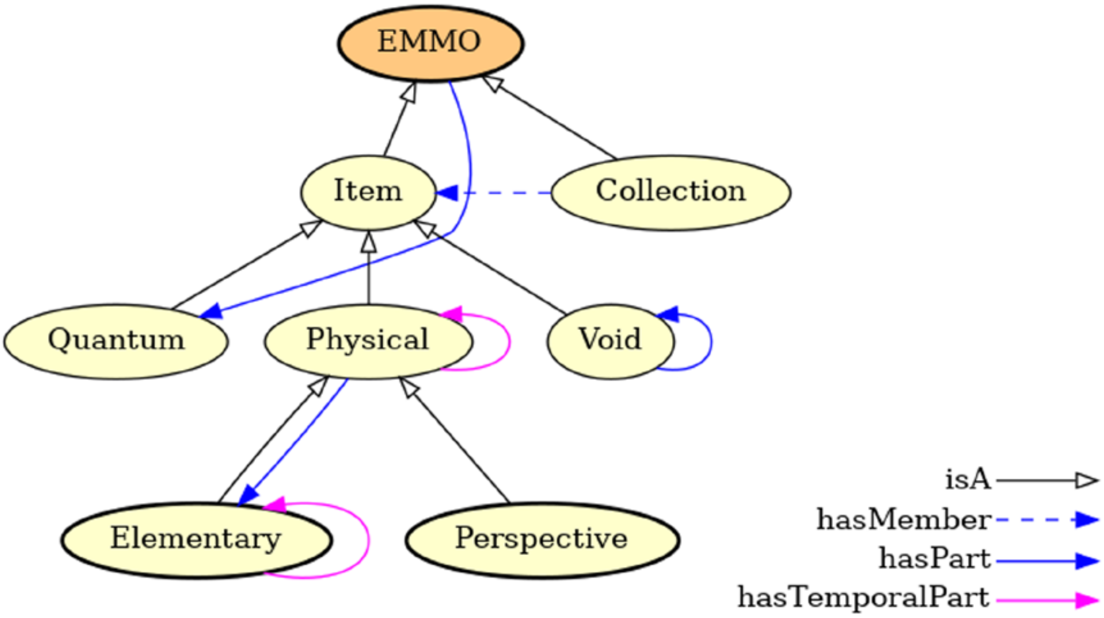
F.11.1 Overview

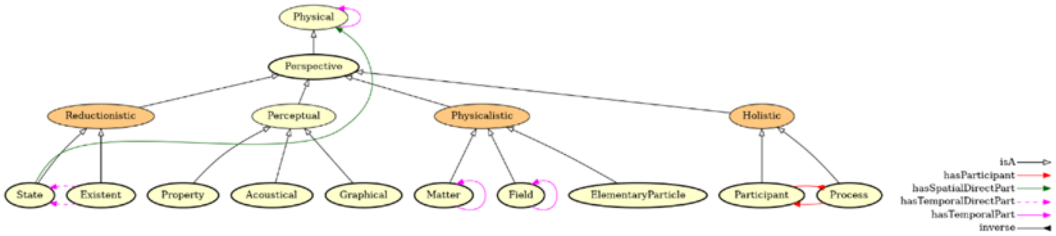
The EMMO top-level is the group of fundamental axioms that constitute the philosophical foundation of the EMMO. Adopting a physicalistic/nominalistic perspective, the EMMO defines real world objects as 4D objects that are always extended in space and time (i.e. real-world objects cannot be spaceless nor timeless). For this reason, abstract objects, i.e. objects that do not extend in space and time, are forbidden in the EMMO. It has been instigated by materials science and provides the connection between the physical world, the experimental world (materials characterisation) and the simulation world (materials modelling).

From https://github.com/emmo-repo/EMMO

See also: https://materialsmodelling.com/2019/06/14/european-materials-modelling-ontology-emmo-release/

F.11.2 Top-level





F.11.3 Key characteristics

This is light-weight ontology.

F.11.4 Relevant extracts

Extracted from https://github.com/emmo-repo/EMMO

The *Reductionistic* perspective class uses the fundamental non-transitive parthood relation, called direct parthood, to provide a powerful granularity description of multiscale real world objects. The EMMO can in principle represents the Universe with direct parthood relations as a direct rooted tree up to its elementary constituents.

The *Holistic* perspective class introduces the concept of real world objects that unfold in time in a way that has a meaning for the EMMO user, through the definition of the classes Process and Participant.

The *Phenomenic* perspective class introduces the concept of real world objects that express of a recognisable pattern in space or time that impress the user. Under this class the EMMO categorises e.g. formal languages, pictures, geometry, mathematics and sounds. Phenomenic objects can be used in a semiotic process as signs.

The *Physics* perspective class introduces the concept of real world objects that have a meaning for the under applied physics perspective.

The *semiotics* module introduces the concepts of semiotics and the Semiosis process that has a Sign, an Object and an Interpreter as participants. This forms the basis in EMMO to represent e.g. models, formal languages, theories, information and properties.

EMMO relations

All EMMO relations are subrelations of the relations found in the two roots: mereotopological and semiotical. The relation hierarchy extends more vertically (i.e. more subrelations) than horizontally (i.e. less sibling relations), facilitating the categorisation and inferencing of individuals.

Imposing all relations to fall under mereotopology or semiotics is how the EMMO force the developers to respect its perspectives. Two entities are related only by contact or parthood (mereotopology) or by standing one for another (semiosis): no other types of relation are possible within the EMMO.

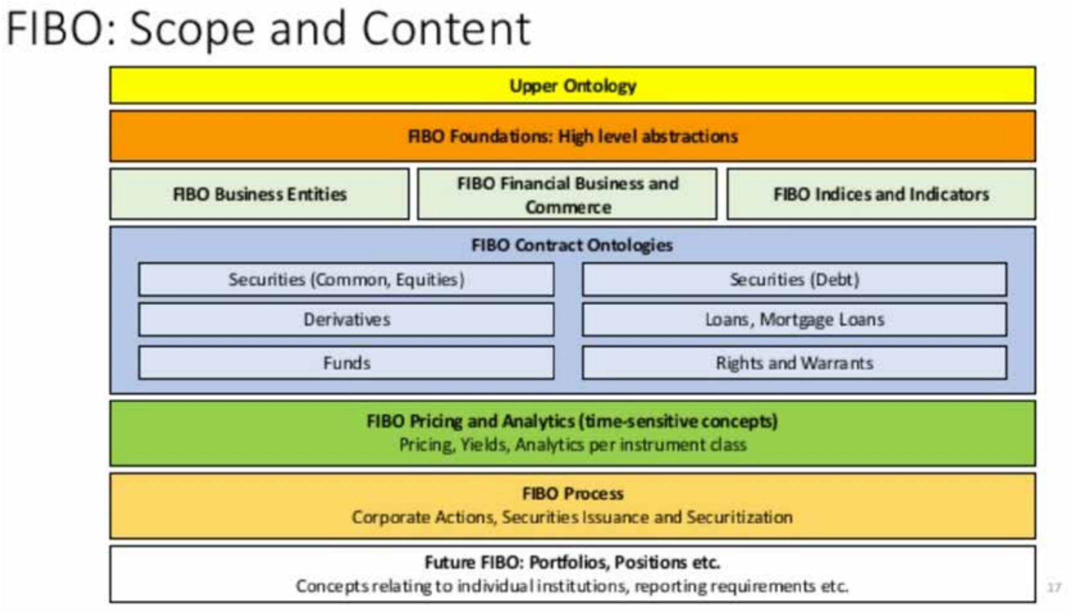
F.12 FIBO – Financial Industry Business Ontology

F.12.1 Overview

The Financial Industry Business Ontology (FIBO) defines the sets of things that are of interest in financial business applications and the ways that those things can relate to one another.

From https://spec.edmcouncil.org/fibo/

F.12.2 Top-level



FIBO: Scope and Content

Upper Ontology

FIBO Foundations: High level abstractions

FIBO Business Entities

FIBO Financial Business and Commerce

FIBO Indices and Indicators

FIBO Contract Ontologies

Securities (Common, Equities)

Securities (Debt)

Derivatives

Loans, Mortgage Loans

Funds

Rights and Warrants

FIBO Pricing and Analytics (time-sensitive concepts)
Pricing, Yields, Analytics per instrument class

FIBO Process
Corporate Actions, Securities Issuance and Securitization

Future FIBO: Portfolios, Positions etc.
Concepts relating to individual institutions, reporting requirements etc.

F.12.3 Key characteristics

We were unable to find adequate resources to assess this TLO.

F.13 FrameNet

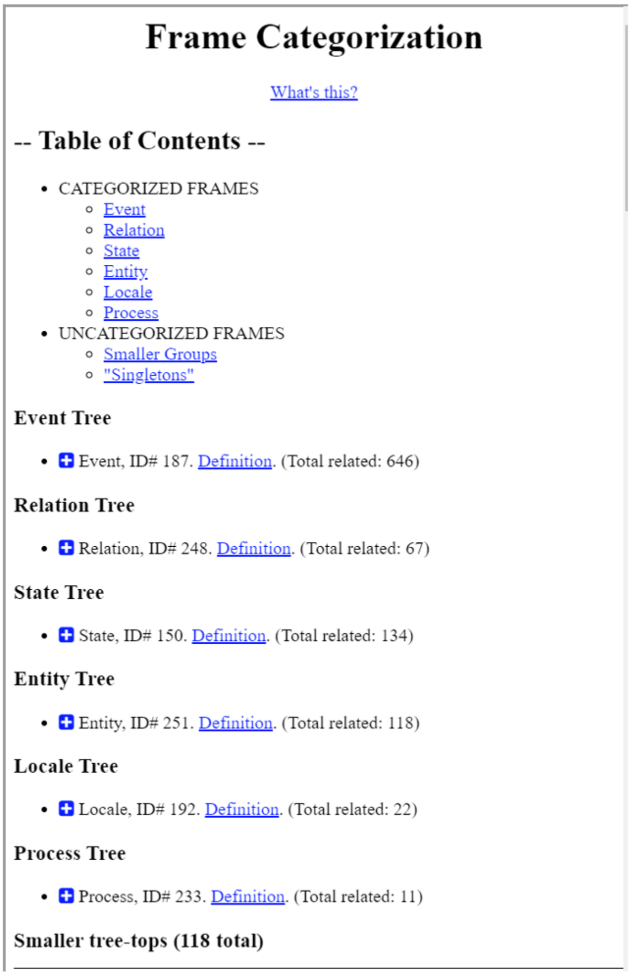
F.13.1 Overview

In computational linguistics, FrameNet is a project housed at the International Computer Science Institute in Berkeley, California which produces an electronic resource based on a theory of meaning called frame semantics. FrameNet reveals for example that the sentence "John sold a car to Mary" essentially describes the same basic situation (semantic frame) as "Mary bought a car from John", just from a different perspective. A semantic frame can be thought of as a conceptual structure describing an event, relation, or object and the participants in it. The FrameNet lexical database contains over 1,200 semantic frames, 13,000 lexical units (a pairing of a word with a meaning; polysemous words are represented by several lexical units) and 202,000 example sentences. FrameNet is largely the creation of Charles J. Fillmore, who developed the theory of frame semantics that the project is based on, and was initially the project leader when the project began in 1997.

From https://en.wikipedia.org/wiki/FrameNet

See also: https://framenet.icsi.berkeley.edu/fndrupal/

F.13.2 Top-level



Frame Categorization

What's this?

-- Table of Contents --

- CATEGORIZED FRAMES
    - Event
    - Relation
    - State
    - Entity
    - Locale
    - Process
- UNCATEGORIZED FRAMES
    - Smaller Groups
    - "Singletons"

Event Tree

- ⊞ Event, ID# 187. Definition. (Total related: 646)

Relation Tree

- ⊞ Relation, ID# 248. Definition. (Total related: 67)

State Tree

- ⊞ State, ID# 150. Definition. (Total related: 134)

Entity Tree

- ⊞ Entity, ID# 251. Definition. (Total related: 118)

Locale Tree

- ⊞ Locale, ID# 192. Definition. (Total related: 22)

Process Tree

- ⊞ Process, ID# 233. Definition. (Total related: 11)

Smaller tree-tops (118 total)

FrameNet has noted several broad categories, including Event, Relation, State, Entity, Locale, and Process. Many frames inherit from these “top-level” categories, and from those inherited frames, many frames are related via relationships such as Using, Precedes, Subframe, etc. Further effort has extracted potential “top-level” frames which do not inherit from any other frames. These potential “top-level” frames (and all related frames) have been gathered as smaller groups. Finally, frames which neither inherit nor have inheritors are listed as “Singletons”.

https://framenet.icsi.berkeley.edu/fndrupal/FrameLatticeList

F.13.3 Key characteristics

A natural language ontology.

F.14 GFO – General Formal Ontology

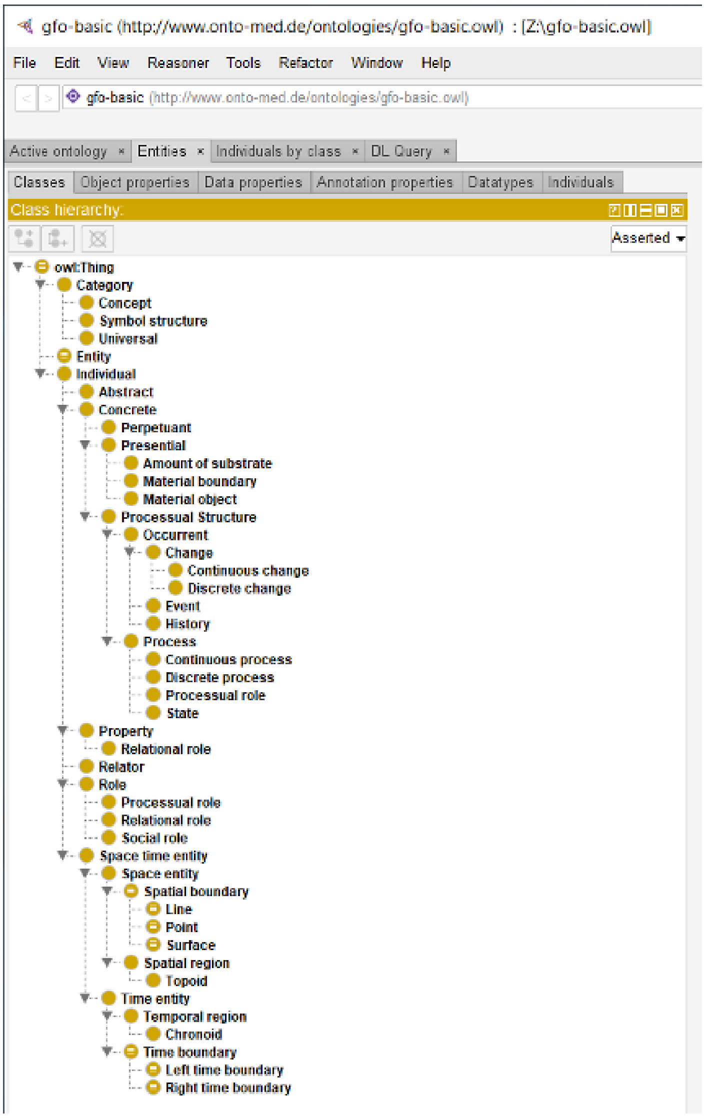
F.14.1 Overview

Realistic ontology integrating processes and objects. It attempts to include many aspects of recent philosophy, which is reflected both in its taxonomic tree and its axiomatizations.

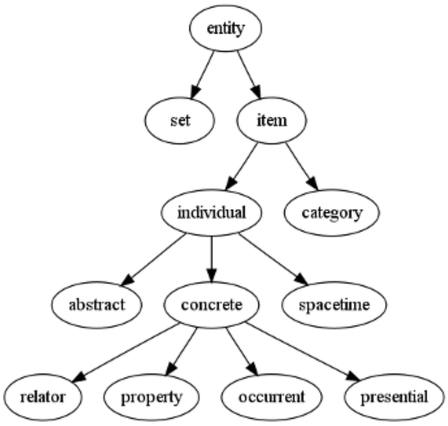
From https://en.wikipedia.org/wiki/Upper_ontology#General_Formal_Ontology_(GFO)

See also: https://www.onto-med.de/ontologies/gfo, https://en.wikipedia.org/wiki/General_formal_ontology

F.14.2 Top-level



- owl:Thing
  - Category
    - Concept
    - Symbol structure
    - Universal
  - Entity
  - Individual
    - Abstract
    - Concrete
      - Perpetuant
      - Presential
        - Amount of substrate
        - Material boundary
        - Material object
      - Processual Structure
        - Occurrent
          - Change
            - Continuous change
            - Discrete change
          - Event
          - History
        - Process
          - Continuous process
          - Discrete process
          - Processual role
          - State
    - Property
      - Relational role
    - Relator
    - Role
      - Processual role
      - Relational role
      - Social role
    - Space time entity
      - Space entity
        - Spatial boundary
          - Line
          - Point
          - Surface
        - Spatial region
          - Topoid
      - Time entity
        - Temporal region
          - Chronoid
        - Time boundary
          - Left time boundary
          - Right time boundary



F.14.3 Key characteristics

GFO is a well-documented heavyweight foundational ontology.

