

Martin Frické

Classification, Facets, and Metaproperties

Abstract

The paper argues that second order properties or metaproperties are essential for classification and navigation of information, for example for faceted classification and the navigation it generates. The paper observes that metaproperties, are not accommodated well within such standard schemes as Z3.1, description logics (DLs), and the formal ontologies OWL, BFO, and DOLCE.

Introduction

Information needs to be classified for the purposes of searching, browsing, and berrypicking (Bates, 18). Considerable progress can be made with this by using just the notions of particular (or item or instance or token), property (or universal or concept or type), and the relations ‘is a subtype of’, ‘is an instance of’, and ‘is a part of’. This paper centers on the relation ‘is an instance of’. Existing approaches make extensive use of this relation. But typically they do so only for the case where it is particulars or items or tokens that are instances of the types. This amounts to applying a property (the type) to an item. As such, this is a case of first order instantiation or the application of first order properties. But this is not the only kind of instantiation there is. First order types themselves can have properties applied to them; when they do, these are cases of second order instantiation or the application of second order properties for example, the property ‘being a tiger’ itself has the property of ‘being a species’, so tiger is an instance of the second order type species (notice the syntax here, ‘tiger is an instance of species’ not ‘a tiger is an instance of species’). (There can be yet higher order properties, but they are not of immediate interest in this paper.)

Sometimes the word ‘metaproperties’ is used to signify second-order properties. On those relatively rare occasions when information scientists and web-site designers want to discuss properties of, or about, properties, they will sometimes call these ‘metaproperties’. (Typically philosophers and logicians would not talk in this way, although they would have no trouble understanding what was being said when others did so.)

Classification requires higher order properties or second order concepts or metaproperties (Sowa 2000). Why? Consider three examples. Genus and Species, the foundations of Aristotelian and Linnaean classification, are higher order properties, period. Second, within librarianship, such entry points as ‘Search by Literary Form’ use second order properties. Finally there is the rising star of classification, namely faceted classification (both seen in the traditional library classifications and central also to the pure faceted classifications now dominating the Web). With faceted classification, it is usual to categorize concepts or types to obtain the facets and categorization of concepts uses second order properties. When Shiyali Ranganathan had the insight, in his Theory of Fundamental Categories, that there were kinds of concepts, that insight was the insight that we should address second order instantiation (Ranganathan 1937, 1951, 1960) . Then, within any facet, there is the notion of an enumeration of foci, required to construct indexes (enumerations which, essentially, are either linear, or hierarchical, ordered tree traversals). And the notion of a sequence or enumeration of foci or concepts is second order. And when Users employ faceted schemes in their searches, the navigation and search process relies on and should rely on, second order properties: entry via facet is entry via a second order property.

But, for many of the modern schemas for digital resources and data curation, second order properties are not first class citizens. Four such schemas are briefly discussed in this paper: the Z3.1 thesaurus construction standard (NISO 2005; Zeng 2005), Description Logics (DLs) (Baader et al 2001; Lutz 2010), Web Ontology Language (OWL) (Horridge 2009; W3C 2004) and the formal ontologies exemplified by Basic Formal Ontology (BFO) (Ifomis 2009) and Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) (Masolo et al 2001).

Some Semi Formal Notation

Somewhat surprisingly, ordinary first order logic is adequate to discuss the second order properties that are the topic of this paper. This can be done first via the ‘intensional abstraction’ of George Bealer (Bealer 1982) which allows properties and relations to be terms and thus have further properties applied to them, and second via order sorted first order logic, restricted quantifiers, and general models (Blasius et al 1998; Kriesel & Krivine 1971; Manzano 1993; Manzano 1996; Oberschelp 1998; Wang 1952) and these can provide clarity to the semantics.

The upshot is, we can take a predicate like Tiger(x) or T(x) and use that to signify ‘x is a Tiger’. Then use the familiar abstraction or comprehension or ‘set builder’ notation

$$\{x: \Phi(x)\}$$

to construct an intensional abstraction or type. The $\Phi(x)$ itself denotes an ‘open sentence’ (formulas of the predicate calculus, usually with a free occurrence of x .) So, as examples,

$$\begin{aligned} &\{x: \text{Friend}(x)\} \\ &\{x: \text{Friend}(x) \& \text{Male}(x)\} \end{aligned}$$

are abstractions.

What exactly is an abstraction, say $\{x: \text{Friend}(x) \& \text{Male}(x)\}$? Syntactically it is a term (ie a name). Semantically, it is a property or notion, or concept, or type, in this case ‘male friend’. Readers with a knowledge of set theory would read $\{x: \text{Friend}(x) \& \text{Male}(x)\}$ as being the set or collection of male friends. But, actually, set theory should not be used here (set theory is extensional and is not suitable for classification work outside of areas which are unchanging and necessary ie outside of mathematics). So $\{x: \text{Friend}(x) \& \text{Male}(x)\}$ is understood as the property or concept or type or intension ‘male friend’ and not the set of the male friends.

The abstraction

$$\{x: \text{Tiger}(x)\} \text{ (or, more briefly, } \{x: T(x)\})$$

can be read ‘the property Tiger’ or ‘the type Tiger’.

These abstractions are intended to be terms or names, so, formally, predicates can be applied to them, and statements of identity can be formed between them and other terms. So

$$\text{Species}(\{x: T(x)\}) \text{ or } S(\{x: T(x)\})$$

applies the ‘species’ predicate to the abstraction. And

$$t = \{x: T(x)\}$$

states an identity between the term ‘ t ’ (ie the type ‘ t ’) and the abstraction. Then

$$S(t) \text{ or } \text{Species}(t)$$

is a second order application of the property or predicate ‘is a Species’ to the type tiger. So there could be formal sentences

$$T(s) \& S(\{x:Tx\}) \text{ or} \\ T(s) \& (t = \{x:Tx\}) \& S(t)$$

to symbolize

'Shere Khan is a tiger and tiger is a species of animal'.

