

EntityModelling

John Cartmell

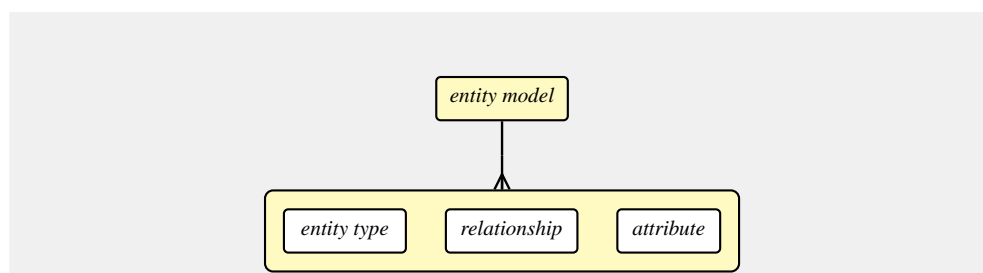
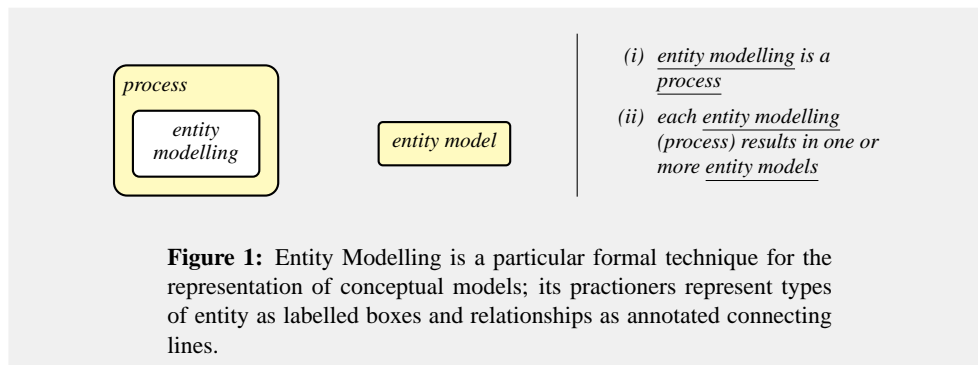
November 24, 2015

1 A Framework for Knowledge

Entity modelling is a technique and a notation for describing and communicating *what is* in the world and we set out to present it here from first principles. More usually, its purpose is given as the structuring of data to be stored in information systems; we extend the notation and describe its use in information systems development but we do not take this use as the starting point for the presentation nor as a subsequent *raison d'être* because to do either would be to hide a greater potential. We take the wider view that the purpose of an entity model is to provide a framework for knowledge and that entity modelling is a form of conceptual modelling - a technique for the elaboration of concepts. We describe the technique from first principles and relate to other meta-conceptual systems. We extend the notation from that commonly used in information systems development by the introduction of additional meta-conceptual machinery. This is done in a way heavily influenced by consideration of conceptual patterns brought to the fore in the branch of mathematics known as category theory.

To illustrate at the outset what we mean by conceptual modelling, consider the experience of reading into a new subject area and finding terms which seemingly have specific patterns of usage, and, it must be assumed, contextual meanings, but which patterns and meanings are unfamiliar to us. In so reading we are drawn into a systematic and iterative arrangement and a classification of the unfamiliar terms; in this process we will likely distinguish terms for individual things, for types or classes of things, for relations between things and also quantitative and adjectival terms. The mental process we follow will build for us a conceptual model. Entity modelling is a particular technique for expressing such models and, indeed, it is a technique and a notation used by information scientists seeking to represent and computerise the sometimes unfamiliar domains in which they work. When asked whether they understand a particular topic, an entity modeller might well affirm they do only if they can sketch an entity model that frames the topic.