F.14.4 Relevant Extracts

From General Formal Ontology (GFO) – Part I: Basic Principles – Version 1.0 – No. 8 – July 2006

Extract 1 – Higher order

14.3 Instantiation and Categories

... Since we assume categories of arbitrary (finite) type, there can be arbitrarily long (finite) chains of iteration of the instantiation relation.

Extract 2 – First order – apart from one exception – persistants, a special category of second order

3.4 Basic Level

The basic level of GFO contains all relevant top-level distinctions and categories. One should distinguish between primitive categories (whose instances are individuals), and higher order categories. In the present document we consider primitive categories and the category of persistants (which is a special category of second order). These categories will be extended in the future using a number of non-primitive categories. Primitive categories and persistants of the basic level will be discussed further in the following sections and are the main content of the current report.

F.15 gist

F.15.1 Overview

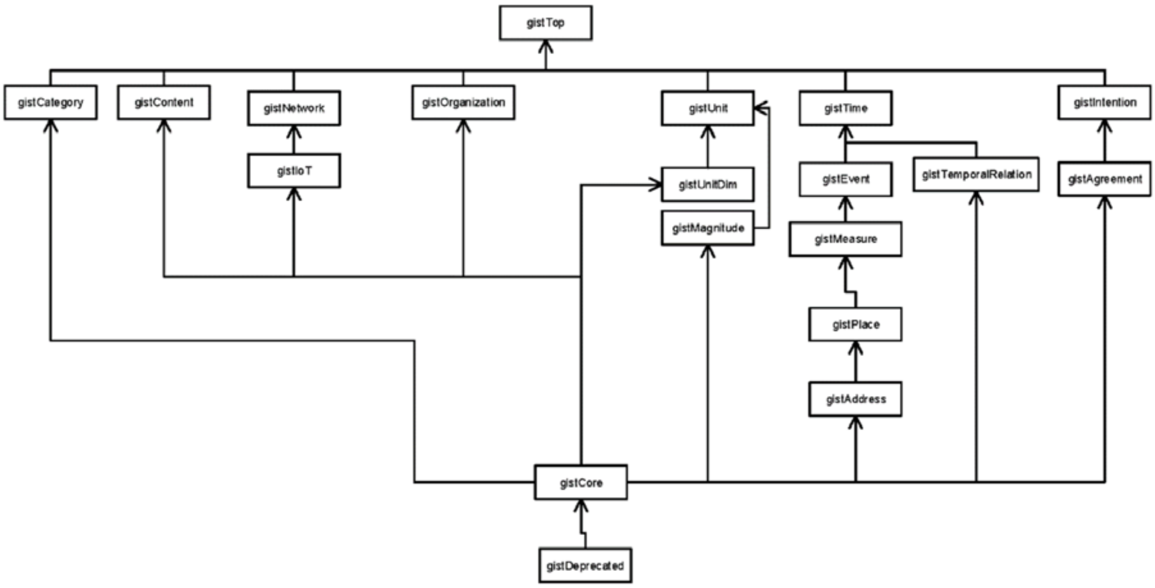
gist is developed and supported by Semantic Arts. gist (not an acronym – it means to get the essence of) is a “minimalist upper ontology”. gist is targeted at enterprise information systems, although it has been applied to healthcare delivery applications. The major attributes of gist are:

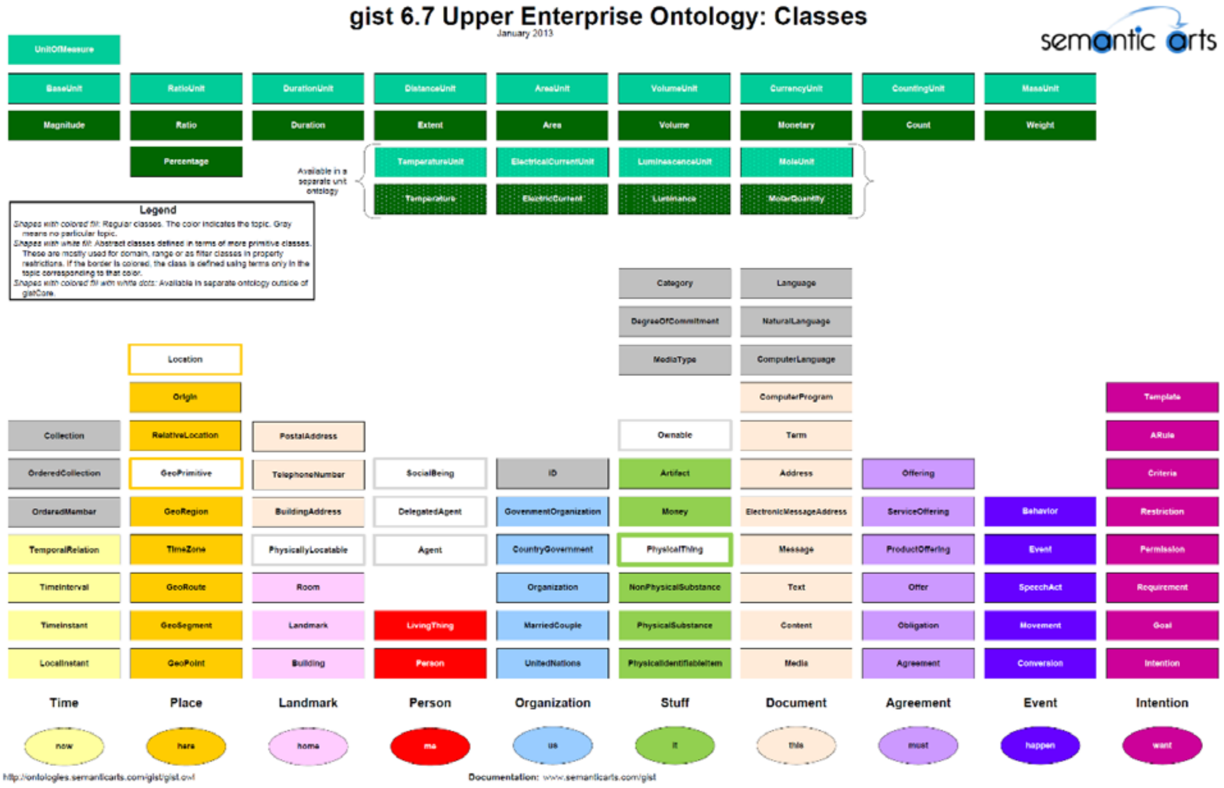
- it is small (there are 140 classes and 127 properties)
- it is comprehensive (most enterprises will not find the need to create additional primitive classes, but will find that most of their classes can be defined and derived from gist)
- it is robust – all the classes descend from 12 primitive classes, which are mostly mutually disjoint. This aids a great deal in subsequent error detection. There are 1342 axioms, and it uses almost all of the DL constructs (it is SROIQ(D))
- it is concrete – most upper ontologies start with abstract philosophical concepts that users must commit to in order to use the ontology. Gist starts with concrete classes that most people already do, or reasonably could agree with, such as Person, Organization, Document, Time, UnitOfMeasure and the like)
- it is unambiguous – ambiguous terms (such as “term”) have been removed as they are often overloaded and confused. Also terms that frequently have different definitions at different enterprises (such as customer and order) have been removed, also to reduce ambiguity.
- it is understandable – in addition to being built on concrete, generally understood primitives, it is extremely modular. The 140 classes are implemented in 18 modular ontologies, each can easily be understood in its entirety, and each imports only the other modules that it needs.

From https://en.wikipedia.org/wiki/Upper_ontology#gist

See also: https://www.semanticarts.com/gist/

F.15.2 Top-level





Legend

Shapes with colored fill: Regular classes. The color indicates the topic. Gray means no particular topic.

Shapes with white fill: Abstract classes defined in terms of more primitive classes. These are mostly used for domain, range or as filter classes in property restrictions. If the border is colored, the class is defined using terms only in the topic corresponding to that color.

Shapes with colored fill with white dots: Available in separate ontology outside of gistCore.

F.15.3 Key characteristics

gist is a generic TLO. It clearly states it intentionally has few ontological commitments.

F.15.4 Relevant Extracts

Extract 1 – Avoids “abstract philosophical concepts”

“it is concrete – most upper ontologies start with abstract philosophical concepts that users must commit to in order to use the ontology. Gist starts with concrete classes that most people already do, or reasonably could agree with, such as Person, Organization, Document, Time, UnitOfMeasure and the like)”

Extract 2 – “Gist has extensive and fine grained disjointness at the highest level.”

“Gist has a small number of top level concepts from which everything else derives. And these concepts are not philosophical abstractions like endurants and perdurants, or qualia, they are normal terms whose definitions are quite close to what you already believe.

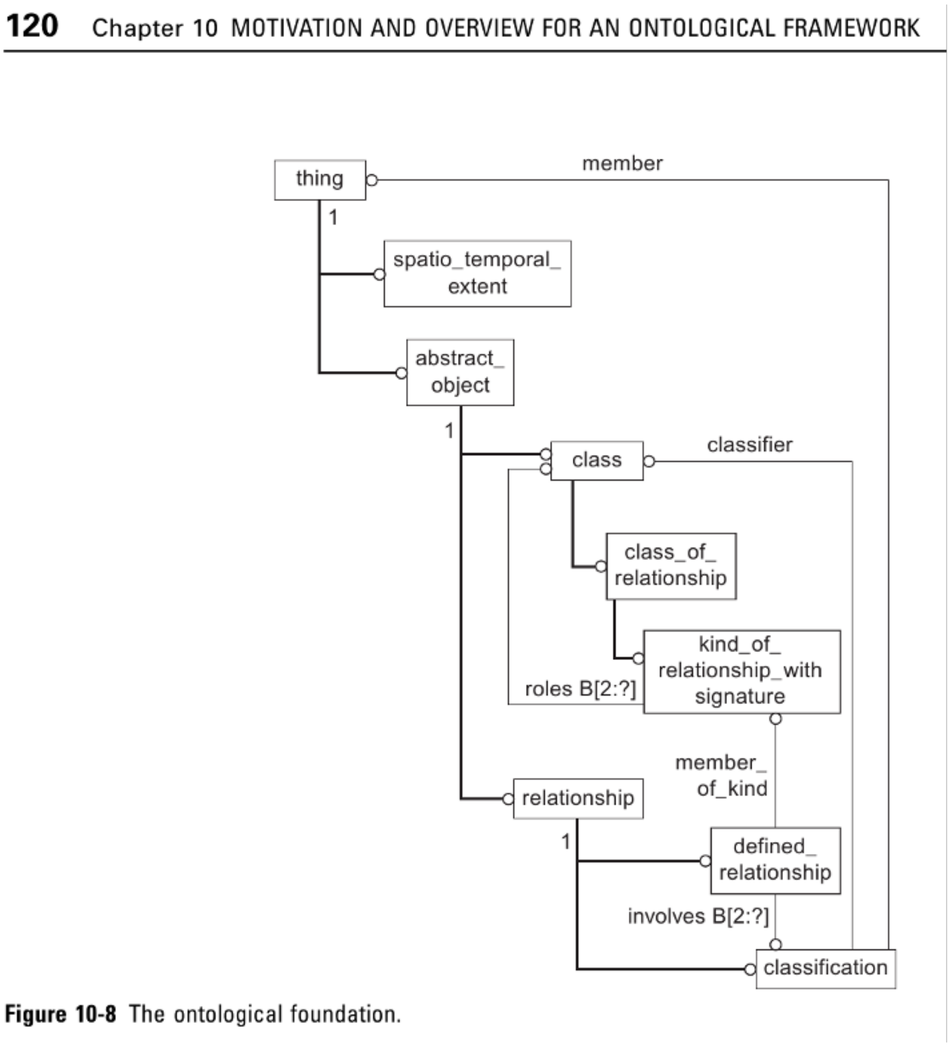
Gist has extensive and fine grained disjointness at the highest level. It turns out that in order for an upper ontology to help you avoid making logical errors in your derived enterprise or application ontology, it needs to make use of disjointness. Without disjointness, the reasoner does not find logic errors.”

F.16 HQDM – High Quality Data Models

F.16.1 Overview

The High Quality Data Models (HQDM) Framework is a four-dimensional top-level ontology with extensional identity criteria that aims to support large scale data integration. As such it aims to ensure there is consistency among data created using the framework. The HQDM Framework is based on work developing and using ISO 15926 and lessons learnt from BORO, which influenced ISO 19526-2.

F.16.2 Top-level



120 Chapter 10 MOTIVATION AND OVERVIEW FOR AN ONTOLOGICAL FRAMEWORK

Figure 10-8 The ontological foundation.

F.16.3 Key characteristics

This is a well-documented heavyweight foundational ontology. It is an extensional ontology with a general unifying approach – illustrated in the journey in Figure 30. It draws heavily on ISO 15926-2 – and shares some of its technical background.

It introduces some novel ideas – such as interpreting possible worlds as branching.

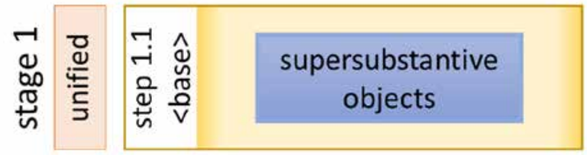


Figure 30 – HQDM Stratification Journey – one stratum

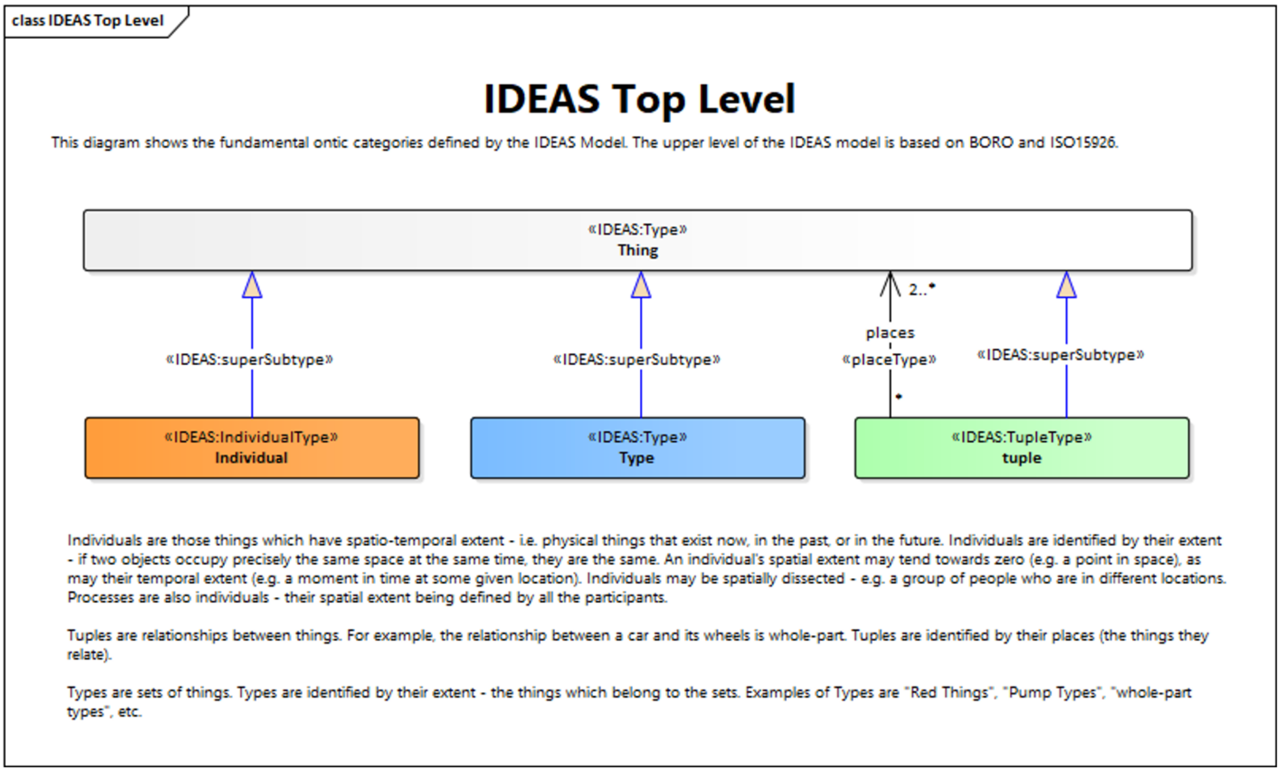
F.17 IDEAS – International Defence Enterprise Architecture Specification

F.17.1 Overview

The upper ontology developed by the IDEAS Group is higher-order, extensional and 4D. It was developed using the BORO Method. The IDEAS ontology is not intended for reasoning and inference purposes; its purpose is to be a precise model of business.