Some Kinds of Classification

The three main 'analytic' relations of classification are 'is a subtype of', 'is an instance of' and 'is a part of'. This paper centers on the first two and their use, and the use of higher order properties, in subtype-supertype hierarchies.

Standard Aristotelian Linnaean Classification Hierarchy

Schematically a fragment of such a classification might look like this

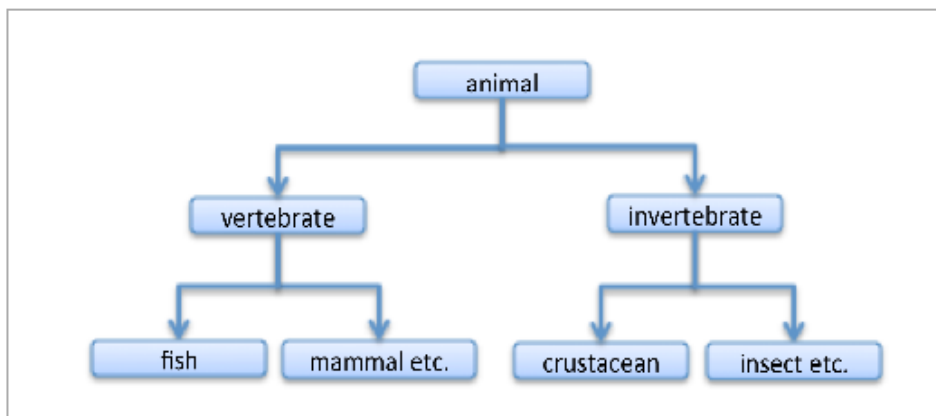


Figure 1

This shows how some types are related to each other. In particular, for example, the fish-type is a subtype of the vertebrate-type, and so on. These classifications are, in a graph-theoretic sense, trees, which is to say they are acyclic and connected.

Additionally there is the notion of levels, which is just the link distance of a type from the (animal) root. So, vertebrate is of level 1, fish of level two, etc. These levels are higher order properties possessed by the types, if $\text{LevelOne}(x)$ is the predicate to express level 1, then, for example $\text{LevelOne}(\text{vertebrate})$ and $\text{LevelOne}(\text{invertebrate})$. And usually these levels have particular names of their own (eg Kingdom, Phylum, Family, Genus, Species).

Aristotelian classification hierarchies are usually taken to require and exemplify five properties: the classification should be exhaustive, exclusive, principled, rich, and narrowing. And Aristotle himself would probably also have required that each of the types be instantiated (Berg 1982). With an Aristotelian hierarchy, it is the leaves that are going to be the classification ‘containers’ or ‘buckets’. The first two requirements, exhaustive and exclusive, sometimes called the JEPD property (jointly exhaustive pairwise disjoint) require that everything is an instance of exactly one leaf (If JEPD holds for the leaves of a tree, it also holds for all the child nodes for any of the internal, non-leaf, parent nodes within the tree).

The principled requirement concerns the ‘sibling separator’, or ‘differentiating condition’, or ‘differentiae’, which is used to separate children types. To be principled, each node should have just a single principle of subdivision which produces its children. A classification which is not principled is suspect in that it might fail as more items are classified. Buchanan gives an example from the London Education Classification (Buchanan, 17).

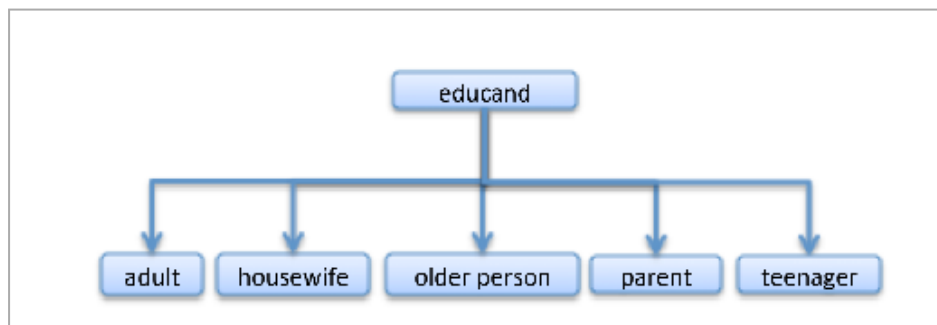


Figure 2

An ‘educand’ is a person being educated; and the types ‘teenager’, ‘adult’, ‘older person’ are being produced on the basis of age, the type ‘parent’ arises on the basis of relationship, and the type ‘housewife’ on the basis of occupation. The division is not principled. Actual educands in London in 1970, say, might, by the luck of it, fall as instances of these types in an exclusive and exhaustive way, but that would be a pure accident because it is quite possible for a housewife to be a parent (or an adult a parent, etc.).

A classification is rich if it does not omit levels. The actual basic classification is done by the leaves. So, in that sense, the two classifications