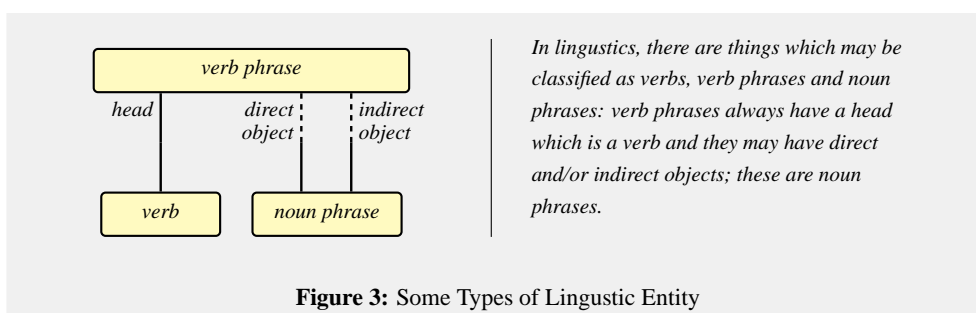
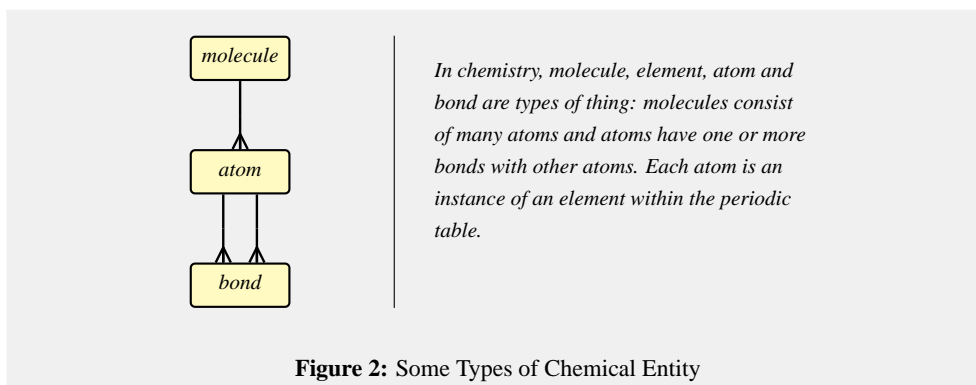
Thinking about understanding a new area and its language, though 'things themselves' are the subjects of the text, the language is of the types of things, the relations among the different types of things and the properties that can meaningfully be attributed to them. This brings us to the three core elements of an entity model: entity types, relationships and attributes. In fact an entity model is little more than a structured list of such things.



The premises of Entity Modelling have in them a pragmatic answer to the question what is knowledge? Knowledge, according to the premises of Entity Modelling, is knowledge about things. For any ‘thing’ the knowledge that we can have of it is a sum of elementary pieces each of which is either the fact of an attribution, by which is meant a property a thing has inherent in itself, or else the fact of a relationship with another thing. More precisely, we can have knowledge of the type of a thing and knowing its type is to know both the kind of attributions which may be made of it and the relationships in which it may participate. This is the theory of knowledge according to the entity modeller; it is also the basis of information modelling and therefore it is a pragmatic view of what knowledge is - it is that which can be represented as information in a structured form suitable for representation in a computer system.