From https://en.wikipedia.org/wiki/IDEAS_Group

F.17.2 Top-level



IDEAS Top Level

This diagram shows the fundamental ontic categories defined by the IDEAS Model. The upper level of the IDEAS model is based on BORO and ISO15926.

Individuals are those things which have spatio-temporal extent - i.e. physical things that exist now, in the past, or in the future. Individuals are identified by their extent - if two objects occupy precisely the same space at the same time, they are the same. An individual's spatial extent may tend towards zero (e.g. a point in space), as may their temporal extent (e.g. a moment in time at some given location). Individuals may be spatially dissected - e.g. a group of people who are in different locations. Processes are also individuals - their spatial extent being defined by all the participants.

Tuples are relationships between things. For example, the relationship between a car and its wheels is whole-part. Tuples are identified by their places (the things they relate).

Types are sets of things. Types are identified by their extent - the things which belong to the sets. Examples of Types are "Red Things", "Pump Types", "whole-part types", etc.

F.17.3 Key characteristics

This is a well-documented heavyweight foundational ontology. It is an extensional ontology with a general unifying approach – illustrated in the journey in Figure 31. It is largely based upon BORO.

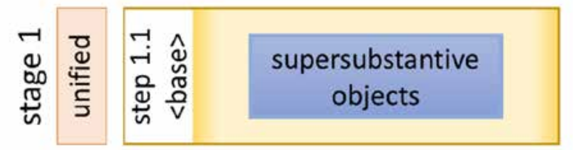


Figure 31 – IDEAS Stratification Journey – one stratum

F.18 IEC 62541

F.18.1 Overview

OPC Unified Architecture (OPC UA) is a machine to machine communication protocol for industrial automation developed by the OPC Foundation.

- Focus on communicating with industrial equipment and systems for data collection and control
- Open – freely available and implementable under GPL 2.0 license
- Cross-platform – not tied to one operating system or programming language
- Service-oriented architecture (SOA)
- Inherent complexity – the specification consists of 1250 pages in 14 documents
- Offers security functionality for authentication, authorization, integrity and confidentiality
- Integral information model, which is the foundation of the infrastructure necessary for information integration where vendors and organizations can model their complex data into an OPC UA namespace to take advantage of the rich service-oriented architecture of OPC UA. There are over 35 collaborations with the OPC Foundation currently. Key industries include pharmaceutical, oil and gas, building automation, industrial robotics, security, manufacturing and process control.

From https://en.wikipedia.org/wiki/OPC_Unified_Architecture

See also: https://opcfoundation.org/developer-tools/specifications-unified-architecture

F.18.2 Top-level

ISO Standard document not available.

F.19 IEC 63088

F.19.1 Overview

IEC PAS 63088:2017(E) describes a reference architecture model in the form of a cubic layer model, which shows technical objects (assets) in the form of layers, and allows them to be described, tracked over their entire lifetime (or “vita”) and assigned to technical and/or organizational hierarchies. It also describes the structure and function of Industry 4.0 components as essential parts of the virtual representation of assets.

From https://webstore.iec.ch/publication/30082

F.19.2 Top-level

ISO Standard document not available.

F.20 ISO 12006-3

F.20.1 Overview

ISO 12006-3:2007 specifies a language-independent information model which can be used for the development of dictionaries used to store or provide information about construction works. It enables classification systems, information models, object models and process models to be referenced from within a common framework.

From https://www.iso.org/standard/38706.html

See also: https://en.wikipedia.org/wiki/ISO_12006

F.20.2 Top-level

ISO 12006-3:2007(E)

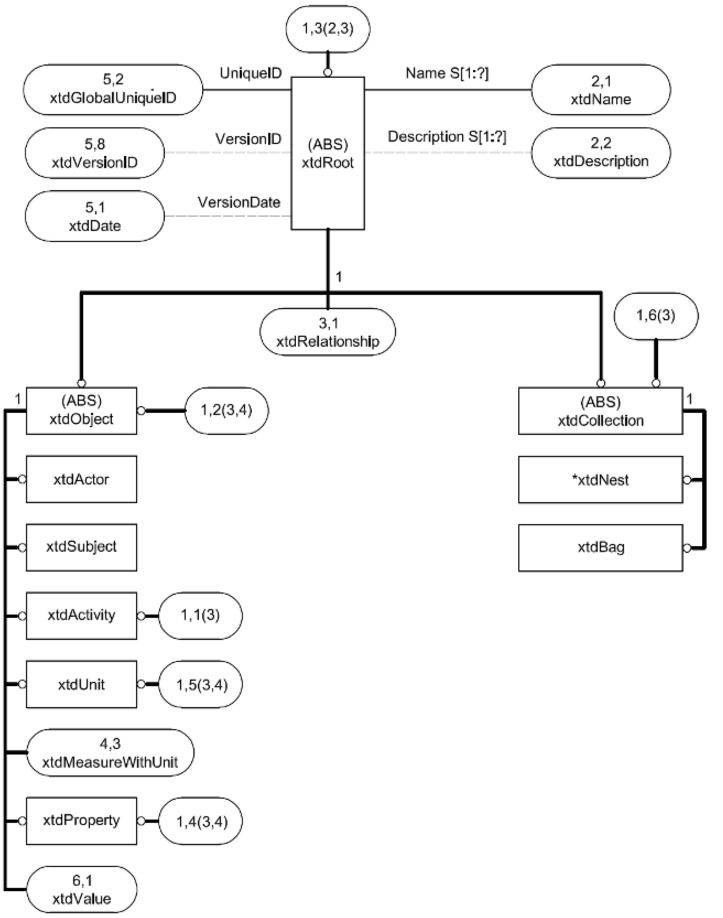


Figure 1 — EXPRESS-G diagram 1 — Top level with root concept

F.20.3 Key characteristics

This is a generic top-level data model

F.21 ISO 15926-2

F.21.1 Overview

ISO 15926-2:2003 specifies a conceptual data model for computer representation of technical information about process plants. [A] generic 4D model that can support all disciplines, supply chain company types and life cycle stages, regarding information about functional requirements, physical solutions, types of objects and individual objects as well as activities.

From https://www.iso.org/standard/29557.html

See also: https://en.wikipedia.org/wiki/ISO_15926

F.21.2 Top-level

ISO 15926-2:2003(E)

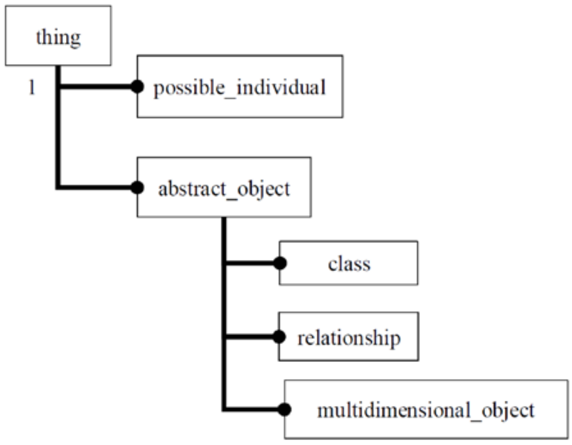


Figure 6 — Part of the model subtype/supertype hierarchy

F.21.3 Key characteristics

This is a well-documented heavyweight foundational ontology. It is an extensional ontology with a general unifying approach – illustrated in the journey in Figure 32. It is an ISO standard, whose development was partly influenced by the BORO.

It was developed in the 1990s using the EXPRESS data modelling language and it includes a meta-model to support the implementation of an RDL to enable domain ontologies to be developed in data as extensions.

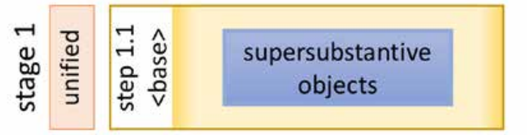


Figure 32 – ISO 15926 Stratification Journey – one stratum

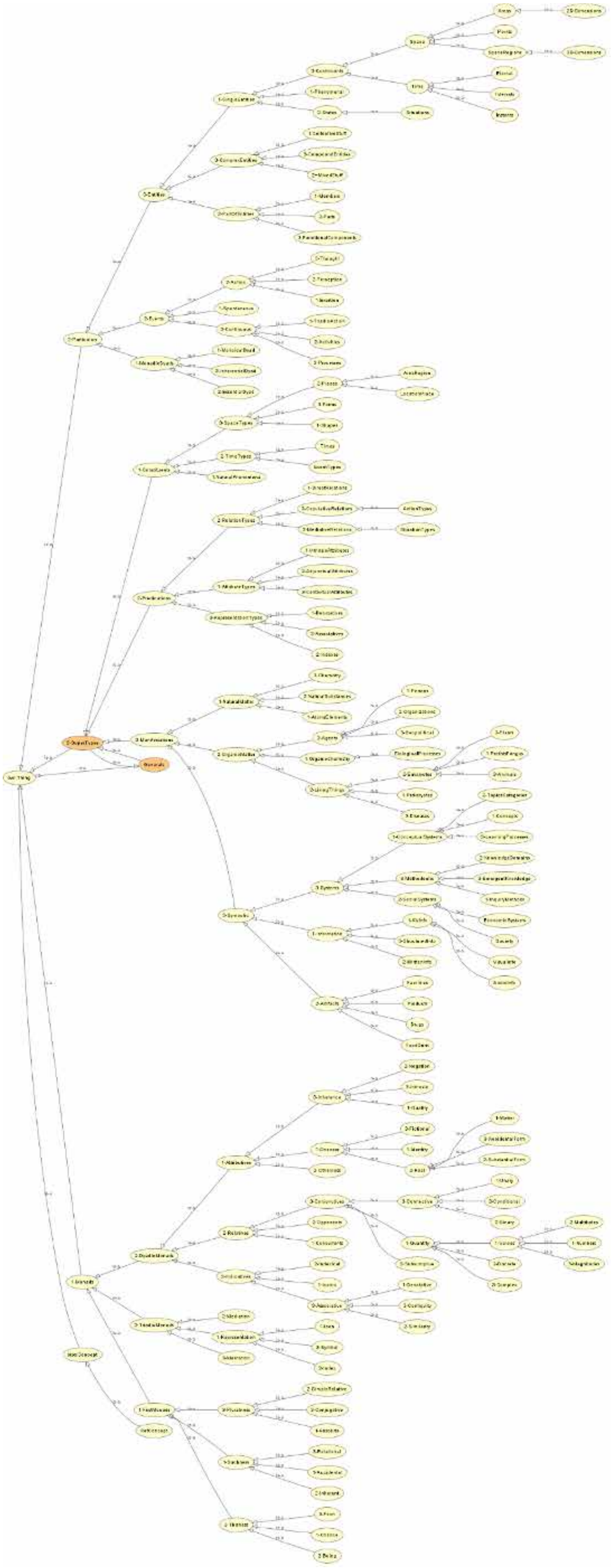
F.22 KKO: KBpedia Knowledge Ontology

F.22.1 Overview

KBpedia is a comprehensive knowledge structure for promoting data interoperability and knowledge-based artificial intelligence, or KBAI. The KBpedia knowledge structure combines seven ‘core’ public knowledge bases – Wikipedia, Wikidata, schema.org, DBpedia, GeoNames, OpenCyc, and standard UNSPSC products and services – into an integrated whole. KBpedia’s upper structure, or knowledge graph, is the KBpedia Knowledge Ontology. We base KKO on the universal categories and knowledge representation insights of the great 19th century American logician, polymath and scientist, Charles Sanders Peirce. The upper structure of the KBpedia Knowledge Ontology (KKO) is informed by the triadic logic and universal categories of Charles Sanders Peirce. This trichotomy, also the basis for his views on semiosis (or the nature of signs), was in Peirce’s view the most primitive or reduced manner by which to understand and categorize things, concepts and ideas.

From https://kbpedia.org/docs/kko-upper-structure/

F.22.2 Top-level



F.22.3 Key characteristics

This is a natural language ontology influenced by Charles Peirce.

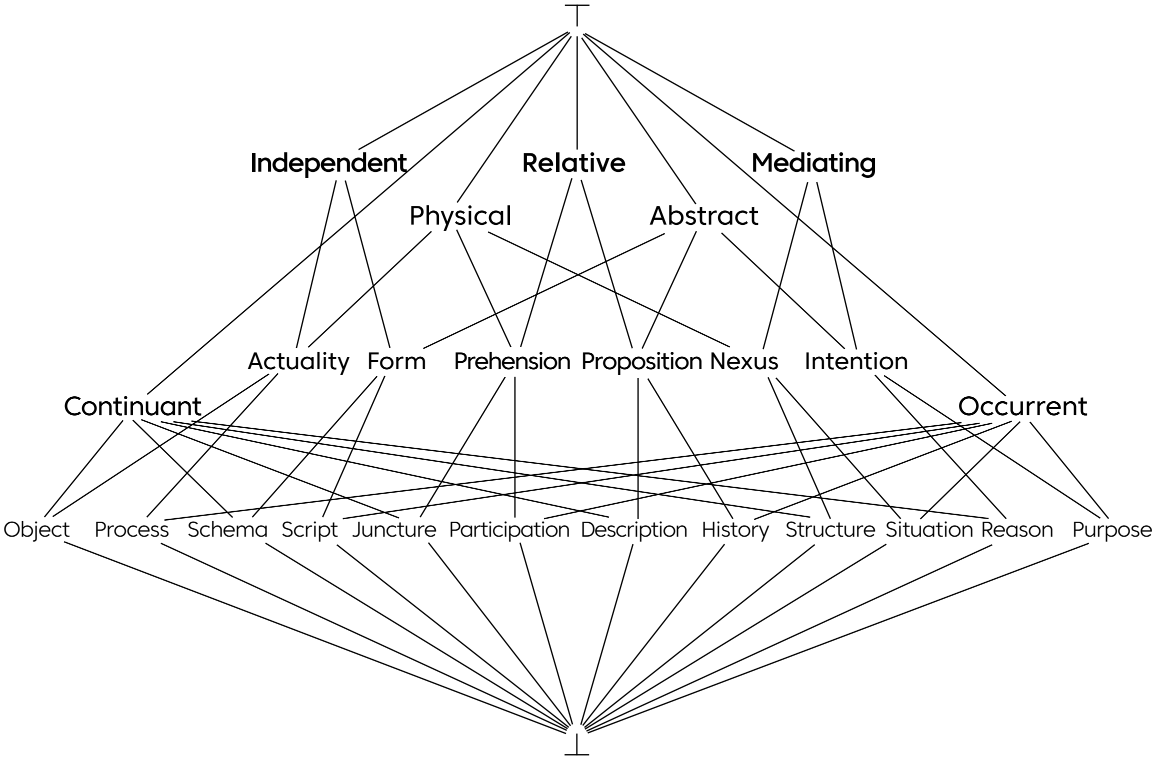
F.23 KR Ontology – Knowledge Representation Ontology

F.23.1 Overview

The KR Ontology is defined in the book Knowledge Representation by John F. Sowa. Its categories have been derived from a synthesis of various sources, but the two major influences are the semiotics of Charles Sanders Peirce and the categories of existence of Alfred North Whitehead. The primitive categories are: Independent, Relative, or Mediating; Physical or Abstract; Continuant or Occurrent.

From http://www.jfsowa.com/ontology/toplevel.htm

F.23.2 Top-level



F.23.3 Key characteristics

KR is a heavyweight foundational ontology influenced by Charles Peirce.