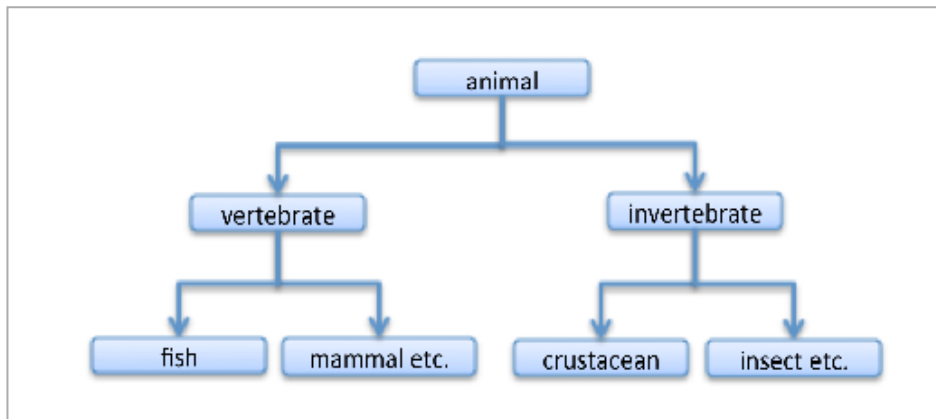


Figure 3

and

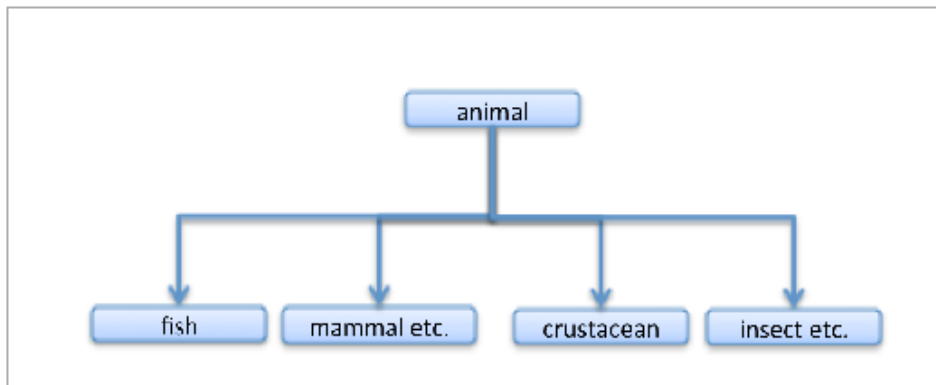


Figure 4

are the same in their capabilities for classification, in as much as they have the same leaves. However, a classification scheme also both contains knowledge or information about how the types relate to each other and has the ability to support inference. For example, the first scheme contains the information that all fish are vertebrates (and, in turn, in conjunction with the fact that Livingstone is a fish, that knowledge supports the inference to Livingstone is a vertebrate). The second scheme, by omitting a layer, omits information that the first scheme contains; it omits some relations between types. Generally, it is not easy to omit layers on other grounds, for example, on considerations relating to principled division. But, even when it is possible, it is not wise to do it, because it leaves out information. Rich schemas contain their full quota of levels, their full quota of information.

The last Aristotelian hierarchy property, narrowing, just means that starting at the root and coming down the appropriate branch refines and sharpens

the classification of an item or items; narrowing almost always follows automatically - each type (other than the leaves) just has to have at least two subtypes. An individual child of a node is never going to have more instances than its parent - that is narrowing

These conditions can be met by conjoining suitable predicates when forming the intensional abstractions; for example, thus

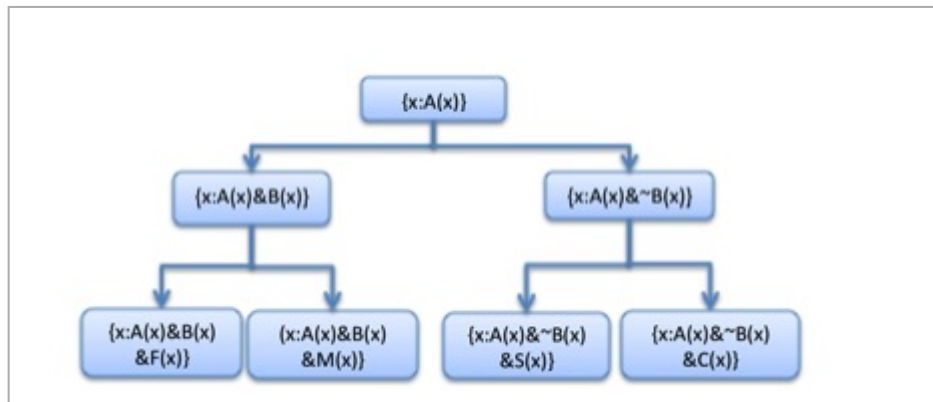


Figure 5

Imagine this example tree being built in stages. That the root type are animals amounts to their possessing the property $A(x)$ (ie x is an animal). For the next division to ‘vertebrate’ and ‘invertebrate’ a suitable predicate is needed that will divide up the animals exclusively and exhaustively between the two types so that every animal will end up being a vertebrate or a non-vertebrate and no animal will end up being both. The property of having a backbone is one that will achieve this. This division is also principled. The use of a single predicate means that there can be only two ‘species’ at this level. Obviously this is insufficient in the general case. However, more than a single predicate can be used; for example, $C(x)$ and $D(x)$ could be used, and that would accommodate 4 species or subtypes, namely $C(x)\&D(x)$, $C(x)\&\sim D(x)$, $\sim C(x)\&D(x)$, $\sim C(x)\&\sim D(x)$. Generalization of this provides for 2, 4, 8, 16, etc species; and other intermediate numbers can be accommodated by combining some of the possible categories together, for example $C(x)\&D(x)$, $C(x)\&\sim D(x)$, and $\sim C(x)$ (ie the combination $(\sim C(x)\&D(x) \vee \sim C(x)\&\sim D(x))$) gives 3 species. The combinations would just have to have the requisite logical properties. And that each of the types be instantiated (Berg 1982) is easy to impose if it is considered desirable (eg with formulas like $\exists x(A(x))$).

A fragment of the Dewey Classification, around 820, is close to being another example:

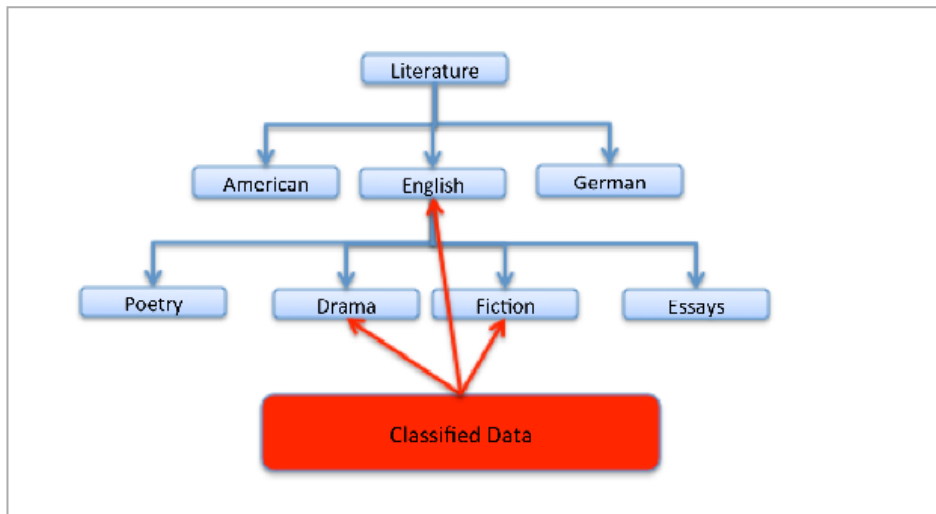


Figure 6

The bottom horizontal level is a level of ‘Literary Form’, and that is a second order property with the types poetry, drama, etc. as instances. Poetry, drama, fiction, etc. are of the higher order type ‘Literary Form’, the classified books themselves are not. (The Dewey Classification is odd in that some internal nodes are used for classification also (as illustrated by the heading English itself). This means that the classification is not ‘exclusive’ and so, strictly speaking, is not Aristotelian.)

Navigation of a class hierarchy can be vertical, but it can also be horizontal via the levels (or Genus and Species etc.). This is useful. A librarian might ask a Patron ‘What Literary Form are you interested in?’ This is to enter a hierarchy on a horizontal level; the technique is important for Information Architecture and website design.

Polyhierarchies

There can be schemes of relationships among types where types can have more than one immediate supertype (so-called ‘multiple inheritance’). Of course, once there is a polyhierarchy, the classification structure is not a tree (because it will have cycles). Also the notion of levels (eg Kingdom, Phylum, Family, Genus, Species etc) is ill-defined or hard to define, because a type might have what amounts to different levels in the different hierarchies.

Here is an example. Say we wanted to classify colored cars, with models and makes and with colors. Then we might let the colored cars inherit their colors. Thus,

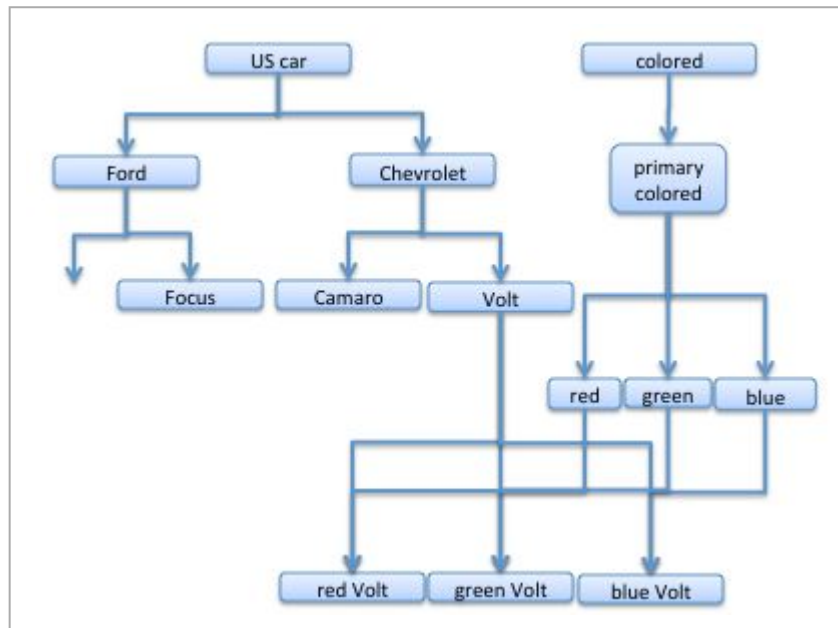


Figure 7. The drawing of red, green, blue Explorers, Focusses, Camaros. etc. has been omitted for clarity

So, a red Volt car is a Volt is a Chevrolet and is a US car: it is also a red item, it is a primary colored item and a colored item. A red Volt has two separate parents that it inherits from. It inherits from red and it inherits from Volt. (And so too for the other colored cars were we to draw them in.)

This particular classification is something of a special case in that the multiple inheritance is coming from the leaves only but, in the general case, any of the nodes might multiply inherit.

This polyhierarchy is not a tree. It has cycles (volt-redVolt-red-primary-green-greenVolt-volt is a cycle).

The logic of a polyhierarchy is merely that of conjoining the properties in the separate hierarchies. So, for example, a red Chevy Volt might be

$$\{x:R(x)\&V(x)\}$$

or, if we include the hierarchy information (that red is a primary color, that a Volt is a Chevrolet and a US Car) we get

$$\{x:R(x)\&P(x)\&C(x)\&V(x)\&D(x)\&U(x)\}$$

The types themselves in a polyhierarchy can have properties or be of kinds. For example, all the types in car hierarchy have the higher-order type ‘being

a car classification type’, and all the types in the color hierarchy have the higher-order type ‘being a color classification type’ (and some types are in both these hierarchies and have both higher order types).

A true polyhierarchy really needs its individual hierarchies to be independent or orthogonal. Types in one should be independent of types in the other. That there is in one hierarchy the type ‘Volt’ should have no implications whatsoever on the color hierarchy. In conception, at least, these hierarchies are going to synthesize. That is to say, there will be Ford Explorers, Ford Focuses etc. from one hierarchy, and red, green, blue from the other; and then, implicitly (and explicitly, if there is literary warrant) all combinations are acceptable eg red Explorers, green Explorers etc. If the hierarchies are not orthogonal, this will be violated. For example, if an envisaged car hierarchy had, of itself, the category ‘red Explorer’, that category could not be combined with the color green—the notion of a green red Explorer is a nonsense. This is because the hierarchies are not orthogonal. The orthogonality requirement points to something else. Each of the component hierarchies has differentiae, the differentiation predicates that do the divisions down the levels. The types of one hierarchy must not be used as the differentiae in another (because that removes full synthesized combinations below). Fugmann alludes to exactly this when he writes

a characteristic of a subdivision should be avoided which is in itself of the categorial kind (ie don't subdivide one category using another). (Fugmann 1993, p. 137)