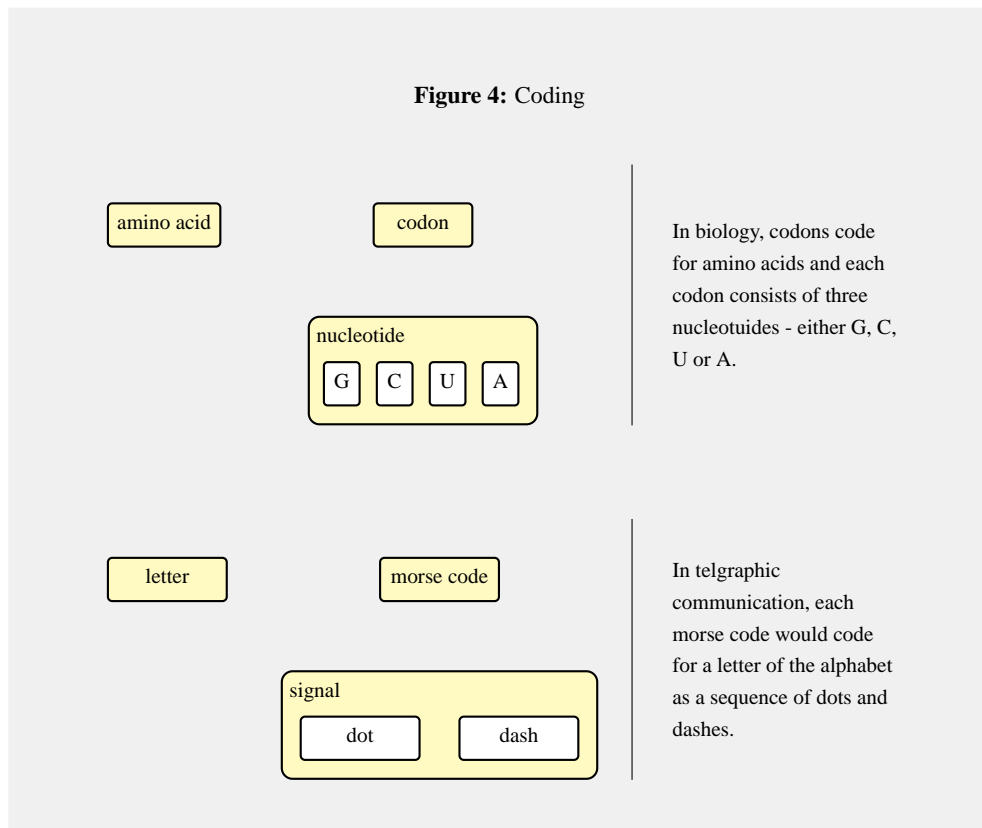
2 Domain of Discourse

In essence an entity model establishes, or at very least proposes, a domain of discourse. The domain can be relatively familiar consisting of entities such as ‘house’, ‘street’, ‘person’ etc. or can be technical consisting of entities such as constituent parts of grammar, for example ‘noun phrase’ and ‘verb phrase’ and so on, or constituent parts of chemical structure such as ‘atom’ and ‘covalent bond’. The primary function of entity modelling is to define and present such domains of discourse and to do so in a way that eliminates error and ambiguity.



Sometimes the domain of discourse can be abstract such as when we use a metaphor based language term such as ‘network’ in the noun phrase ‘social network’ and by which means we seek to identify a conceptual pattern and bring to mind imagery of points and lines independently of what the points represent - whether that be social entities, railway stations or atoms. It is one of the functions of entity modelling to present and propose such deliberately abstract domains of discourse. In this one mentioned the principal entities will be ‘point’ and ‘line’. Breakthrough abstractions in science often move into language as metaphor and this is just because abstractions have a metaphorical shape which gives them utility beyond their origins; by virtue of being abstract they are reusable and ever applicable in new situations. As diagrams of concepts, entity models give visual shape to this metaphorical shape of our abstractions.

Figure 4: Coding



3 Data

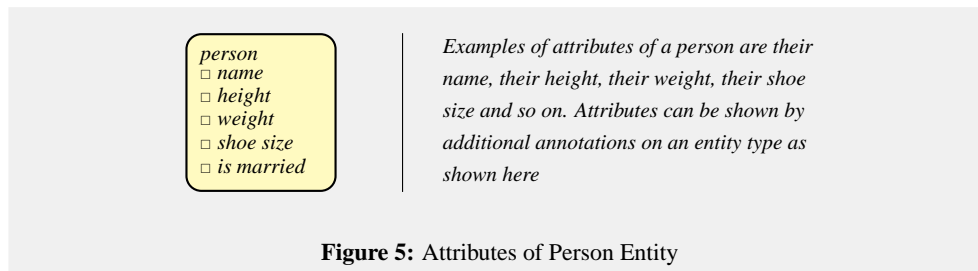
When we call to mind data, then we think of names, quantities, monetary values, addresses, dates, temperatures, geographical coordinates, and so on. Now, such items of data as these convey information only within specific contexts and when attributed to subjects at hand. A temperature, a colour, a price, a height, a distance all these tell us nothing less they be the temperature, the colour, the price, the height or the distance of some thing. We can paraphrase in the language of entity modelling and say they tell us nothing less they be attributed to an entity.