F.24 MarineTLO: A Top-Level Ontology for the Marine Domain

F.24.1 Overview

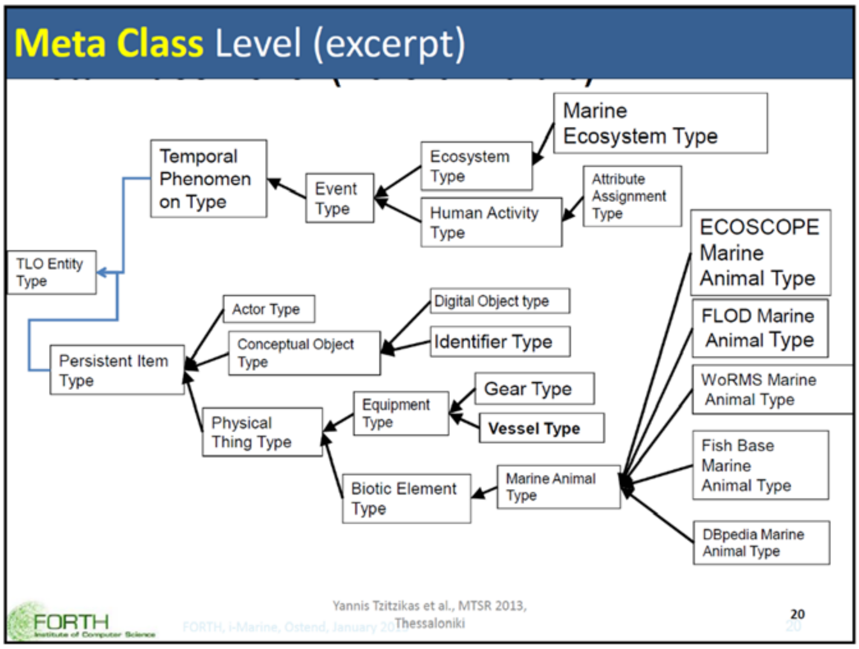
Is a top-level ontology, generic enough to provide consistent abstractions or specifications of concepts included in all data models or ontologies of marine data sources and provide the necessary properties to make this distributed knowledge base a coherent source of facts relating observational data with the respective spatiotemporal context and categorical (systematic) domain knowledge.

From https://en.wikipedia.org/wiki/Upper_ontology#MarineTLO

See also: https://projects.ics.forth.gr/isl/MarineTLO/

F.24.2 Top-level

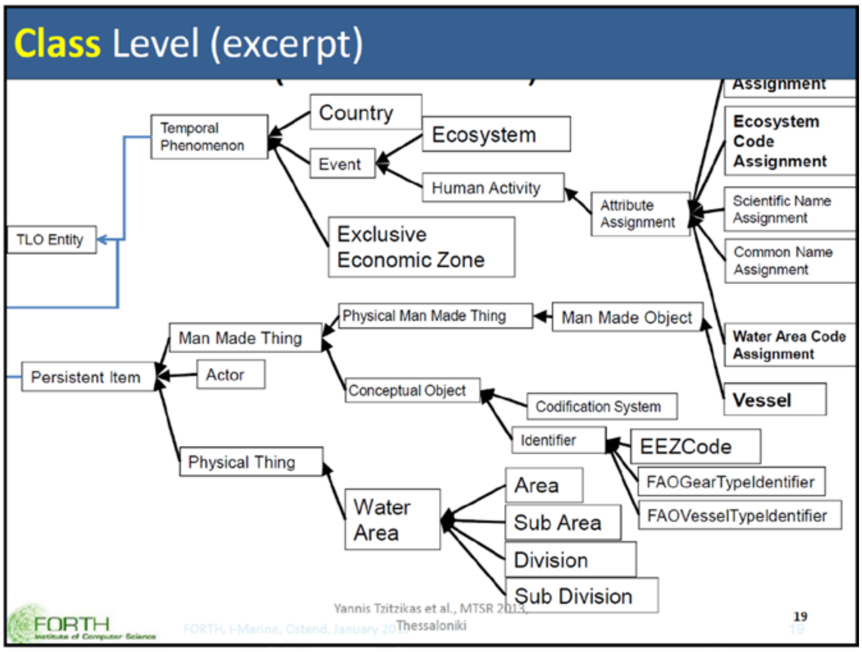
The model is formulated as an object-oriented semantic model, hence it has meta-class and class levels, as shown below.



Meta Class Level (excerpt)

Marine Ecosystem Type
Temporal Phenomenon Type
Ecosystem Type
Event Type
Human Activity Type
Attribute Assignment Type
ECOSCOPE Marine Animal Type
TLO Entity Type
Actor Type
Digital Object type
Conceptual Object Type
Identifier Type
FLOD Marine Animal Type
Persistent Item Type
WoRMS Marine Animal Type
Gear Type
Equipment Type
Physical Thing Type
Vessel Type
Fish Base Marine Animal Type
Biotic Element Type
Marine Animal Type
DBpedia Marine Animal Type

FORTH Institute of Computer Science
Yannis Tzitzikas et al., MTSR 2013, Thessaloniki
20



Class Level (excerpt)

Assignment
Temporal Phenomenon
Country
Ecosystem
Ecosystem Code Assignment
Event
Human Activity
Attribute Assignment
Scientific Name Assignment
TLO Entity
Exclusive Economic Zone
Common Name Assignment
Physical Man Made Thing
Man Made Object
Man Made Thing
Water Area Code Assignment
Persistent Item
Actor
Conceptual Object
Vessel
Codification System
Identifier
EEZCode
Physical Thing
FAOGearTypeIdentifier
Area
Water Area
Sub Area
FAOVesselTypeIdentifier
Division
Sub Division

FORTH Institute of Computer Science
Yannis Tzitzikas et al., MTSR 2013, Thessaloniki
19

The first broad division into persistent items and temporal phenomenon, looks similar to the endurantist’s continuant and occurrent distinction.

F.24.3 Key characteristics

Appears to have a lightweight top-level – with few ontological commitments.

Possibly an endurantist commitment.

F.24.4 Excerpts

“Formulation – It is an object-oriented semantic model, expressed to a form comprehensible to both documentation experts and information scientists while readily can be converted to machine-readable formats such as RDF Schema, OWL, etc”

F.25 MIMOSA CCOM – (Common Conceptual Object Model)

F.25.1 Overview

MIMOSA CCOM (Machinery Information Management Open Systems Alliance – Common Conceptual Object Model) serves as an information model for the exchange of asset information. Its core mission is to facilitate standards-based interoperability between systems: providing an XML model to allow systems to electronically exchange data.

From https://www.mimosa.org/mimosa-ccom/

See also: https://en.wikipedia.org/wiki/OpenO%26M

F.25.2 Top-level

Class		
OrganizationType	DataQualityType	PurchaseConditionType
Organization	DocumentType	ReadinessType
EffectiveStatusType	EngineeringStudyEntryType	SeverityLevelType
AgentRoleType	EngineeringStudyType	RegionType
AgentType	EventType	RequestType
AmbiguitySetType	GPSDatumType	SegmentType
AssetType	GPSElevationType	SignalProcessBlockType
Enumeration	GPSPrecisionType	SignalProcessStreamType
EnumerationItem	HealthLevelType	SiteType
UOMQuantity	HighlightType	Site
UnitOfMeasure	Document	SolutionPackageType
AttributeSetType	InfoSource	SourceDetectorType
AttributeType	InfoSourceType	StandardDataType
AverageSynchType	LifecycleStatusKind	TestComponentType
AverageType	LifecycleStatusType	TestType
AverageWeightType	LogicalConnectorType	TransducerAxisDirectionType
BLOBDataType	LogisticResourceType	TransducerType
BreakdownStructureType	MeasurementLocationType	WindowType
CalculationType	MeasurementSourceType	WorkStatusType
CCOMClass	MeshType	WorkManagementType
ChangePatternType	OrderedListType	WorkTaskType
ConnectionType	PostScalingType	
CriticalityScaleType	PriorityLevelType	

F.25.3 Key characteristics

A generic data model with no explicit top-level ontological commitment.

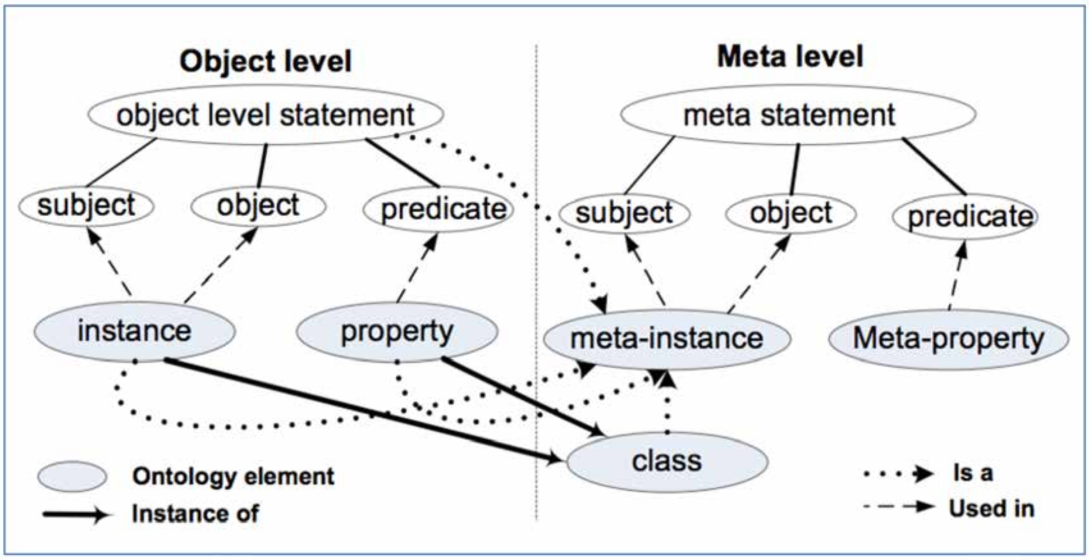
Note the significant number of higher order types in the list of entities (the entity types ending in ‘Type’).

F.26 OWL – Web Ontology Language

F.26.1 Overview

The Web Ontology Language (OWL) is a family of knowledge representation languages for authoring ontologies. Ontologies are a formal way to describe taxonomies and classification networks, essentially defining the structure of knowledge for various domains: the nouns representing classes of objects and the verbs representing relations between the objects.

F.26.2 Top-level



F.26.3 Key characteristics

A generic top-level data model with a lightweight (or no) foundational ontological commitments.

F.27 ProtOn – PROTo ONtology

F.27.1 Overview

PROTON (PROTo ONtology) is a basic subsumption hierarchy which provides coverage of most of the upper-level concepts necessary for semantic annotation, indexing, and retrieval.

From https://en.wikipedia.org/wiki/Upper_ontology#PROTON

See also: https://ontotext.com/documents/proton/Proton-Ver3.0B.pdf

F.27.2 Top-level

Figure 1. A view of the top part of the PROTON class hierarchy

F.27.3 Key characteristics

A natural language ontology.

F.27.4 Relevant extracts

Extracts from: https://ontotext.com/documents/proton/Proton-Ver3.0B.pdf

Extract 1 – Design principles

The PROTON ontology contains about 500 classes and 150 properties, providing coverage of the general concepts necessary for a wide range of tasks, including semantic annotation, indexing, and retrieval. The design principles can be summarized as follows:

- domain-independence;
- lightweight logical definitions;
- alignment with popular metadata standards;
- good coverage of named entity types and concrete domains (i.e. modelling of concepts such as people, organizations, locations, numbers, dates, addresses, etc.); and
- good coverage of instance data in Linked Open Data Reasonable view Fact Forge.

The ontology is encoded in a fragment of OWL Lite and split into four modules: System, Top, Extent, and KM (Knowledge Management). A snapshot of the PROTON class hierarchy is given on Figure 1, showing the Top and the Extent modules.

Extract 2 – PROTON is relatively un-restrictive

1. Design Rationales

PROTON is designed as a lightweight upper-level ontology for use in Knowledge

Management and Semantic Web applications. The above mission statement has two important implications:

- PROTON is relatively un-restrictive. It specifies only a hierarchy of classes and domain and range of properties defined within it, but it does not impose any other restrictions on the meaning of the classes and properties.
- PROTON is not precise in some aspects, for instance regarding the conceptualization of space and time. This is partly because proper models for these aspects would require using a logical apparatus, which is beyond the limits acceptable for many of the tasks to which we wish to apply PROTON (e.g. queries and management of huge datasets/ knowledge bases); and partly because it is very hard to craft strict and precise conceptualizations for these concepts, which are adequate for a wide range of domains and applications.

F.28 Schema.org

F.28.1 Overview

Schema.org is a collaborative, community activity with a mission to create, maintain, and promote schemas for structured data on the Internet, on web pages, in email messages, and beyond.

From https://en.wikipedia.org/wiki/Schema.org
See also: https://schema.org/

F.28.2 Top-level

Thing

- Action
- CreativeWork
- Event
- Intangible
- MedicalEntity
- Organization
- Person
- Place
- Product

F.28.3 Key characteristics

A generic top-level data model with lightweight (or no) foundational ontological commitments.

F.28.4 Relevant extracts

Extracts from: https://schema.org/docs/datamodel.html

Extract 1 – Data Model Design

The data model used is very generic and derived from RDF Schema (which in turn was derived from CycL, see History section for details ...).

1. We have a set of types, arranged in a multiple inheritance hierarchy where each type may be a sub-class of multiple types.
2. We have a set of properties:
    1. each property may have one or more types as its domains. The property may be used for instances of any of these types.
    2. each property may have one or more types as its ranges. The value(s) of the property should be instances of at least one of these types.

The decision to allow multiple domains and ranges was purely pragmatic. While the computational properties of systems with a single domain and range are easier to understand, in practice, this forces the creation of a lot of artifical types, which are there purely to act as the domain/range of some properties.

Like many other systems, the schema presented here can be extended (with a few types like Class and Property and a few properties like domainIncludes and rangeIncludes) to allow for reflection, i.e., for the schema to be represented in terms of itself.

Extract 2 – Not intended to be a 'global ontology'

The type hierarchy presented on this site is not intended to be a 'global ontology' of the world. When founded in 2011 it was strictly focussed around the types of entities for which the project's founders (Microsoft, Yahoo!, Google and Yandex), could reasonably expect to provide some special treatment for via search engines. As the project has evolved, introducing more community collaboration and extension mechanisms, its scope has expanded gradually. However it is still the case that schema.org is not intended as a universal ontology. We expect it to be used alongside other vocabulary that shares our basic datamodel and our use of underlying standards like JSON-LD, Microdata and RDFa

F.29 SENSUS

F.29.1 Overview

We have constructed SENSUS, a 70,000-node terminology taxonomy, as a framework into which additional knowledge can be placed. SENSUS is an extension and reorganization of WordNet.

From https://www.isi.edu/natural-language/projects/ONTOLOGIES.html

F.29.2 Top-level

See Wordnet.

F.29.3 Key characteristics

A natural language ontology based upon Wordnet.

F.30 SKOS

F.30.1 Overview

SKOS is an area of work developing specifications and standards to support the use of knowledge organization systems (KOS) such as thesauri, classification schemes, subject heading systems and taxonomies within the framework of the Semantic Web.

From: https://www.w3.org/2004/02/skos/
See also: https://en.wikipedia.org/wiki/Simple_Knowledge_Organization_System

F.30.2 Top-level

Element categories

The principal element categories of SKOS are concepts, labels, notations, semantic relations, mapping properties, and collections. The associated concepts are listed in the table below.

SKOS Vocabulary Organized by Theme					
Concepts	**Labels & Notation**	**Documentation**	**Semantic Relations**	**Mapping Properties**	**Collections**
Concept	prefLabel	note	broader	broadMatch	Collection
ConceptScheme	altLabel	changeNote	narrower	narrowMatch	orderedCollection
inScheme	hiddenLabel	definition	related	relatedMatch	member
hasTopConcept	notation	editorialNote	broaderTransitive	closeMatch	memberList
topConceptOf		example	narrowerTransitive	exactMatch	
		historyNote	semanticRelation	mappingRelation	
		scopeNote			

From https://en.wikipedia.org/wiki/Simple_Knowledge_Organization_System#Element_categories

F.30.3 Key characteristics

A natural language ontology that has a structure similar to a thesaurus.

F.30.4 Relevant extracts

Extracts from: https://www.w3.org/TR/skos-reference/

Extract 1 – Similar structure to thesauri, etc.

Many knowledge organization systems, such as thesauri, taxonomies, classification schemes and subject heading systems, share a similar structure, and are used in similar applications. SKOS captures much of this similarity and makes it explicit, to enable data and technology sharing across diverse applications.

Extract 2 – Higher order concepts

9. Concept Collections

9.1. Preamble

SKOS concept collections are labeled and/or ordered groups of SKOS concepts.