Basic and Advanced Synthetic Classification

An easier and better way of doing polyhierarchies is not to try to combine two (or more) hierarchies into one large graph. Instead, it is to use two or more separate classification schemes simultaneously in parallel. A common example of this is where there is one classification scheme of, say, items or subject matters, and then one or more ‘Auxiliary Tables’ (ie additional classification schemes) of Periods or Places; and then the classification types are formed by synthesizing together a subject matter and a Period (and perhaps even a Place). So, the subject, or literary form, Poetry might come from one classification scheme, the period 1th Century from another, and French from a third, to make the synthesized type ‘1th Century French Poetry’. This technique can be seen, to a greater or lesser degree, by design or by happenstance, in many traditional librarian schemes, such as Universal Decimal Classification (UDC), Bliss, Dewey Decimal Classification (DDC), Library of Congress Subject Headings (LCSH) and even, mildly, in the Library of Congress Classification (LCC).

Typically, with schemes that permit (basic) synthesis, the actual synthesis consists of the addition (meaning intersection) of the component types. This is easy to reproduce or portray with symbolic logic. For instance, the components

$$\begin{aligned} &\{x:NineteenthC(x)\} \\ &\{x:French(x)\} \\ &\{x:Drama(x)\} \end{aligned}$$

are synthesized by being logically ‘anded’ together to yield

$$\{x:NineteenthC(x)\&French(x)\&Drama(x)\}$$

It is possible to do synthesis from a starting point of just one classification, and synthesize it with itself. For example, the Paul Otlet and Henri LaFontaine’s Universal Decimal Classification (UDC), with its operator ‘+’, permits exactly this (UDC Consortium 2010).

Producing (basic) synthetic classifications using symbolic logic is straightforward. The starting point is a list of acceptable predicates, or conditions, for example

$$\begin{aligned} &EighteenthC(x), \\ &NineteenthC(x), \\ &Fiction(x), \\ &Drama(x), \\ &French(x), \\ &German(x) \end{aligned}$$

(In a realistic case this list might run to thousands of conditions, but that does not affect the principles being explained.) And then each of the types, or intensional abstractions, the grammar, is formed by logical synthesis of these predicates to produce any well formed open sentence and its associated intensional abstraction, as examples

$$\begin{aligned} &\{x:(EighteenthC(x)\&Fiction(x))\}. \\ &\{x:(Drama(x)\&German(x))\} \end{aligned}$$

Notice also that the conditions have kinds—there are metaproperties (for example, Periods and Places) that organize them. And there still will be a need to capture and represent in logic any hierarchies, the semantics, in the synthetic classification; for example, maybe it should have Europe as a Place that encompasses France, etc..

Basic synthesis consists of the addition (‘anding’) of the component types; for

example

$$\{x:NineteenthC(x)\&French(x)\&Drama(x)\}$$

Symbolic logic allows you to do much more. The synthesis in logic does not have to consist solely of ‘anding’ components. Full Boolean operations are permitted to form the open sentence of the intensional abstract, and, indeed, further predicate logic operations that go beyond truth functional connectives. For example,

$$\{x:NineteenthC(x)\&\sim French(x)\&Drama(x)\&\exists y\exists z(\sim(y=z)\&Wrote(y,x)\&Wrote(z,x))\}$$

would be a synthesized classification entry for ‘1th Century non-French Drama, written by at least two authors’.

Synthesis allows us to go beyond polyhierarchies. To explain how and why, we will detour via some theories of Ranganathan. In a polyhierarchy, the classifying classes, usually the leaves, are non-elemental or non-atomic. The concepts or types involved, for example, ‘red Explorer’ involve two or more conditions (in this case, being red and being an Explorer), and it is these conditions that inherit from the different hierarchies. Notice here that the red Explorer is both a red colored item and an Explorer; it is an instance that inherits up each of the separate hierarchies to their roots. As a class it is a superimposed class (Ranganathan 1960) (and that shows in the logic, namely $\{x:Red(x)\&Explorer(x)\}$). Instance of the class are Explorers which are also red (or red objects which are also Explorers). We also consider indexed classes like 1th Century Poetry to be superimposed classes. Such a class starts with one condition (being Poetry) and then forms a subtype of that by adding a second condition to it (being written in the 1th Century). So 1th Century Poetry is Poetry which has also been written in the 1th Century.

But there are non-elemental classes which are not superimposed classes. Consider ‘Explorer sales (ie the sales of Explorers)’, ‘Explorer prices (ie prices of Explorers)’, ‘Off road trips using an Explorer’, and so on. An Explorer off road trip is not a superimposition of an Explorer and an off road trip (it is not an Explorer which is also an off road trip). These are compound classes related to Explorers-- information about them, or their instances, may well belong in a single magazine, web page, or television advertisement about Explorers. But none of the instances of these compound classes are Explorers, Fords or US cars; to a degree the instances are sales, prices, and off road trips, but that is not the important part, the important part is that they relate to

Explorers. These compounds do not belong in polyhierarchies, because they are not superimpositions. However, they can be synthesized, for example, either of

$$\{x:\exists y(\text{Explorer}(y)\&x=\text{offRoadTrip}(y))\} \text{ or } \\ \{x:x=\text{offRoadTrip}(\text{explorer})\}$$

will produce the type ‘Off road trips using an Explorer’.