As stored within information systems then the individually represented items - the names, colours, quantities, the monetary amounts, the dates, etc - in the language of entity modelling, are said to be the values of attributes. Thus an actual name like "John Smith" is said to the value of a 'name' attribute of a person entity. and it is clear that computer programs are effective only in so long as the data items they manipulate are intended and understood as attributes of subject entities. It follows that to have an effective information system we must first have agreed types of subject entity and we must first have agreed the attributes of each of these types of subject entity. In this agreement we agree the data content of the program and in turn we agree its subject matter.

Now attribute values are not stored alone and independent of one another but grouped together by the subject entities to which they apply. So it is that to be able to model the data held in information system there is third element - that of an attribute. The concept of at-

tribute sits alongside those of entity and relationship to form the basis of entity modelling. Each attribute is posited as a named property associated with a specific or general type of entity (i.e. with a particular species or with all species within a genus).

An attribute of an entity gives information about the entity above and beyond the information given by its relationships.



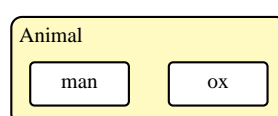
4 Types Specific and General

4.1 Aristotles Categories

In the statement ‘man and the apes are descendent from a common ancestor’ - it is clear that singular ‘man’ is being used to denote a specific type which in biology, and as far back as Aristotle’s Categories, is termed a species, and that plural ‘apes’ is denoting a more general class of thing which following Aristotle we might call a genus or in biology are said to be ‘a classification of a higher rank’ and which classifications includes genera, families, orders and kingdoms. As a reader one does not need know the individual species of apes but can infer that a multiplicity of species is implied simply by noting the use of the plural form. In entity modelling both such specific and general types of thing are represented; these are *species* and *genera* not just in the strict sense used in biological nomenclature but in the more general sense that we find in translations of Aristotle’s Categories, a species is a specific type such as the type ‘man’ and a particular individual man is said to be predicated by that species. A genus, on the other hand, is a more general type such as ‘animal’ and ‘ape’.

For if any one should render an account of what a primary ousia is, he would render a more instructive account by stating the species than by stating the genus. Thus, he would give a more instructive account of an individual man by stating that he was a man than by stating that he was an animal, for the former description is peculiar to the individual in a greater degree, while the latter is too general.

In Aristotle’s description both the individual man and the species man are predicated by the genus animal, and so to, by example, the species ox is predicated by the genus animal. From what we have already said this we will represent thus:



In some ways, such diagrams as these show similarity to Venn diagrams. This one can be interpreted as showing the set of men and the set of oxen included in the set of animals. However, to the entity modeller, there is no set of all men, nor of all oxen, nor of all animals for the question is not ‘what exists?’ but ‘what types of things exist and what can be said of them?’. The diagram can be interpreted as saying is ‘what can be said of animal’ can be said of ‘man’ and of ‘ox’ also.

Aristotle says it like this:

Whenever one thing is predicated of another as a subject, all things said of what is predicated will be said of the subject also. For example, man is predicated of the individual man, and animal of man; so animal will be predicated of the individual man also - for the individual man is both a man and an animal.

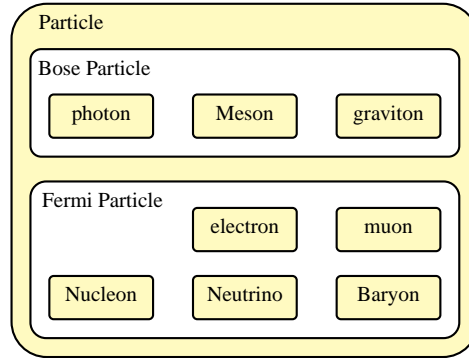
Subsequently, after the time of Aristotle, used in biological nomenclature, the term genus adopted a more specific meaning, in contradistinction to use of the term species for the lowest rank in the system - individuals of the same species varying in minor ways and able to interbreed - the term genus became used for a group of related species, the genera strictly forming just the second rank in a multi-ranked system. For the purposes of entity modelling we do not use the genus in its specific taxonomic sense but nonetheless have recourse to a convention from taxonomy by which the names of species are written with a leading lower case letter (e.g. sapiens) and the name of the genus is written with a leading upper case letter (e.g. Homo). For us the names of general types of things will be capitalised or have a leading capital letter whereas the names of the most specific types of things will be written entirely with lower letters.