Collections are useful where a group of concepts shares something in common, and it is convenient to group them under a common label, or where some concepts can be placed in a meaningful order.

F.31 SUMO

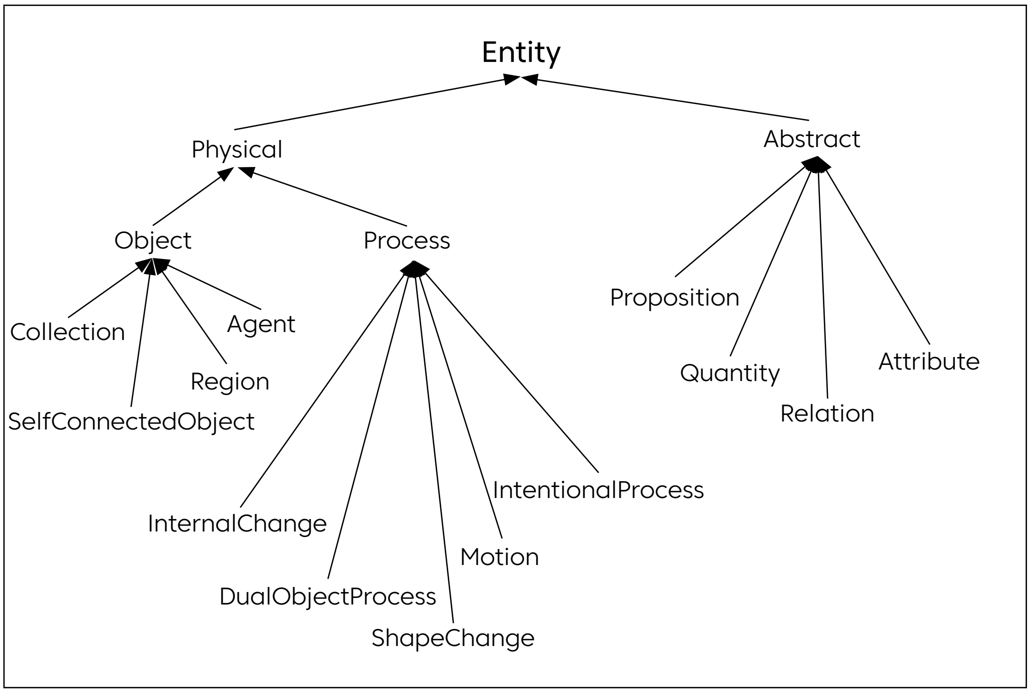
F.31.1 Overview

Is an upper ontology intended as a foundation ontology for a variety of computer information processing systems. SUMO defines a hierarchy of classes and related rules and relationships. These are expressed in a version of the language SUO-KIF which has a LISP-like syntax. A mapping from WordNet synsets to SUMO has been defined. Initially, SUMO was focused on meta-level concepts (general entities that do not belong to a specific problem domain), and thereby would lead naturally to a categorization scheme for encyclopedias. It has now been considerably expanded to include a mid-level ontology and dozens of domain ontologies. SUMO is organized for interoperability of automated reasoning engines.

From https://en.wikipedia.org/wiki/Upper_ontology#SUMO_(Suggested_Upper_Merged_Ontology)

See also: http://www.adampease.org/OP/, https://en.wikipedia.org/wiki/Suggested_Upper_Merged_Ontology

F.31.2 Top-level



F.31.3 Key characteristics

A natural language ontology that supports higher order types (see ‘class’ and ‘set’ below).

F.31.4 Relevant Extracts

Extract 1 – Collection, Class and Set

(documentation Collection EnglishLanguage “Collections have members like Classes, but, unlike Classes, they have a position in space-time and members can be added and subtracted without thereby changing the identity of the Collection. Some examples are toolkits, football teams, and flocks of sheep.”)

(documentation Class EnglishLanguage “Classes differ from Sets in three important respects. First, Classes are not assumed to be extensional. That is, distinct Classes might well have exactly the same instances. Second, Classes typically have an associated `condition’ that determines the instances of the Class. So, for example, the condition `human’ determines the Class of Humans. Note that some Classes might satisfy their own condition (e.g., the Class of Abstract things is Abstract) and hence be instances of themselves. Third, the instances of a class may occur only once within the class, i.e. a class cannot contain duplicate instances.”)

(documentation Set EnglishLanguage “A SetOrClass that satisfies extensionality as well as other constraints specified by some choice of set theory. Sets differ from Classes in two important respects. First, Sets are extensional – two Sets with the same elements are identical. Second, a Set can be an arbitrary stock of objects. That is, there is no requirement that Sets have an associated condition that determines their membership. Note that Sets are not assumed to be unique sets, i.e. elements of a Set may occur more than once in the Set.”)

NB: Collections of collections are fusion of their members and so do not ascend a type.

F.32 TMRM/TMDM – Topic Map Reference/Data Models

Overview

A **topic map** is a standard for the representation and interchange of knowledge, with an emphasis on the findability of information. Topic maps were originally developed in the late 1990s as a way to represent back-of-the-book index structures so that multiple indexes from different sources could be merged. However, the developers quickly realized that with a little additional generalization, they could create a meta-model with potentially far wider application. The ISO standard is formally known as **ISO/IEC 13250:2003**.

A topic map represents information using

- *topics*, representing any concept, from people, countries, and organizations to software modules, individual files, and events,
- *associations*, representing hypergraph relationships between *topics*, and
- *occurrences*, representing information resources relevant to a particular *topic*.

Topic maps are similar to concept maps and mind maps in many respects, though only topic maps are ISO standards. Topic maps are a form of semantic web technology similar to RDF.

From https://en.wikipedia.org/wiki/Topic_map

See also: https://www.isotopicmaps.org/tmrm/, https://www.iso.org/standard/40757.html (ISO/IEC 13250-5:2015 Information technology – Topic Maps – Part 5: Reference model), https://www.isotopicmaps.org/sam/sam-model/ (Topic Maps – Data Model)

F.32.1 Top-level

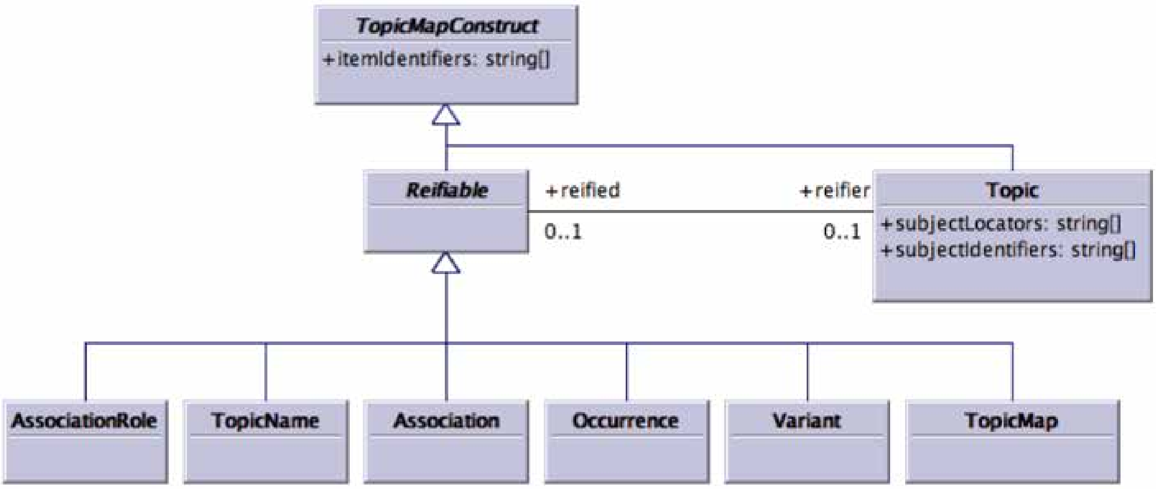


Figure 1 — The class hierarchy

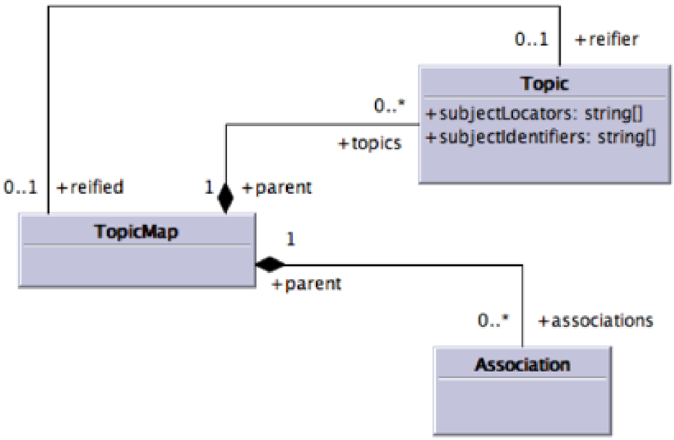


Figure 2 — The topic map item

The topic map item represents the topic map. Topic map items have the following properties:

1. **[topics]**: A set of topic items. All the topics in the topic map.
2. **[associations]**: A set of association items. All the associations in the topic map.
3. **[reifier]**: A topic item, or null. If not null, the topic that reifies the topic map.
4. **[item identifiers]**: A set of locators. The item identifiers of the topic map.

F.32.2 Key characteristics

A generic ontology with little ontological commitment. Allows higher order levels. Potentially non-well bounded (no notion of particulars). Instantiation is possibly gunky and junky across topics. Sub-class is possibly gunky and junky across topics. No explicit commitment to mereological relations.

F.32.3 Relevant Extracts

Extract 1 (https://en.wikipedia.org/wiki/Topic_map#Ontology_and_merging)

Ontology and merging

Topics, associations, and occurrences can all be typed, where the types must be defined by the one or more creators of the topic map(s). The definitions of allowed types is known as the ontology of the topic map.

Extract 2 (https://www.isotopicmaps.org/tmrm/ – http://www.isotopicmaps.org/TMRM/TMRM-7.0/tmrm7.pdf)

The Topic Maps Reference Model - 5 Ontological Commitments

This Standard deliberately leaves undefined the methods whereby subject proxies are derived or created. No specific mechanism of subject identification is inherent in or mandated by this Standard, nor does it predefine any subject proxies.

NOTE 1 Any subject proxy design choices would be specific to a particular application domain and would exclude equally valid alternatives that might be appropriate or necessary in the contexts of various requirements.

Two types of relationships, sub (subclass of) and isa (instance of), are defined. These predicates are always interpreted relative to a given map m:

a) Two proxies c, c_0 can be in a subclass-superclass relationship, sub$_m$ _ m × m. In such a case, the same relationship can be stated either c is a subclass of c_0 or c_0 is a superclass of c.

   sub$_m$ is supposed to be reflexive and transitive. Reflexive implies that any proxy is a subclass of itself, regardless whether the proxy is used as a class in the map or not: x sub$_m$ x for all x 2 m.

   Transitive implies that if a proxy c is a subclass of another, c_0, and that subclasses c_{00}, then c is also a subclass of c_{00}, i.e. if c sub$_m$ c_0 and c_0 sub$_m$ c_{00} then also c sub$_m$ c_{00} must be true.

   NOTE 2 Circular subclass relationships may exist in a map.

b) Two proxies a, c can be in an isa relationship, isa$_m$ _ m × m. In such a case, the same relationship can be stated either a is an instance of c or c is the type of a.

   The isa relationship is supposed to be non-reflexive, i.e. x isa$_m$ x for no x 2 m, so that no proxy can be an instance of itself. Additionally, whenever a proxy a is an instance of another c, then a is an instance of any superclass of c: if x isa$_m$ c and c sub$_m$ c_0, then x isa$_m$ c_0 is true.

NOTE 3 This Standard does not mandate any particular way of representing such relationships inside a map. One option is to model such a relationship simply with a property using a certain key (say type). An alternative way is to provide a proxy for each such relationship. Such relationship proxies could, for example, have properties whose keys are instance and class, or respectively subclass and superclass.

Extract 3 (https://www.isotopicmaps.org/sam/sam-model/#sect-pubsubj)

7 Core subject identifiers

…

7.2 The type-instance relationship

A **topic type** is a subject that captures some commonality in a set of subjects. Any subject that belongs to the extension of a particular topic type is known as an instance of that topic type. A topic type may itself be an instance of another topic type, and there is no limit to the number of topic types a subject may be an instance of.

The type-instance relationship is not transitive. That is, if B is an instance of the type A, and C is an instance of the type B, it does not follow that C is an instance of A.

…

7.3 The supertype-subtype relationship

The supertype-subtype relationship is the relationship between a more general type (the supertype) and a specialization of that type (the subtype). If B is the subtype of A, it follows that every instance of B is also an instance of A. The converse is not necessarily true. A type may have any number of subtypes and supertypes.

The supertype-subtype relationship is transitive, which means that if B is a subtype of A, and C a subtype of B, C is also a subtype of A.

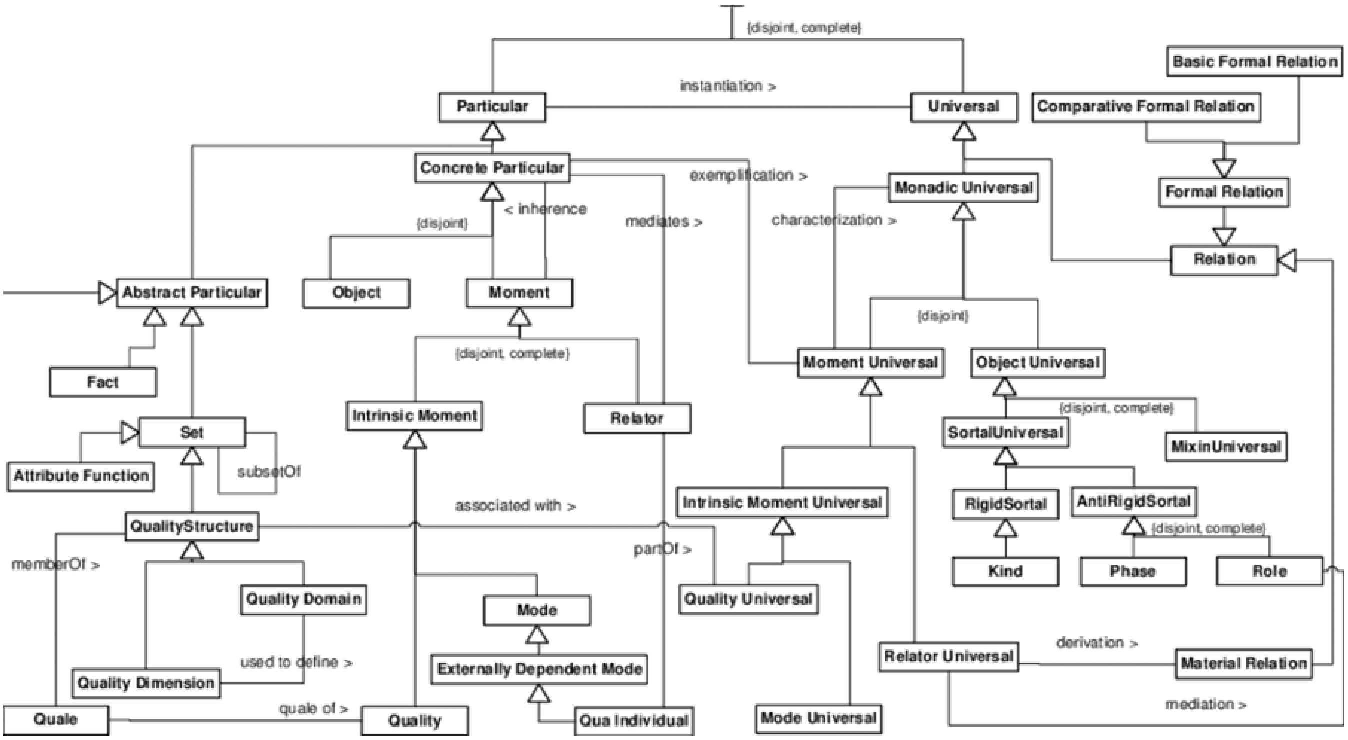
> **NOTE:**
>
> Loops in this relationship are allowed, and should be interpreted to mean that the sets of instances for all types in the loop are the same. This does not, however, necessarily imply that the types are the same.
>
> **NOTE:**
>
> The semantics of the supertype-subtype relationship implies the existence of further type-instance and supertype-subtype relationships in addition to those explicitly represented by associations in the topic map. This part of ISO/IEC13250 does not require associations to be created for inferred relationships.