And supplementary hierarchical or semantic information can also be included. Suppose also there is the standard car hierarchy relating Explorers to Fords to US Cars, those two pieces of semantic information would allow a logical inference, or reasoning, engine to relate ‘Off road trips using an Explorer’ to ‘Road trips using a US Car’. In turn this means that information objects (IOs) tagged with one of these topics would be related to IOs tagged with the other. This may or may not be important, but standard Aristotelian hierarchies or Polyhierarchies have no general way of relating topics like ‘Off road trips using an Explorer’ and ‘Road trips using a US Car’

Really what is happening with these non-superimposed compound types is that the types are using functions to construct the classes and not just plain (Boolean) logical connectives. Here is another case. Subjects like ‘Statistics for biologists’ are an interesting example of what Ranganathan called ‘bias’ (Buchanan 1997). The subject matter statistics has certain subtopics or subfields to it; for example, ‘Parametric statistics’ and ‘Non-parametric statistics’. But ‘Statistics for biologists’ is not one of those subfields. A statistics student attempting to master statistics does not study parametric statistics, non-parametric statistics, and then, also, statistics-for-biologists. And nor is ‘Statistics for biologists’ a sub-topic of biology. This means that ‘Statistics for biologists’ is not a superimposed type. Statistics for biologists is so-to-speak all of statistics, but all of statistics filtered (‘biased’) in a certain way, perhaps to omit certain technical proofs and subfields, and to add certain relevant biological examples. And ‘filtering’ is just applying a function, so what is needed is

$$\{x: x=\text{forBiologists}(\text{statistics})\}$$

In sum, generalized advanced synthesis, in particular logical synthesis centered on the construction of intensional abstractions, may take us a very long way with classification.

Ersatz and Real Faceted Classification

One form of synthetic classification is the very important faceted

classification. All faceted classifications are (or should be) synthetic, but not all synthetic classifications are faceted (for example, a synthesis of a classification with itself is not faceted). Also there is what one might call real faceted classification (of subjects, concepts, types, or tags), as opposed to ersatz faceted classification (of things, or attributes of things).

In conception, basic or simple or ersatz faceted classification is an easier and better way of doing polyhierarchies. It uses two or more classification schemes simultaneously in combinations. Each of these individual schemes is a 'facet' or 'face'. Real faceted classification goes beyond polyhierarchies (in exactly the same way that advanced synthesis goes beyond basic synthesis). And maybe, as theoreticians have been telling us for sixty years, faceted classification is the only kind we need (Broughton 2006; Classification Research Group 1955).

So, for example, a Car Dealership might use a faceted classification consisting of a Make-Model scheme and separately, simultaneously, in parallel, a Color scheme. So, any particular car for sale will have a Make-Model facet (or face) and a Color facet (or face).

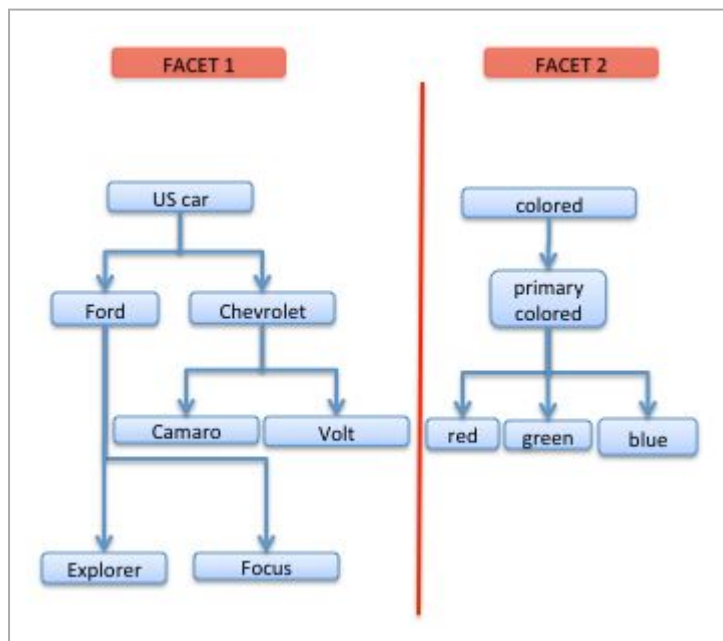
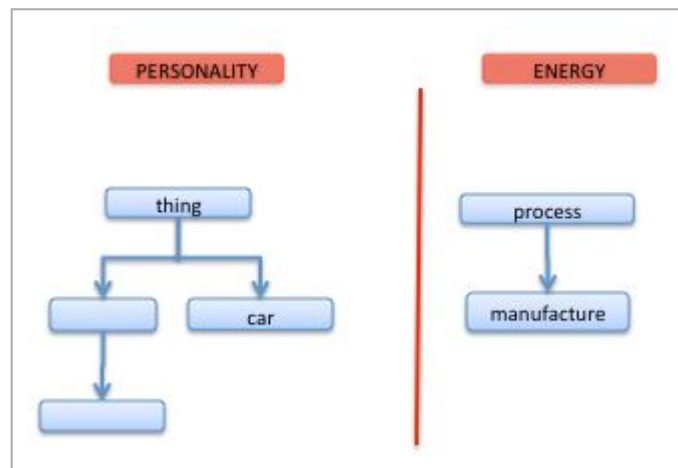


Figure 8

In many ways this is a toy example in that really it is only a distribution of attributes among one kind of thing (see also, for instance, Broughton's socks in (Broughton 2004)). The leaf classes are superimposed classes. A better example is to choose 'true facets', for example



Figure

And this could give us ‘car manufacture’. In the cases where facets can be conceived of as simultaneous and parallel classifications, the second order property of level once again becomes redefined (so there are the analogs of Kingdom, Phylum, Family, Genus, Species for each of the facets separately). There are valuable second order properties for each of the facets individually.

Faceted classification is widespread nowadays. Interestingly enough when the originators devised it, they observed or asserted that there are kinds of concepts. As is well known, Ranganathan (following Otlet and La Fontaine in style) said that concepts are of the kinds PMEST (personality, matter, energy, space, and time). And earlier than this, Julius O. Kaiser anticipated faceting with his synthesis of Concretes and Processes (Kaiser 1911). Categorizing concepts is the approach of many others (Austin 1984; Foskett 1977; Lambe 2007; Rosenfeld & Morville 2006; Willetts 1975). The notion of a kind of concept or category of concept employs higher-order properties or higher-order types. It is a higher-order classification of types, not a first order classification of items or things. The Classification Research Group drew attention to 13 facets (Broughton 2006; Classification Research Group 1955). And modern facet analysis just brings in as many facets as needed for the purposes at hand, for the discipline or area that is being analyzed (Aitchison et al 2000; Buchanan 1997).