4.2 An example from Physics

A more technical example is given by the types of particle discussed in Feynman’s Lecture Notes on Physics:

Particles which interfere with a positive sign are called Bose particles and those which interfere with a negative sign are called Fermi particles. The Bose particles are the photon, the mesons, and the graviton. The Fermi particles are the electron, the muon, the neutrinos, the nucleons, and the baryons.

As in the usage ‘man and the apes’ singular and plural terms are used in this passage to distinguish between specific types of things and more general classes of things. The author’s respective use of singular and plural terms inform as to which are the fundamental types of particle, the species, and which the related families, the genera. In diagramming this in an entity model, instead of retaining the plural form, so to speak, we may capitalise and underline the genera and so arrive this diagram:



4.3 Examples from Linguistics

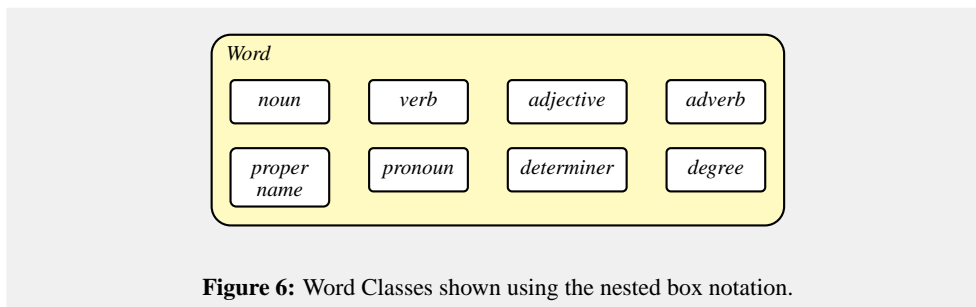
In linguistics the entity type ‘word’ is generally represented as a generalisation of more specific types often referred to as word classes. These include *noun*, *verb*, *adjective* and so on as shown in figure 6 and these are illustrated in table 1.

The basis for the recognition of word classes in linguistics is the observation that certain words can be freely interchanged in sentences without altering the acceptability of the sentence grammatically. For example we can apply substitutions replacing various words of the sentence *I received beautiful flowers for my birthday* by randomly chosen other words and some will deliver equally grammatical sentences and some will not. For example we can replace ‘beautiful’ by ‘ugly’ or ‘red’ or ‘expensive’ without losing sentence structure whereas we cannot replace by ‘the’ or ‘very’ or ‘at’. So the words ‘beautiful’, ‘ugly’, ‘red’, ‘expensive’ are of the same class *adjective* and the words ‘the’, ‘very’, ‘at’ are not of this class - in fact the word ‘the’ is classed as a *determiner*, the word ‘very’ as a *degree* word and the word ‘at’ as a *preposition*.

class	abbreviation	examples
noun	N	<i>athelete, house, race, record, stream, water</i>
pronoun	Pro	<i>I, you, he, she, we, they</i>
determiner	Det	<i>a, the</i>
verb	V	<i>arrive, run, set</i>
auxiliary	Aux	<i>had, will</i>
preposition	Prep	<i>at, by, from, in, to</i>
prepositional specifier	Pspec	<i>close, right, straight, three seconds</i>
adjective	A	<i>fierce, long, new, red, right, rosy, silk, young</i>
general adverb	Adv	<i>abruptly, brightly, clearly, quickly</i>
degree adverb	Deg	<i>more, most, quite, rather, so, too, very</i>

Table 1: Types of word and abbreviations used.

It is common to use abbreviations to identify the word classes; as to how many there are then it has to be said that they cannot be enumerated unequivocally; linguist C.C Fries defined nineteen types as the nineteen parts of speech of English in 1952 (he also distinguished content bearing types of word: *nouns*, *verbs*, *adjectives* and *adverbs* from function types such as *prepositions*, *determiners* and *coordinating conjunctions*). For our purposes here we will use the classes and the abbreviations shown in table 1.