F.33 UFO

F.33.1 Overview

Incorporates developments from GFO, DOLCE and the Ontology of Universals underlying OntoClean in a single coherent foundational ontology.

F.33.2 Top-level



F.33.3 Key characteristics

UFO is a well-documented heavyweight foundational ontology that have evolved over time.

F.34 UMBEL

F.34.1 Overview

Is a logically organized knowledge graph of 34,000 concepts and entity types that can be used in information science for relating information from disparate sources to one another. Since UMBEL is an open-source extract of the OpenCyc knowledge base, it can also take advantage of the reasoning capabilities within Cyc.

F.34.2 Top-level

See Cyc.

F.34.3 Key characteristics

A natural language ontology with a generic top-level.

F.35 UML

F.35.1 Overview

The Unified Modeling Language (UML) is a general-purpose, developmental, modeling language in the field of software engineering that is intended to provide a standard way to visualize the design of a system.

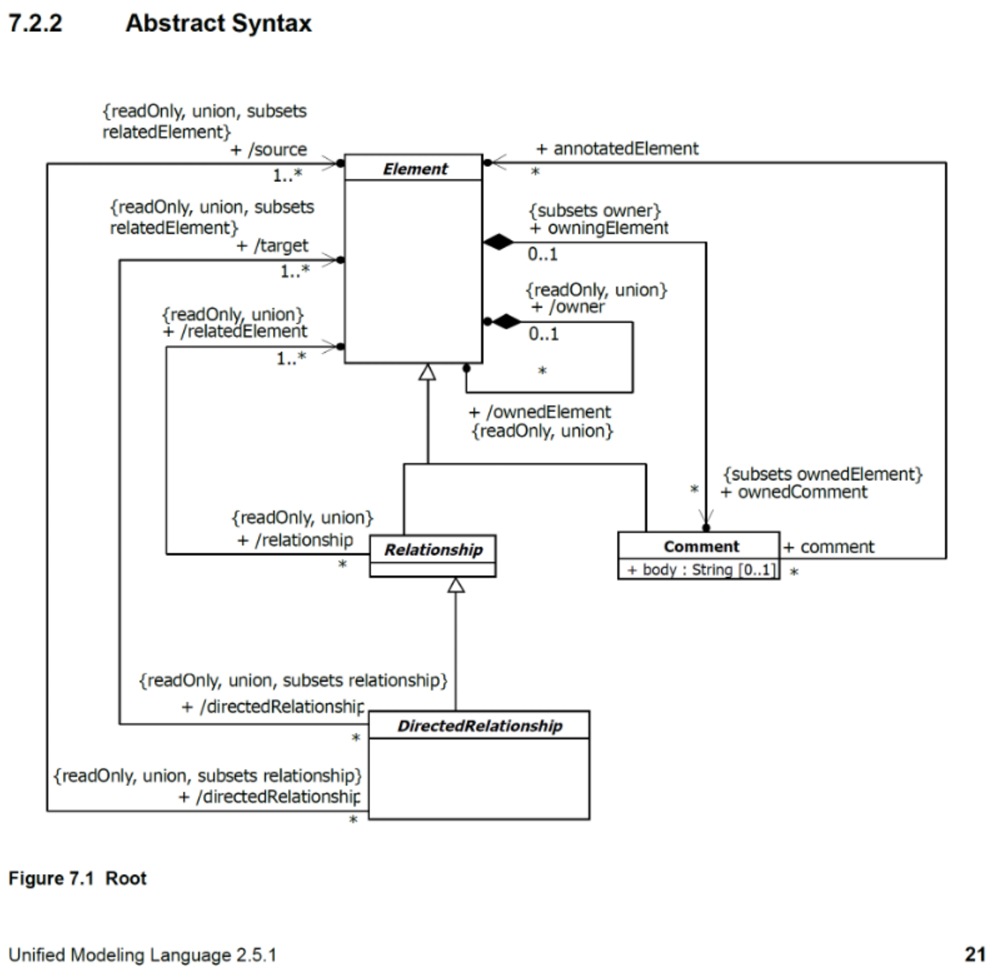
The creation of UML was originally motivated by the desire to standardize the disparate notational systems and approaches to software design. It was developed by Grady Booch, Ivar Jacobson and James Rumbaugh at Rational Software in 1994 – 1995, with further development led by them through 1996.

In 1997, UML was adopted as a standard by the Object Management Group (OMG), and has been managed by this organization ever since. In 2005, UML was also published by the International Organization for Standardization (ISO) as an approved ISO standard. Since then the standard has been periodically revised to cover the latest revision of UML.

From https://en.wikipedia.org/wiki/Unified_Modeling_Language

See also: http://uml.org/, https://www.iso.org/standard/32620.html (ISO Standard NB v1.4.2)

F.35.2 Top-level



7.2.2 Abstract Syntax

Figure 7.1 Root

Unified Modeling Language 2.5.1 **21**

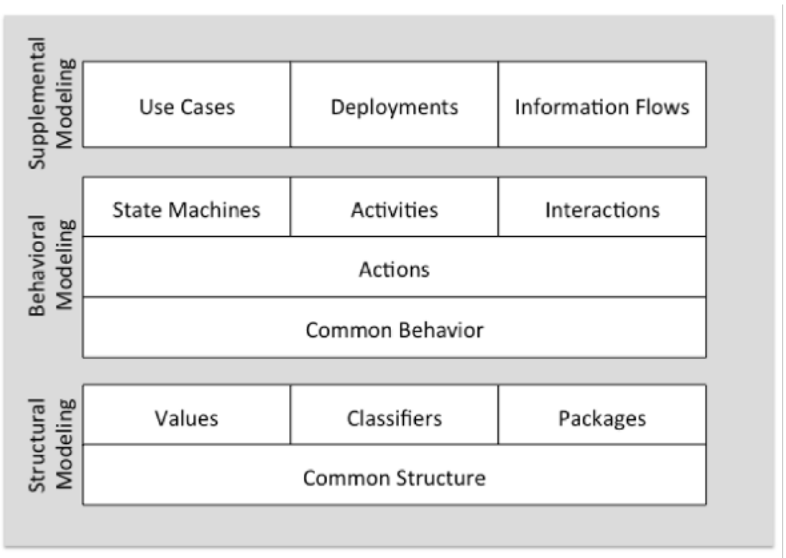


Figure 6.1 Semantic Areas of UML

From OMG® Unified Modeling Language® (OMG UML®) – Version 2.5.1 (Normative URL: https://www.omg.org/spec/UML/)

F.35.3 Key characteristics

A generic data model with a focus on describing computer systems. Supports multiple inheritance and classification, though as these are typically not feasible in programming languages, these are not usually used. Typically, first order, though limited functionality through generalisation sets (powertypes) to move up to higher orders (see extract below).

F.35.4 Relevant Extracts

From OMG® Unified Modeling Language® (OMG UML®) – Version 2.5.1 (Normative URL: https://www.omg.org/spec/UML/)

Extract 1

6.3 On the Semantics of UML

6.3.1 Models and What They Model

A model is always a model of something. The thing being modeled can generically be considered a system within some domain of discourse. The model then makes some statements of interest about that system, abstracting from all the details of the system that could possibly be described, from a certain point of view and for a certain purpose. For an existing system, the model may represent an analysis of the properties and behavior of the system. For a planned system, the model may represent a specification of how the system is to be constructed and behave.

A UML model consists of three major categories of model elements, each of which may be used to make statements about different kinds of *individual things* within the system being modeled (termed simply "individuals" in the following). These categories are:

- *Classifiers*. A classifier describes a set of objects. An object is an individual with a state and relationships to other objects. The state of an object identifies the values for that object of properties of the classifier of the object. (In some cases, a classifier itself may also be considered an individual; for example, see the discussion of static structural features in sub clause 9.4.3.)
- *Events*. An event describes a set of possible occurrences. An *occurrence* is something that happens that has some consequence with regard to the system.
- *Behaviors*. A behavior describes a set of possible executions. An *execution* is a performance of a set of actions (potentially over some period of time) that may generate and respond to occurrences of events, including accessing and changing the state of objects. (As described in sub clause 13.2, behaviors are themselves modeled in UML as kinds of classifiers, so that executions are essentially modeled as objects. However, for the purposes of the present discussion, it is clearer to consider behaviors and executions to be in a separate semantic category than classifiers and objects.)

Extract 2

9.7 Generalization Sets

9.7.1 Summary

GeneralizationSet provides a way to group Generalizations into orthogonal dimensions. A GeneralizationSet may be associated with a Classifier called its powertype. These techniques provide additional expressive power for organizing classification hierarchies.

…

9.7.3 Semantics

Generalizations may be grouped to represent orthogonal dimensions of generalization. Each group is represented by a GeneralizationSet. The generalizationSet property designates the GeneralizationSets to which the Generalization belongs. All of the Generalizations in a particular GeneralizationSet shall have the same general Classifier.

The isCovering property of GeneralizationSet specifies whether the specific Classifiers of the Generalizations in that set are complete, in the following sense: if isCovering is true, then every instance of the general Classifier is an instance of (at least) one of the specific Classifiers. The isDisjoint property specifies whether the specific Classifiers of the Generalizations in that set may overlap, in the following sense: if isDisjoint is true, then no instance of any of the specific

Classifiers may also be an instance of any other of the specific Classifiers. By default, both properties are false.

A GeneralizationSet may optionally be associated with a Classifier called its powertype. This means that for every Generalization in the GeneralizationSet, the specializing Classifier is uniquely associated with an instance of the powertype, i.e., there is a 1-1 correspondence between instances of the powertype and specializations in the GeneralizationSet, so that the powertype instances and the corresponding Classifiers may be treated as semantically equivalent. How this semantic equivalence is implemented and how its integrity is maintained is not defined within the scope of UML.

F.36 UMLS – Unified Medical Language System

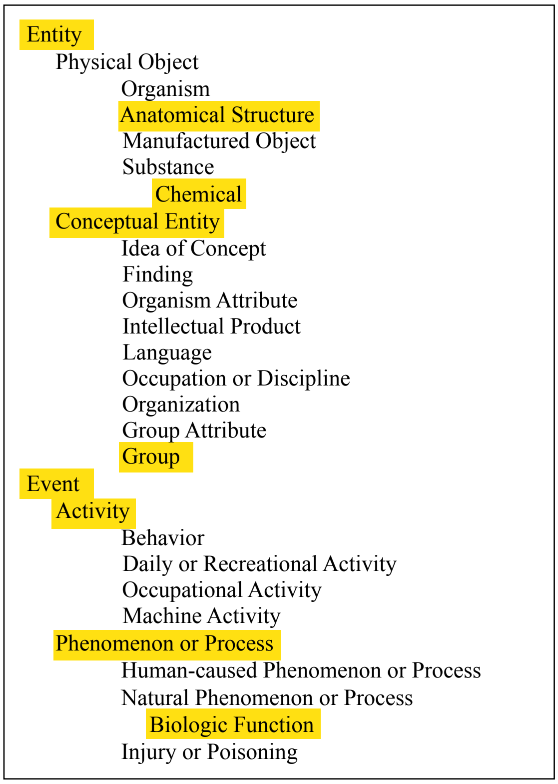
F.36.1 Overview

The Unified Medical Language System (UMLS) is a compendium of many controlled vocabularies in the biomedical sciences (created 1986). It provides a mapping structure among these vocabularies and thus allows one to translate among the various terminology systems; it may also be viewed as a comprehensive thesaurus and ontology of biomedical concepts. UMLS further provides facilities for natural language processing. It is intended to be used mainly by developers of systems in medical informatics.

From https://en.wikipedia.org/wiki/Unified_Medical_Language_System

See also: https://www.nlm.nih.gov/research/umls/index.html

F.36.2 Top-level



- Entity
  - Physical Object
    - Organism
    - Anatomical Structure
    - Manufactured Object
    - Substance
      - Chemical
  - Conceptual Entity
    - Idea of Concept
    - Finding
    - Organism Attribute
    - Intellectual Product
    - Language
    - Occupation or Discipline
    - Organization
    - Group Attribute
    - Group
- Event
  - Activity
    - Behavior
    - Daily or Recreational Activity
    - Occupational Activity
    - Machine Activity
  - Phenomenon or Process
    - Human-caused Phenomenon or Process
    - Natural Phenomenon or Process
      - Biologic Function
    - Injury or Poisoning

F.36.3 Key characteristics

A natural language ontology.

F.36.4 Relevant extracts

https://semanticnetwork.nlm.nih.gov/

Semantic Network

The Semantic Network consists of (1) a set of broad subject categories, or Semantic Types, that provide a consistent categorization of all concepts represented in the UMLS Metathesaurus, and (2) a set of useful and important relationships, or Semantic Relations, that exist between Semantic Types. This section of the documentation provides an overview of the Semantic Network, and describes the files of the Semantic Network. Sample records illustrate structure and content of these files.

F.37 WordNet

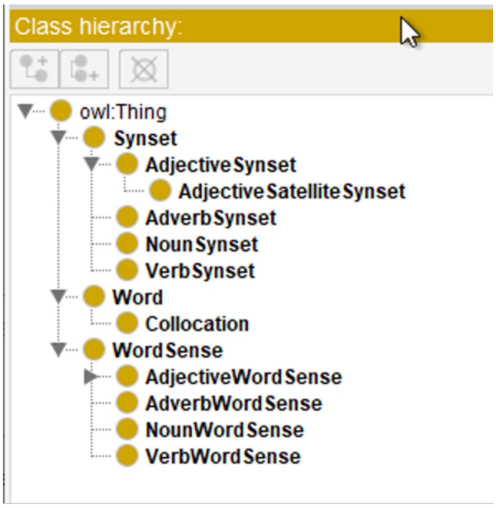
F.37.1 Overview

WordNet is a lexical database of semantic relations between words in more than 200 languages. WordNet links words into semantic relations including synonyms, hyponyms, and meronyms. The synonyms are grouped into synsets with short definitions and usage examples. WordNet can thus be seen as a combination and extension of a dictionary and thesaurus.

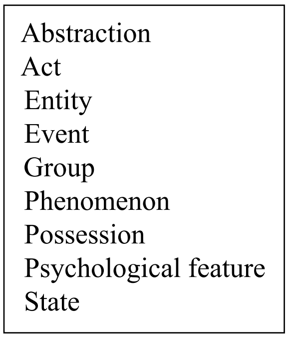
From https://en.wikipedia.org/wiki/WordNet

See also: https://wordnet.princeton.edu/, https://en.wikipedia.org/wiki/Upper_ontology#WordNet

F.37.2 Top-level



- owl:Thing
  - Synset
    - AdjectiveSynset
      - AdjectiveSatelliteSynset
    - AdverbSynset
    - NounSynset
    - VerbSynset
  - Word
    - Collocation
  - WordSense
    - AdjectiveWordSense
    - AdverbWordSense
    - NounWordSense
    - VerbWordSense



Abstraction
Act
Entity
Event
Group
Phenomenon
Possession
Psychological feature
State

F.37.3 Key characteristics

A natural language ontology.

F.37.4 Relevant extracts

From: What is WordNet? (https://wordnet.princeton.edu/)

Extract 1 – Type-instance distinction

WordNet distinguishes among Types (common nouns) and Instances (specific persons, countries and geographic entities). Thus, armchair is a type of chair, Barack Obama is an instance of a president. Instances are always leaf (terminal) nodes in their hierarchies.

F.38 YAMATO – Yet Another More Advanced Top-level Ontology

F.38.1 Overview

YAMATO is developed by Riichiro Mizoguchi, formerly at the Institute of Scientific and Industrial Research of the University of Osaka, and now at the Japan Advanced Institute of Science and Technology. Major features of YAMATO are:

1. an advanced description of quality, attribute, property, and quantity,
2. an ontology of representation,
3. an advanced description of processes and events,
4. the use of a theory of roles.

YAMATO has been extensively used for developing other, more applied, ontologies such as a medical ontology, an ontology of gene, an ontology of learning/instructional theories, an ontology of sustainability science, and an ontology of the cultural domain.