Instantiation Classification using Higher Order Types

In most standard classifications, at the vertical level, among the nodes of a tree, everything is a type of the same order. They are homogeneous. And the linking connection is that between subtype and supertype.

It is possible to mix orders of types in a classification. Using higher order types is less common and it is a bit of a test for us and for our understanding. But even children can do it. Ask a child for some colors and you might get

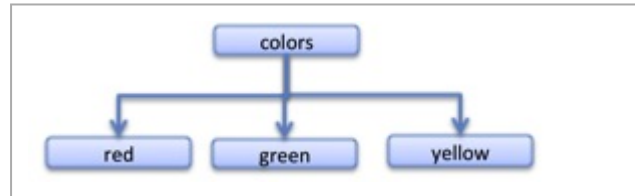


Figure 10

We need to be careful with this diagram. Earlier there was a hierarchy containing ‘colored items’, ‘red colored items’, etc. This is not that hierarchy. In this there is the color red and the type colors. The relationship between the red-type and the colors-type is not that of subtype to supertype (it is not the case that all red things are colors, a toy red fire engine is not a color); rather it is that the red-type itself has the (higher-order) type of being a color. This diagram is not a subtype–supertype classification; it is not a plain hierarchical classification. Rather it is a statement of instances or instantiation, but it is higher order instantiation. What this diagram is doing is taking what we have previously described as a horizontal level (for example, for Species). And then displaying it vertically. To continue. Ask a child for some sizes and you might get

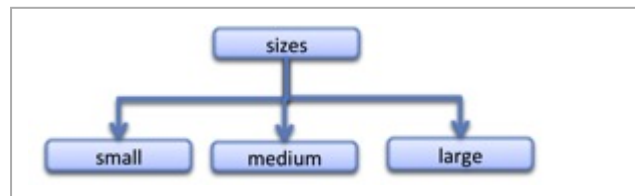


Figure 11

And these two can be combined. Say the child has a box of blocks to play with. Suppose these blocks come in different colors and different sizes. A child may well understand the following scheme to differentiate those blocks.

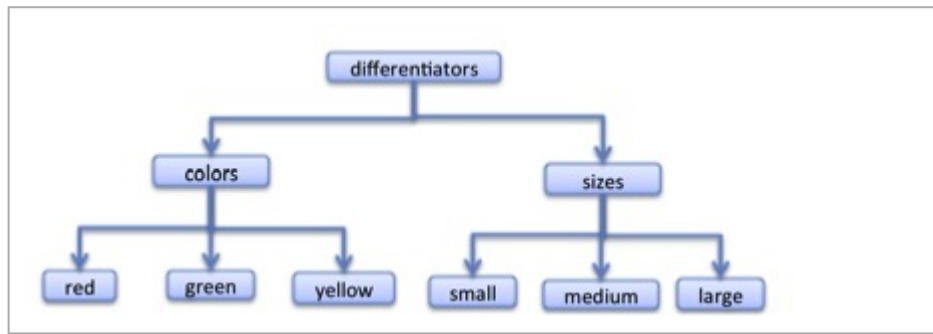


Figure 12

This tree is not using the subtype relation to connect nodes; it is ascending the type hierarchy using instantiation. ‘Colors’ is a second order type that applies to types (and so too is ‘sizes’). And ‘differentiators’ is a third order type that applies to the second order types ‘colors’ and ‘sizes’.

A scheme like this might be used to provide ‘horizontal access’. There could be a standard homogeneous subtype supertype classification (eg blocks->small-blocks medium-blocks large-blocks->red-small-blocks red-medium-blocks red-large-blocks green-small-blocks etc.) And this classification could be used to generate ‘classification’ and ‘call’ numbers to ‘shelve’ the blocks. But, overlaid on this, there could be a retrieval system that supported retrieval by color, retrieval by size, etc. ie horizontal or ‘random’ access. And the child, as User, may have an interest in different kinds of horizontal, or other direct, access (she might want to know who gave her the individual blocks, she might even want to retrieve by donor).

Catalogs versus Propaedia

Aristotle basically cataloged and classified mostly tokens, individual types, kinds, entities, or things; and that style, of forming Catalogs (of things), was the main approach until Francis Bacon. But Bacon, in his 1620 *The Great Instauration* (Bacon, 1620), wanted to classify knowledge; his aim was to catalog knowledge not things. That led to the Encyclopaedias of Ephraim Chambers, Denis Diderot and Jean le Rond d’Alembert and to the encyclopaedic tradition which continues until today (d’Alembert 1751; Yeo 1996, 2001a, 2001b). The encyclopedias themselves contained the actual knowledge, in principle whole libraries, and access to the contents of those encyclopaedias was either via alphabetical index, which is less interesting, or via Trees of Knowledge, themes, or Propaedias, which are more interesting. A Propaedia is a classification of knowledge: see, for example, *Encyclopaedia Britannica* 1974. Mortimer Adler, the designer of the Propaedia for the

Encyclopaedia Britannica, intended for Propaedia to be ‘circles of knowledge’ as opposed to hierarchical trees (Adler 1986). That suggestion is not adhered to in this paper. Here ‘Propaedia’ is used for any graph theoretic thematic structure of knowledge.