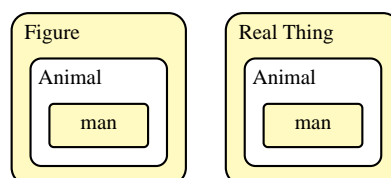


5 Biological Nomenclature - Equivocal Naming

In biological nomenclature the species name alone is not always enough to uniquely identify a type of thing less it be used alongside of the genus name - this system of naming, using the genus name alongside the species name, being called binomial and having been introduced by Linnaeus. Resonant and antecedent to this can be seen in the very first section of Aristotle's Categories which is the about the proper delineation of the of types of things:

Things are said to be named 'equivocally' when, though they have a common name, the definition corresponding with the name differs for each. Thus, a real man and a figure in a picture can both lay claim to the name 'animal'; yet these are equivocally so named, for, though they have a common name, the definition corresponding with the name differs for each. For should any one define in what sense each is an animal, his definition in the one case will be appropriate to that case only.

The appropriate diagram to fit Aristotle's text would seem to be this:



With such a configuration of types you might suppose a trinomial notation is required - or... simply a diagram and the ability to point at it. And this is the point of such diagrams or at least a very good part of it.

6 Entity Modelling and the Esoteric

Entity modelling has the questions 'What is?' and 'What can be said of it?' at its heart. We can define it to be the process of defining what can be predicated of entities and it necessarily seems to embrace, in passing, many questions of philosophy, specifically of Ontology, the branch of metaphysics dealing with the nature of being.

In ancient times, Aristotle had used the Greek word *ousia* (being) to describe the subject, to which predicates are ascribed. (The Philosophy of Aristotle, D.J. Allan, pg 104). Traditionally in translation the Greek *ousia* has been rendered as substance, a term with broad connotations whereas the Latin word *ens* which is, as is also the Greek *ousia*, the present participle of the verb *to be*, yielded our modern English word entity defined as:

entity n. things existence, as opposed to its qualities or relations; thing that has real existence. (Concise Oxford Dictionary)

or as:

entity n. (the quality of having) a single separate and independent existence. (Longman Dictionary of Contemporary English).

The term entity, as we use it here in the term ‘entity modelling’, was introduced into information science by Chen in 1966. An entity for our purposes is simply something about which it is possible to have knowledge and which can be counted. Thus we can have fictional entities such as characters from a book or entities whose state of existence we can debate such as the number zero or the transcendental number π .

In Aristotelian ontology, as outlined in Categories, there are ten genera of being. The first genus of being, *ousia*, is of two types: primary and secondary. Primary *ousia* are individual things - they are our entities, secondary *ousia* are classes of things, or the genera and species of things - for us these are entity types. The other genera of being (quantity, quality, relation, place, time, position, state, action, affection) are properties inherent in the primary *ousia*.

Though Metaphysics has such ancient origins, there is an intersection of concerns between it and a most practical and modern discipline, namely of software engineering and the programming of computer based systems. The overlap exists because metaphysics has as its subject matter what is most general about things in general - not just physical properties of physical things - and the software development discipline starts with the representation of the very same generalities. So whereas most computer programs have as their subjects, everyday if not concrete and physical things, things such as people, accounts, orders, contracts, airline bookings, and so on; other computer programs have as their subjects programs themselves, mathematical functions, relationships in general, not particular, types of things as distinct from the things themselves, and so on.

The ontology of computer data is myriad: computer databases hold data, but also data about data, and data about data about data; the referentially rich, and it sometimes seems, tottering, structures of data involved in real working software can seem unreal; as to can the arguments made by metaphysicians and the positions they hold to. Distinctions are made in the modelling of data, for the most practical purposes and to achieve the ends of programming, which rendered in metaphysical discussion could be comfortably described as esoteric.