From https://en.wikipedia.org/wiki/Upper_ontology#YAMATO_(Yet_Another_More_Advanced_Top_Ontology)

F.38.2 Top-level

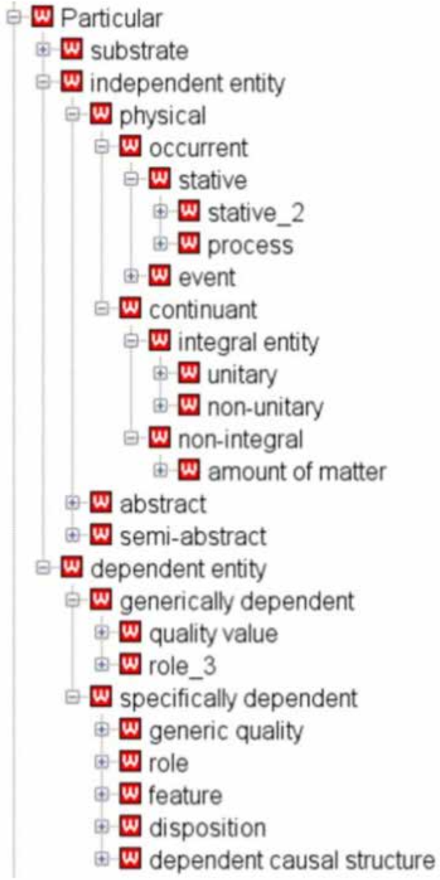


Figure 1.: Top-level categories in YAMATO

F.38.3 Key characteristics

This is a first order ontology - vide the top object called ‘Particular’. For the endurant horizontal aspect it separates occurrents and continuants - vide continuant-occurrent division of physical.

F.38.4 Relevant extracts

From: Mizoguchi, R. (2010). YAMATO: Yet another more advanced top-level ontology. In Proceedings of the sixth Australasian ontology workshop (pp. 1-16).

Extract 1 – Strict single inheritance

It adopts strict single inheritance in is-a hierarchy which is organized according to the rigid definition of is-a and instance-of relations based on the set membership with the notion of essential property.

Extract 2 – Genuine multiple inheritance

For the cases where genuine multiple inheritance is necessary, Hozo prepares IS-A relation which is nothing to do with identity problem of instances but only with property inheritance like subclassof relation in OWL. It may be used only when is-a relation already exists between the two types of interest.

Appendix G
Prior ontological commitment literature

Year	Reference
2002	Partridge, C. (2002). “Note: A couple of meta-ontological choices for ontological architectures.” LADSEB CNR, Padova, Italy.
2002	Borgo, S., Gangemi, A., Guarino, N., Masolo, C. & Oltramari, A. (2002). “Wonderweb deliverable d15.” Laboratory For Applied Ontology-ISTC-CNR.
2003	Masolo, C., Borgo, S., Gangemi, A., Guarino, N. & Oltramari, A. (2003). “WonderWeb deliverable D18 ontology library (final).” ICT Project, 33052, 31.
2003	Grenon, P. (2003). “BFO in a nutshell: A bi-categorial axiomatization of BFO and comparison with DOLCE. IFOMIS Report 06/2003.” Institute for Formal Ontology and Medical Information Science (IFOMIS), University of Leipzig, Leipzig, Germany.
2004	Semy, S. K., Pulvermacher, M. K. & Obrst, L. J. (2004). Toward the use of an upper ontology for US government and US military domains: An evaluation.
2007	Mascardi, V., Cord\`\i, V. & Rosso, P. (2007). “A Comparison of Upper Ontologies.” In WOA (pp. 55–64).
2010	Obrst, L. (2010). “Ontological architectures.” In Theory and applications of ontology: computer applications (pp. 27–66). Springer.
2012	Khan, Z. (2012a). “ROMULUS: Foundational Ontology Comparison: Ontological Commitments.” Retrieved from http://www.thezfiles.co.za/ROMULUS/ontologicalCommitments.html
2012	Khan, Z. & Keet, C. M. (2012b). “ONSET: Automated foundational ontology selection and explanation.” In International Conference on Knowledge Engineering and Knowledge Management (pp. 237–251).
2013	Partridge, C., Mitchell, A. & de Cesare, S. (2013). “Guidelines for developing ontological architectures in modelling and simulation.” In Ontology, Epistemology, and Teleology for Modeling and Simulation (pp. 27–57). Springer.
2016	Partridge, C. & de Cesare, S. (2016). “Grounding for Ontological Architecture Quality: Metaphysical Choices.” In Sebastian Link and Juan Carlos Trujillo (Ed.), Advances in Conceptual Modeling: ER 2016 Workshops, AHA, MoBiD, MORE-BI, MReBA, QMMQ, SCME, and WM2SP, Gifu, Japan, November 14–17, 2016, Proceedings (Vol. 9975, pp. 9–15–XXIII, 251). Gifu, Japan: Springer International Publishing. https://doi.org/10.1007/978-3-319-47717-6

Appendix H
Criteria for a good scientific theory

Thomas Kuhn studied the characteristics of successful improvements in scientific theories, uncovering this list of six features (reported in *Objectivity, value judgment, and theory choice* – reference below):

- Generality: where the scope of the improved theory increased.
- Simplicity: where the improved theory is less complicated (it is typically more 'deeply simple' in the complexity theory sense).
- Explanatory power: the ability of the improved theory to give in-creased meaning.
- Fruitfulness: the ability of the improved theory to meet currently un-specified requirements or to be easily extendable to do so.
- Objectivity: the ability of the improved theory to provide a more objective (shared) understanding of the world.
- Precision: the ability of the improved theory to give a more precise picture of the world.

Making the ontological choices explicit provides an opportunity to take a position that improves on a number of features; explanatory power and objectivity are obvious candidates.

References

Kuhn, T. (1977) "Objectivity, value judgment, and theory choice". In: The Essential Tension: Selected Studies in Scientific Tradition and Change. University of Chicago Press, pp 320-339

Noted as useful for assessing ontologies in, for example:

Partridge, C. (2004). "Setting the Scene: 42 Objects Business Ontology Based Software Development." In Philosophy, Ontology, and Information Systems, colocated with ECOOP 2004. Oslo, Norway.
Retrieved from https://www.academia.edu/30180313/

Partridge, C., Mitchell, A. & de Cesare, S. (2013). "Guidelines for developing ontological architectures in modelling and simulation." In Ontology, Epistemology, and Teleology for Modeling and Simulation (pp. 27–57). Springer.
Retrieved from https://www.academia.edu/32864750/

Appendix I
Detailed notes on 3.2.1 Basis

I.1 Simplicity

The virtues of simplicity have been recognised since Aristotle (“We may assume the superiority *ceteris paribus* of the demonstration which derives from fewer postulates or hypotheses” – *Posterior Analytics*) to Einstein (It can scarcely be denied that the supreme goal of all theory is to make the irreducible basic elements as simple and as few as possible without having to surrender the adequate representation of a single datum of experience).

David Lewis describes the distinction between qualitative and quantitative parsimony in *Counterfactuals* and voiced his support for just qualitative parsimony: I subscribe to the general view that qualitative parsimony is good in a philosophical or empirical hypothesis; but I recognize no presumption whatever in favor of quantitative parsimony.

The restriction of Ockham’s razor to just fundamental entities is promoted by (Cameron, 2010), (Schaffer, 2010), and (Sider, 2013). (Schaffer, 2015, p. 647) dubs this version "The Laser" and formulates it as an injunction not to multiply fundamental entities beyond necessity, together with the implicit understanding that there is no such injunction against multiplying derivative entities.

I.2 Explanatory sufficiency

The principle has its origin in the same medieval controversies that spawned Occam's Razor. Ockham's contemporary, Walter of Chatton, proposed that: [I]f three things are not enough to verify an affirmative proposition about things, a fourth must be added, and so on. A related counter-principle was later defended by Kant: The variety of entities should not be rashly diminished (*The Critique of Pure Reason*).

> [T]he grand aim of all science...is to cover the greatest possible number of empirical facts by logical deductions from the smallest possible number of hypotheses or axioms (Einstein, quoted in (Nash, 1963, p. 173)).

Einstein’s quote above respects the principle, when he says, “without having to surrender the adequate representation”.

References

Cameron, R. P. (2010). “How to have a radically minimal ontology.” Philosophical Studies, 151(2), 249–264.

Nash, L. K. (1963). “The nature of the natural sciences.”

Schaffer, J. (2010). “Monism: The priority of the whole.” Philosophical Review, 119(1), 31–76.

Schaffer, J. (2015). “What not to multiply without necessity.” Australasian Journal of Philosophy, 93(4), 644–664.

Sider, T. (2013). “Against parthood.” Oxford Studies in Metaphysics, 8(2013), 237–293.

See also:

Baker, A. (2016). “Simplicity.” In Edward N. Zalta (Ed.), The Stanford Encyclopedia of Philosophy (Winter 2016). Metaphysics Research Lab, Stanford University.
Retrieved from https://plato.stanford.edu/entries/simplicity/

Appendix J
Ontological commitments – technical details

In this appendix, we cover a range of the relevant background detailed technical issues. We also include relevant extracts at the end, as a gateway to the wider literature on this topic.

J.1 Natural language ontology – foundational ontology

One use of ontology is to describe the language (and associated concepts) we use to talk about things – hence a natural language ontology. Friederike Moltmann gives a good outline of this in the extract below; noting, for example, that it "is better to be characterized as the ontology that a speaker *implicitly accepts* when using the language" and so "may include merely conceived objects besides objects that happen to be actual ones."

She also contrasts this with "the ontology of what there really is" which she, following Fine (2017), calls 'foundational ontology'. There, Fine contrasts foundational ontology with what he calls naïve ontology. As Moltmann points out (see extract below I.1.1), "Fine's (2017) term 'naïve metaphysics' thus is misleading" – natural language or descriptive ontology is a better description. Fine (2017) also suggests that "naive metaphysics will be the metaphysics of appearance while foundational metaphysics will be the metaphysics of reality, with the one concerned to discern the nature of the world as it presents itself to us and the other concerned to discern the nature of the world as it is in itself."

Note, in earlier ontological commitment analyses (Borgo (2002), Semy (2004) – extracts below – and Orbst (2010)) there is a similar distinction between descriptive and revisionary ontologies (based upon Strawson's (1959) distinction). This nomenclature then evolved to descriptive and realist ontologies in ROMULUS (2013). However, as you can see from the extracts, this distinction was often expanded to cover with linked distinctions such as between space and time and space-time, and endurants and perdurants.

J.1.1 Friederike Moltmann – Natural Language Ontology

Extract from: Friederike Moltmann (2020) Chapter 13 – *Abstract Objects and the Core-Periphery Distinction in the Ontological and the Conceptual Domain of Natural Language.* Online: https://www.academia.edu/38640396/

13.3 Natural Language Ontology

The following is a brief outline of the discipline whose subject matter is the ontology of natural language, natural language ontology. With its referential noun phrases, which take entities as semantic values as well as with its lexical predicates and constructions that involve entities in other ways, natural language reflects an ontology. This ontology is the subject matter of a particular branch of metaphysics, that of natural language ontology. More specifically, natural language ontology is part of descriptive metaphysics in Strawson's (1959) sense or what Fine (2017) calls 'naïve metaphysics'. Descriptive metaphysics has as its subject matter the ontology reflected in our ordinary judgments. Natural language ontology has as its subject matter the ontology reflected in linguistic intuitions, that is, judgments about the acceptability or grammaticality of natural language sentences and constructions.

What is important about descriptive metaphysics is that it is not about the ontology of what there really is. This is instead the subject matter of a different branch of metaphysics, what Fine calls 'foundational metaphysics'.[4] Descriptive metaphysics and natural language ontology in particular concerns itself with how things appear, given the data, without addressing the question of whether they are real (which is to be addressed only by foundational metaphysics). For natural language ontology; this means that no foundationalist consideration should come into play when positing objects as semantic values, such as assumptions as to whether those objects really exist (in the sense of being fundamental) or what they may be reduced to. More important is what sorts properties the semantic values of referential noun phrases may have, as is reflected (at least to an extent) in the applicability of types of natural language predicates. The domain of objects in the ontology of natural language thus may include merely conceived objects besides objects that happen to be actual ones. (This will also be important for how to make sense of the expandability of the ontological domain of the language through technical or philosophical discourse in Sect. 13.4.)

The subject matter of natural language ontology is not the ontology that ordinary speakers (non-philosophers) naively accept when thinking about what there is. [5] The latter is the subject matter of folk metaphysics, not natural language ontology. What speakers accept when they reflect does not matter for natural language ontology. Natural language ontology rather deals with the ontological categories, notions, and structures that are implicit in language whether or not speakers would accept them upon reflection. The ontology of natural language thus is better to be characterized as the ontology that a speaker *implicitly accepts* when using the language and as such is distinguished from both the reflective ontology of ordinary speakers as well as philosophers and the ontology of what there really is (Moltmann 2017, 2019).

[4] For Strawson (1959), descriptive metaphysics rather contrasts with what he calls ‘revisionary metaphysics’. The aim of revisionary metaphysics, for Strawson, is to conceive of a better ontology than the one we ordinarily accept. (Strawson does not specify further how ‘better’ is supposed to be understood.)

[5] Fine’s (2017) term ‘naïve metaphysics’ thus is misleading, and ‘descriptive metaphysics’ a better term to use for the branch of metaphysics that comprises natural language ontology.

J.1.2 Wonderweb deliverable d15: Ontology Roadmap (2002)

2 Design Options and Ontological Choices

Before addressing specific issues about domain of discourse, basic categories, and their relations [4], it may be important to clarify the general attitude towards ontological analysis, or – in other words – the motivations and the constraints that drive our conceptualization of reality. It comes to no surprise that the design options for building foundational ontologies reflect the main categorical distinctions discussed in philosophy. However, among all the philosophical stances and distinctions, foundational ontologists seem particularly interested in two general attitudes: a) descriptive vs. revisionary, and b) multiplicative vs. reductionist.

(a) A descriptive ontology aims at capturing the ontological stances that shape natural language and human cognition. It is based on the assumption that the surface structure of natural language and the so-called commonsense have ontological relevance. As a consequence, the categories refer to cognitive artifacts more or less depending on human perception, cultural imprints and social conventions. Under this approach, there are no major restrictions on the postulation of ontological categories because overall philosophical or scientific paradigms are neglected. This attitude stands in contrast to the revisionary approach. The revisionist considers linguistic and cognitive issues at the level of secondary sources (if considered at all), and does not hesitate to paraphrase linguistic expressions (or to re-interpret cognitive phenomena) when their ontological assumptions are not defensible on scientific grounds. The following example should make this contraposition clear. Commonsense distinguishes between things (spatial objects like houses and computers) and events (temporal objects like bank transfers and computer repairs). In the wake of relativity theory, how- ever, time is viewed as another dimension of objects on a par with the traditional spatial dimensions. Considering the consequences of this scientific theory (or theories), some philosophers and computer scientists have come to believe that the commonsense distinction between things that are and things that happen should be abandoned in favor of a unified viewpoint. According to these revisionist researchers, everything extends in space and time, and the distinction between things and events is an (ontologically irrelevant) historical and cognitive accident. This example shows that a revisionary ontology is committed to capture the intrinsic nature of the world by providing structures that are independent from the conceptualizing agents.

[4] A. Gangemi, N. Guarino, C. Masolo, and A. Oltramari. Understanding top-level ontological distinctions. In IJCAI-01 Workshop on Ontologies and Information Sharing, pages 26-33, Seattle, USA, 2001. AAAI Press. http://SunSITE.Informatik.RWTH-Aachen.DE/Publications/CEUR-WS/Vol-47/

J.1.3 MITRE: Toward the Use of an Upper Ontology for U.S. Government and U.S. Military Domains (2004)

3.2.1 Descriptive vs. Revisionary

Descriptive and revisionary ontologies [67], [49] are based on ontological stances or attitudes towards the effort of modeling ontologies, i.e., how one conceptualizes the world and what an ontological engineering product is or should be. A descriptive ontology tries to capture the more commonsensical and social notions based on natural language usage and human cognition, emphasizing the agent who conceives and deemphasizing scientific and philosophical considerations. A revisionary (sometimes called prescriptive) ontology, on the other hand, does emphasize (or even, strictly adheres to) the scientific and philosophical perspectives, choosing to base its constructs and modeling decisions on scientific theories and a philosophical stance that tries to capture the world as it really is (it prescribes the world), and not necessarily as a given historical agent conceives it to be. A revisionary ontology therefore says that its modeling constructs are about real things in the world as it is.

In practical terms, all of the constructs in a revisionary ontology will be space-time objects, i.e., necessarily having temporal properties; in a descriptive ontology, that will not be the case. In the latter, entities (sometimes called endurants, but perhaps better called continuants) such as “hammer” and “tank” that have only incidental temporal properties and events (processes, actions, activities, etc., sometimes called perdurants, but perhaps better called occurrents) such as “attacking” and “cashing a check” that have explicit temporal properties, are modeled with or without those temporal properties, respectively. Often in natural language there are two correlated forms/usages that express the distinction: the nominal and the verbal. A nominal (noun) “attack” is expressed as in “The attack on the enemy began at 600 hours.” A verbal (verb) “attacked” is expressed as in “We attacked the enemy at 600 hours.”