7 Context, Composition, Reference and Scope

In entity modelling we shall see that two seeming different aspects of representing what is:

1. the representation of entities as wholes or composite entities having parts of which they are composed
2. the modelling of context for which entities depend for their logical existence

are two ways of looking at the same distinction and that in this way there is a duality between composition and context. For example we can say that the uses of the term ‘intra’ in contrast to ‘inter’ in the adjective ‘intranational’ and like constructions, derive applicability from context dependent entities having relationships within (intra) and across (inter) contexts. Equally we can say that these uses depend on existence of recognised composite entities (‘nations’) and relationships between parts of the same composites (‘intra’) and between parts of different composites (‘inter’).

Modelling wholes and parts is equivalent to modelling context in which things exist. It is achieved by distinguishing composition relationships to be described in Chapter ?? - those between wholes and parts - from relationships for which we use the term reference relationship as described in Chapter ?. It makes sense to ask of a reference relationship whether it is intra- or inter- any of the contexts in which its participating entities are defined to exist. Answering this question is said to be scoping the relationship and the answer itself is said to be the scope of the relationship. We will describe a distinction between composition relationships and reference relationships and see that composition relationships do not have defined scopes, rather that they serve to define contexts and therefore scopes in the first place. In summary, reference relationships have defined scopes and composition relationships do not.

8 The Absolute

We use the term ‘the Absolute’, which has a varied history in metaphysical writing, to mean the whole of everything and, equally, the context for everything. In giving two meanings we are following a rich tradition for definitions abound. The term was central to much of the philosophy of G.W.F Hegel - in Shorter Logic section 87, among a number of other definitions, we find:

...the Absolute is the Nought...The Nothing which the Buddhists make the universal principle, as well as the final aim and goal of everything, is the same abstraction.

whereas in Phenomenology of Spirit in section 20 we find:

The True is the whole.

and in section 75 of the same we find:

...the Absolute alone is true, truth alone is absolute.

Some sense can be made of these statements when we consider from a logical perspective and from the point of view of information theory. We can illustrate by considering the, at first sight at least, puzzling fact that in some programming languages (such as ML) there is an in-built concept with the name ‘Unit’ and described as a singleton type, whereas in some other earlier programming languages (Algol68, C) the name given to the very same concept has been ‘void’. The apparently striking disparity in naming, the one versus the zero, has come about as a result of the concept being named on the one hand on the basis of the number of things predicated by the type (i.e the number of things we can say are of that type), which is exactly one, and on the other hand based on the number of bits (binary digits) of information carried in communicating a member of the type, which is precisely zero. We get the name ‘unit’ from one point of view and the name ‘void’ from the other. In this way we can say of the Absolute that it is the whole of everything. From it being the whole we can say there is

only one of the type. From there being only one of the type we can say that it's information content is zero. If you present to me the absolute you present me with nothing. Like the true it can be assumed in all contexts for it carries nothing new with it. This is the logic of the absolute.

The whole of a modelling situation can be considered a single composite and this is both 'the ultimate whole' when considered as a composite and equally 'the absolute' when considered as a context. If what we have said above can be summarised as saying that there is a duality between context and composition then in this duality 'the whole' and 'the absolute' are duals: they are the same logical entities.

Another useage that we have is to speak of concepts that are absolute. What we usually mean by saying of a concept that it is absolute is that it does not vary - that it is not relative to the context in which it appears. As we seek to construct models of usage and thereby a conceptual model we can expect to find a dichotomy of relative and absolute terms: some terms, such as father, daughter, length, colour, that vary in so much as they reference different items in different contexts; and terms, such as the earth, the pole star, the London Times that are absolute or constant in what they reference. Whereas relative terms are conceptualised as relationships or as quantitative or adjectival attributes of subject entities, absolute terms cannot be so interpreted unless we posit the existence of a singular entity and then interpret the absolute terms as relational to or as attributive to this singular entity. In this way the matter is finessed for we can say of a relative term such as father that it varies as the person varies - different subject persons having different fathers - and we can say of an absolute term pole star that it varies depending on the singular entity - the Absolute - which is not all.

9 Context as Structure

Entity models enable representation of the contexts within which things exist.

It would be hard not to agree with Gilbert Ryle that context is most important to the meaning of things both said and written:

A given word will, in different sorts of context, express ideas of an indefinite range of differing logical types and, therefore, with different logical powers. And what is true of single words is also true of complex expressions and of grammatical constructions. (1945, 206)

If you are familiar with computer programming you will be aware that computer instructions require context for their execution and of the fact that they perform according to data within the contexts of their execution. Likewise, if we were to note the inferences that we make in reading the first few sentences of a novel, to construct the opening scene and the characters and to place the narrator, then, focusing in this way on these mental steps, it would seem that novelists ask a lot of us the readers to construct a novel's opening context - that they make us work. Give it some thought, for a moment, on the opening lines of a novel you have at hand or consider the opening lines from James Joyce's 'Portrait of the Artist as a Young Man'.

Once upon a time and a very good time it was there was a moocow coming down the road met a nicens little boy named baby tuckoo... His father told him that story: his father looked at him through a glass: he had a hairy face.

When computer programs are written then context is represented explicitly - variables, i.e. individuals, are introduced explicitly - the moocow and the nicens little boy named tuckoo -

and must be used consistently otherwise the software doesn't work. So computer programs are long and heavy on detail and they are in a style which is the antithesis of literary.

Contexts contain individual persons, and times (once upon a time) and places (the road) and they are modelled and represented as data in both computer programs and in novels. In both storytelling and programming, individuals from one context become inaccessible from others - they become 'out of scope' to use a programming term - so by the second paragraph quoted above the individuals of the first paragraph, in and by themselves, are 'out of scope' - to be replaced by the first paragraph itself, which, in its totality, is 'in scope' as a story.

To introduce a real number as an unknown into a mathematical discussion is no different in principle to introducing a moocow or a tiny baby into a discourse. In both cases an individual is introduced into a context and becomes part of the context for the remaining discussion - until that is - that storyline is escaped, becomes embedded, or otherwise goes out of scope - in the mathematical book of exercises, by convention, unless otherwise stated, the x of question 1 is out of scope in question 2.

So contexts, computational, mathematical and discursive, are populated by individuals. 'Moocow', 'baby' and 'Real number' are types of individual. Man, ox, animal, tree, plant - large proportions of our vocabularies comprise words for types of things. Entity modelling is a tool enabling the expression of types (of entity) but also dependencies between individuals, types and contexts.

10 Howlers and Other Violations

Taking a bearing from a map, choosing a meal, calculating a trajectory, developing a new drug, if errors are made then it is preferable to find them early - before the impasse, the disappointment, the failed flight, the failed drug. So it is with computer programming and logical analysis of any kind - if errors are to be found, if systems are to fail, if rethinks are to ensue then better rethink or reprogram early rather than backtrack later. In software development, there is a class of errors called 'type errors' such as, in the context of logical analysis, philosopher Gilbert Ryle called 'category mistakes' or, more strikingly, 'category howlers'.

There is a class of programming language[the statically typed languages] in which type errors are not tolerated - they need to be fixed at the earliest possible stage - unless they are fixed the program cannot be deployed or executed. Use of such a programming language helps the programmer avoid such 'howlers' but H G Wells took Logicians to task for not helping 'thinking men in avoiding such errors:

Finally : the Logician, intent upon perfecting the certitudes of his methods rather than upon expressing the confusing subtleties of truth, has done little to help thinking men in the perpetual difficulty that arises from the fact that the universe can be seen in many different fashions and expressed by many different systems of terms, each expression within its limits true and yet incommensurable with expression upon a differing system. There is a sort of stratification in human ideas. I have it very much in mind that various terms in our reasoning lie, as it were, in different planes, and that we accomplish a large amount of error and confusion by reasoning terms together that do not lie or nearly lie in the same plane.

Other errors arise through confusions of context and some such, in programming parlance, are called scope violations. These arise when relationships are asserted between individuals whose contexts are *a priori* inconsistent with them. Scope violations are attempts to relate across contexts illegally.

Of all errors, in logical reckoning or in programming, category howlers and scope violations we need to fix; without fixing them we don't have proper basis to continue. To avoid these howlers we need to be clear about types and scopes - it is establishing clarity on types of entity and types and scoping of relationships that is the purpose of an entity model.