[67] Strawson, P.F. 1959. Individuals: An Essay in Descriptive Metaphysics. London: Methuan University Press.

[49] Obrst, L., H. Liu, R. Wray. 2003. Ontologies for Corporate Web Applications. Artificial Intelligence Magazine, special issue on Ontologies, American Association for Artificial Intelligence, Chris Welty, ed., Fall, 2003, pp. 49-62.

J.2 Extensional and intensional criteria of identity

J.2.1 Making a single broad choice

Typically, top-level ontologies will adopt one of these broad choices for their main entities, though they may adopt the alternative for selected entities.

Typically, ontologies with an extensional strategy pick very general groups as the bearers of criteria of identity. BORO, ISO 15926, HQDM and IDEAS are examples of this. They include types/sets with an extensional membership criterion and particulars with a spatio-temporal parts criterion (often simplified to spatio-temporal extension).

Typically, ontologies with an intensional strategy pick less general, more specific groups, often known as sortals, as the bearers of criteria of identity, as the essential characteristics that inform the object's identity are less general. BFO, DOLCE and UFO are examples of this.

J.2.2 Bearers of specific identity criteria

Often entities are labelled with their identity choice; so extensional entities and intensional entities. As the examples above show, for various reasons (including pragmatism) these choices are not made for each individual entity, rather they are made for groups of entities, where the choice applies to all members of the group. Often the entities are grouped by their specific identity criteria, where different groups have different criteria – the groups bear these identity conditions. Normally these groups are disjoint, as otherwise there is a requirement for meta-rules to arbitrate for objects that fall under two different criteria of identity (Guarino (2000) offers this meta-rule: Properties carrying incompatible Identity Criteria are necessarily disjoint).

J.2.3 Taming intensional promiscuity

We can illustrate the core difference between the two ways of capturing identity with an example – showing the potential promiscuity of the intensional approach. As already noted, general objects have an extension; the objects that instantiate the object. Under extensional criteria, this extension determines its identity. Under intensional criteria, it does not. Consider the earlier example of an equilateral triangle – defined as having equal sides. Now consider equiangular triangles – with three equal angles. One can prove that these definitions are equivalent. However, they are different, so it makes sense to say they are different intensions, even though one can logically show they always have the same extension. One can easily develop more and more baroque definitions that have this same extension. Under a permissive intensional strategy, these would all be different objects. One is then faced with deciding what counts as a different meaning. As far as we have been able to determine, in so far as the candidate top-level ontologies have adopted an intensional strategy, they have adopted a permissive one.

J.2.4 Extensionality as varieties of formal grounding

One can view extensional identity as a form of grounding. In the case of a set, the set is composed of and grounded and dependent upon its members and their identity. In the case of material objects, one could argue that they are composed of and grounded and dependent upon their parts (and their identity). The identity of the extensional object is grounded in the formal collection of the identities of the grounding objects.

In both cases, the composing and so grounding is simpliciter. Tuples – ordered lists of objects – where the order is important, and repetition is allowed – show this is not always the case. For example, <a, b, a> is a valid tuple where 'a' is a component twice. The tuple <b, a, b> has the same base components, a and b, but different orders (and repetitions) of them. In this case, the formal extension (and so criteria of identity) needs to recognise both order and repetition. (For more detail see Partridge (2019))

J.2.5 A solid basis for two types of extensional identity emerges

Two development have changed the capability of extension to capture identity.

The first related to the extension of material objects. René Descartes, in Principia Philosophiæ (1644), introduced the notion of material objects as res extensa – as extended in space. This notion took hold. For example, John Locke, in An Essay Concerning Human Understanding (1690), defined extension as "only the Space that lies between the Extremities of those solid coherent Parts" of a body. However, this raised questions about compenetration, where to two or more extensions occupying the same space at the same time – a problem raised by the Ancient Greeks. In the early 20th century, physicists developed a new way to think about extension, as space-time. Philosophers developed this as a more solid basis for identity. At one time, two things might be in the same place but not at a later time, having different spatio-temporal extensions. One could accommodate both (spatial) compresence and identity.

The second related to the extension of general entities. It was normal, until the middle of the 20th century to regard the extension of general terms to be the actual objects that fell under the term. Work done by Kripke and then Lewis (see (Yagisawa, 2005)) on possible worlds provided an extensional way of capturing modal intensions (the extensions over possible worlds were sometimes called 'intentions') and so tracking identity.

These two developments provided a solid basis for extensional identity.

J.2.6 BFO example of universals' non-extensional (intensional) identity criteria

We have not been able to find any specific mention of identity criteria in the BFO manuals. However, the extract below (from the BFO manual) even though it does not mention identity criteria directly shows that it distinguishes between the universal and its extension in an intensional way.

> "2.6 Universals and classes
>
> Universals have instances, which in BFO are in every case particulars (entities located in specific regions of space and time). Universals also have extensions, which we can think of as collections of their instances. (Traditionally the extension of a concept is viewed is set-theoretical terms as the set of all the things that fall under the concept.) Such extensions fall outside the scope of this specification, but it is important for the understanding of BFO that the distinction is recognized. It implies further distinctions not only between universals and their extensions but also between universals and classes in general, including arbitrary classes such as: {the moon, Napoleon, redness}."

Basic Formal Ontology 2.0: SPECIFICATION AND USER'S GUIDE
June 26, 2015

J.2.7 OntoClean discussion of Intensional Identity

Other ontologies address the question of intensional identity directly. The extract below is from the Wikipedia entry for OntoClean ("OntoClean – Identity," n.d.) – a tool developed by the creators of DOLCE. This clearly talks about intentional criteria of identity – saying:

> "Identity criteria should be ***informative***, they should help us and others understand what a class means." It also offers two possible criteria for a triangle: it "... can be identified by the length of its three sides, or by two sides and an interior angle, etc. This says a lot about what is ***intended*** by the triangle class here, e.g. the same triangle could be in many places at the same time." It seems to be relaxed that these are different objects as they have different definitions.
>
> Identity
>
> Identity is fundamental to ontology, and especially to information systems ontologies. Identity is well known in metaphysics and in database conceptual modeling. In the latter case, it is an accepted best practice to specify a ***primary key*** for rows in a table. If "two" rows have identical primary keys, they are considered the same row.
>
> More importantly for ontology are questions of identity that expose the existence of, or at least the need to represent, other entities. Here the issue at stake is finding the conditions under which a proposed entity would be both the same and different. The classic example is an amount of clay that is shaped into a statue. If you use the ***same*** clay but reshape it into a ***different*** statue, is it the same entity? If so, how could it be ***different***? If not, how could it be ***the same***. In conceptual modeling, it is understood that when such an ambiguity arises, one should treat it as two different entities to account for a situation where one changes and the other stays the same.
>
> In OntoClean, ***identity criteria*** are associated with, or carried by, some classes of entities, called ***sortals***. A sortal is a class all of whose instances are identified in the same way. In information systems, these criteria are often extrinsic, like a social security number or universally unique id, which is not interesting from an ontological point of view. Identity criteria should be ***informative***, they should help us and others understand what a class means. A triangle, for example, can be identified by the length of its three sides, or by two sides and an interior angle, etc. This says a lot about what is ***intended*** by the triangle class here, e.g. the same triangle could be in many places at the same time. Someone else may have an ontology in which the triangle class has different identity criteria, such that different drawings are always different triangles, even if they are the same size. Identity criteria (and OntoClean, for that matter) do not tell you that one of these definitions of triangle is right or wrong, just that they are different and thus that the classes are different.
>
> Identity criteria and sortals are intuitively meant to account for the linguistic habit of associating identity with certain classes. In the classical statue and clay example, we naturally say "the same ***clay***" or "the same ***statue***", indicating that there are identity criteria that are peculiar to each class.
>
> ***Being a sortal*** is the first OntoClean metaproperty, indicated with the **+I** superscript (-I for non-sortals) on a class in the original notation. +I (but not -I) is inherited down the class hierarchy, if a class is a sortal then all its subclasses are as well.

J.3 Indexicality

In (Galton, 2018, pp. 37–8): "3.3. Indexicality: Past, Present, and Future: Should an ontology of time include reference to the present moment? As noted above, if 'right now' is seriously proposed, as in [1], as an example of a zero-dimensional temporal region, then the implicit answer is that it should. ... If this is right, then it would be reasonable for a formal ontology that claims to provide an account of objective reality to include pastness, presentness, and futurity as attributes of times, though it is hard to see how to integrate this into the overall temporal framework, and as far as I am aware none of the currently existing formal upper ontologies has attempted to do this. Until such time as this is done, it would be best to steer clear of problematic entities such as 'right now' as examples of instants."

As I noted above and Galton does here, having a separatist spatial structure provides a good starting point for identifying the present.

While Galton is right that many top-level ontologies do not have a way of representing the present, he is not right in that they all do not. There are top-level ontologies that include indexicality – BORO is an example. See Partridge, 1996: Business objects: re-engineering for re-use. Chapter 8 – Section – 4 – The time-based 'consciousness' of information systems – which discusses a 'now' and 'here' object. In a later paper (Partridge, 2018), a more sophisticated way of handling indexicality using agentology is described. This has already been discussed in the FDM forum – https://groups.google.com/d/forum/uk-ndt-fdm. (This is a public forum, join on request). The paper suggests that there is an agentology layer indexed to the agent/system under the ontology.

Appendix K
Glossary

Term	Description
formal ontology	The term formal ontology; itself was coined by Edmund Husserl in the second edition of his Logical Investigations (1900-01), where it refers to an ontological counterpart of formal logic. Formal ontology for Husserl embraces an axiomatized mereology and a theory of dependence relations, for example between the qualities of an object and the object itself. Formal; signifies not the use of a formal-logical language, but rather: non-material, or in other words domain-independent (of universal application)." From https://en.wikipedia.org/wiki/Formal_ontology#Historical_background
ontology	Jonathon Lowe in The Oxford Companion to Philosophy described ontology as "the set of things whose existence is acknowledged by a particular theory or system of thought."
material, physical and concrete	Objects consisting of matter (or energy). Not abstract. Three terms used interchangeably.
ontological model	A model of an ontology. An ontology can have multiple models.
ontological architecture	The fundamental structures of the ontology. Typically includes the major choices of ontological commitment. Can function as a blueprint for developing an ontology.

References

Ainsworth, T. (2020). “Form vs. Matter.” In Edward N. Zalta (Ed.), The Stanford Encyclopedia of Philosophy (Summer 2020). Metaphysics Research Lab, Stanford University. Retrieved from https://plato.stanford.edu/entries/form-matter

Cameron, R. P. (2010). “How to have a radically minimal ontology.” Philosophical Studies, 151(2), 249–264.

Casati, R. & Varzi, A. C. (1999). Parts and places: The structures of spatial representation. MIT Press.

Galton, A. (2018). “The Treatment of Time in Upper Ontologies.” In FOIS (pp. 33–46). Retrieved from http://ebooks.iospress.nl/publication/50243

Gilmore, C., Costa, D. & Calosi, C. (2016). “Relativity and three four-dimensionalisms.” Philosophy Compass, 11(2), 102–120. Retrieved from https://philpapers.org/rec/GILRAT-10

Hetherington, J. & West, M. (2020). “The pathway towards an Information Management Framework-A ‘Commons’ for Digital Built Britain.”

ISO. (2019). “Information technology – Top-Level Ontologies (TLO) – Part1: Requirements.”

Kant, I. (1964). Critique of pure reason. London, New York, E. P. Dutton & Co.: J. M. Dent & Sons Ltd.;

Locke, J. (1975). An Essay Concerning Human Understanding. Oxford: Clarendon Press.

Lowe, E. J. (1998). The Possibility of Metaphysics: Substance, Identity, and Time. Oxford: Clarendon Press.

Nash, L. K. (1963). “The nature of the natural sciences.”

“OntoClean – Identity.” (n.d.). Retrieved from https://en.wikipedia.org/wiki/OntoClean#Identity

Partridge, C. (1996). Business objects: re-engineering for re-use. Butterworth-Heinemann.

Partridge, C., de Cesare, S., Mitchell, A., Gailly, F. & Khan, M. (2017). “Developing an Ontological Sandbox: Investigating Multi-Level Modelling’s Possible Metaphysical Structures.” Retrieved from https://www.academia.edu/35222673

Partridge, C., de Cesare, S., Mitchell, A., Leon, A., Gailly, F. & Khan, M. (2018). “Ontology then Agentology: A Finer Grained Framework for Enterprise Modelling.” In Proceedings of the 6th International Conference on Model-Driven Engineering and Software Development (MODELSWARD 2018)
(pp. 454–463). Funchal, Madeira, Portugal: SCITEPRESS – Science and Technology Publications. Retrieved from https://www.academia.edu/35335427

Partridge, C., de Cesare, S., Mitchell, A. & Odell, J. (2016). “Formalization of the classification pattern: survey of classification modeling in information systems engineering.” Software & Systems Modeling, 1–37. Retrieved from https://link.springer.com/article/10.1007/s10270-016-0521-5

Partridge, C., Mitchell, A., Loneragan, M., Atkinson, H., de Cesare, S. & Khan, M. (2019). “Coordinate Systems: Level Ascending Ontological Options.” In 2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C) (pp. 78–87). Retrieved from https://www.academia.edu/40354620

Schaffer, J. (2010). “Monism: The priority of the whole.” Philosophical Review, 119(1), 31–76.

Schaffer, J. (2015). “What not to multiply without necessity.” Australasian Journal of Philosophy, 93(4), 644–664.

Sider, T. (2013). “Against parthood.” Oxford Studies in Metaphysics, 8(2013), 237–293.

Strachey, C. (2000). “Fundamental concepts in programming languages.” Higher-Order and Symbolic Computation, 13(1-2), 11–49.

Varzi, A. (2019). “Mereology.” In Edward N. Zalta (Ed.), The Stanford Encyclopedia of Philosophy (Spring 2019). Metaphysics Research Lab, Stanford University.
Retrieved from https://plato.stanford.edu/entries/mereology

Yagisawa, T. (2005). “Possible Objects.” In Edward N. Zalta (Ed.), The Stanford Encyclopedia of Philosophy. Metaphysics Research Lab, Stanford University.
Retrieved from https://plato.stanford.edu/entries/possible-objects/

Acknowledgements

Lead Authors:

Chris Partridge
Andrew Mitchell

Authors:

Al Cook
David Leal
Jan Sullivan
Matthew West

Contributors:

Hayden Atkinson
Hugh Boyes
Sergio de Cesare
Peter El Hajj
Salvatore Florio
César González-Pérez
Ian Gordon
James Harris
Derwen Hinds
Mesbah Khan
Steven Kraines
Alexandra Luck
Neil Thompson
Oscar Xiberta Soto

Partridge C, Mitchell A, Cook A, Leal D, Sullivan J and West M, CDBB, 2020
A Survey of Top-Level Ontologies - to inform the ontological choices for a Foundation Data Model
DOI: https://doi.org/10.17863/CAM.58311

About the Construction Innovation Hub

The Construction Innovation Hub brings together world-class expertise from the Manufacturing Technology Centre (MTC), BRE and the Centre for Digital Built Britain (CDBB) to transform the UK construction industry.

With £72 million from UK Research and Innovation's Industrial Strategy Challenge Fund, and working around the four core themes of Value, Manufacturing, Assurance and Digital, we are changing the way buildings and infrastructure are designed, manufactured, integrated and connected within our built environment.

We are a catalyst for change. We are driving collaboration to develop, commercialise and promote digital and manufacturing technologies for the construction sector. We are helping build smarter, greener and more efficient buildings much faster and cheaper than we currently do.

Research is helping us understand how the industry needs to change in terms of skills, product standards, capacity and innovation. This is combined with an academic programme to create the security-minded frameworks and rules that will underpin the future digital built environment and grow exports for UK know-how.

We are working closely with other initiatives as part of the Government's Transforming Construction challenge programme. Through collaboration across the sector, we can provide a better-built environment for future generations.

Further information
For further details about the Construction Innovation Hub
or the Value Toolkit, please contact:

Construction Innovation Hub
info@constructioninnovationhub.org.uk
www.constructioninnovationhub.org.uk

The Construction Innovation Hub is funded by UK Research and Innovation through the Industrial Strategy Challenge Fund

The Construction Innovation Hub is a partnership between:

constructioninnovationhub.org.uk
#TransformingConstruction