What are the differences between Catalogs (of things) and Propaedia (of knowledge)? The differences are not as much as one might think. With both, there are concepts or types, and particulars falling under or instantiating those types. So there is the natural kind type Tiger and the particular or token Shere Khan (were he to be real) instantiating that type; and there is the knowledge type Botany and the particular Linnaeus’s Species Plantarum text instantiating that type. Of course, when we talking about a book or IO instantiating a subject matter or area of knowledge we are not saying that the book is the, or a, subject matter, Species Plantarum is not a Botany (in the way that Shere Khan is a Tiger); rather, there is some elliptical talk going on and when it is expanded out what it says is ‘books have one or more subject matters, so there is a property ‘having a subject matter X’ and in the case of Species Plantarum one of the subject matters that it has is Botany’. Then the concepts or types can be related to other concepts of types is fairly elaborate ways, as Ontologies or as Trees of Knowledge or Propaedia. These classifications or ‘schedules’ mean, first of all, that the species Tiger is related to other kinds, for example, a Tiger is an Animal and the subject matter Botany is related to other subject matters, for example, the subject matter Botany is a sub-subject of the subject matter Reason (in the d’Alembert Tree of Knowledge). Then, second, Shere Khan and Species Plantarum pick up all sorts of other properties from the interrelations in the schedules. So Shere Khan (also) is an animal and Species Plantarum (also) has the subject matter Reason.

All of this is common across both types of cataloging-cataloging kinds and cataloging subjects. What is different, or what can be different, is the number of (narrowest) types, ie the buckets, that a token typically instantiates, and the number of times ‘one and the same’ type can appear in the schedule. With natural kind ontologies, the numbers are one and one. So, when we are trying to locate a species for Shere Khan, he can have only one species-he is a Tiger and that, as far as species are concerned, is all he can be. In turn, the species Tiger appears in the schedule, along with other species, genera, etc., but it can appear only once. In contrast, with a Tree of Knowledge, the numbers can be many and ‘apparently many’. So, when we are trying to locate a subject matter for Species Plantarum it could have more than one-it could be Botany and, separately, Nomenclature. In turn, the subject matter Botany itself appears in the classification schedule, along with other subject matters, but ‘apparently’ it could appear more than once-Botany could be a child subject of Natural Philosophy and, also, a child subject of Biological

Classification.

This latter point needs some clarification and explanation. There is a preliminary distinction that helps. There is the notion of a discipline, which is an area of study, and then there are ‘things’, or ‘phenomena’, or ‘items’ which those areas of study actually do study-- such as, for example, Tigers-- let us call those the objects of study. It is quite possible that the same objects of study occur in different disciplines. Tigers may be studied by the discipline of Biology, and they may also be studied by the discipline of Conservation. So, some knowledge about Tigers will lie with Biology and other knowledge about Tigers, perhaps the same or perhaps different knowledge, will lie with Conservation. The objects-of-study-knowledge will be scattered with a discipline based organization. It is quite possible to organize knowledge by objects of study, which could place all the knowledge about Tigers in one place, but that would be at the expense of scattering the areas of study or disciplines. There are choices (and reasons for and against), but most of the organizations of knowledge, and certainly these historical ones from Bacon and his followers, are discipline based. In the literature this consideration of objects of study is sometimes discussed under the heading of ‘aspect classification’ (Broughton 2004; Hjørland 2002; Mills & Broughton 1977) The substance, and subject matter, water can be used as an illustration. A token liquid spill on a carpet might need to be identified for cleaning purposes, and it might be identified or classified as being water--the spill’s one and only kind is water. As an example of the other, subject matter style, Dewey had this to say about water as a subject matter

a work on water may be classed with many disciplines, such as metaphysics, religion, economics, commerce, physics, chemistry, geology, oceanography, meteorology, and history. No other feature of the DDC is more basic than this: that it scatters subjects by discipline (Dewey 1997, p. xxxi; emphasis added Quoted from Hjørland 2008).

That is, water, the subject, is scattered in different places in the Dewey Decimal Classification (DDC) system. Dewey is not being overly careful here. The fact that water-in-economics appears in a different place in the schedule to water-in-chemistry does not mean that the same subject, water, appears in different places, for water-in-economics is, or might be, a different subject to water-in-chemistry. Think for one moment about an ordinary genealogical family tree; it might have ‘Henrys’ all over the place; but it does not have the same Henry all over the place; it has Henry Jones in one place, Henry Smith in another, and so on. ‘Water-in-economics’ is a compound or complex subject, and one of its components, namely water, appears elsewhere in other compound subjects, for example, in water-in-chemistry. Many subjects are compounds. Foskett mentions this real

example, from the British Technology Index, 'the manufacture of multi-wall kraft paper sacks for the packaging of cement' (Foskett 1977). And, of course, many of the components of this compound, eg manufacture, kraft, paper, sacks, cement, etc., may appear elsewhere as components of other compound subjects. But it is an open question as to whether the exact same atomic or compound subject should appear in several places.

A better example as a candidate for multiple location is a compound subject like economic history which might, in some classification schedules, appear to have its rightful place as a child of economics, and, as exactly the same subject, as a child of history. If carried through, this would make economic history a child of two parents, namely of Economics and of History. And that creates a cycle in the graph, which means that the classifying structure is no longer a tree. And, in fact, the result might need to be treated as a polyhierarchy or by means of facets.

So, a clue for differentiating schedules for knowledge from other schedules lies in whether non-overlapping type instantiation is unique and whether there are cycles or duplicate types at different locations in the schedules. And both of these properties can be discerned by looking at the schedule only--there is no need to look at the items that the schedule classifies. Whether an item can instantiate only one non-overlapping type is a matter of whether the types are exclusive, and that can be determined by logic without looking at the items. And whether a schedule has cycles can be determined just by looking at it.

The Approach of Some Common Systems For Classification

Z3.1 Thesaurus Standard

Z3.1 is about vocabularies and terms ie words, not about properties, types, concepts, and abstract entities (NISO 2005; Zeng 2005). However, types should have terms that name them and vocabularies that describe them. So there is a parallel. Z3.1 is the distilled wisdom of a generation or so of the very best theoreticians and practitioners. As such, it does most everything extremely well. But it is not strong on second order properties or concepts. It handles these under Node Labels 8.5.5 (principle of diffusion among sibling terms). Here is the text

When terms are arranged in hierarchies (such as tree structures) in a taxonomy, thesaurus, or Web navigation layout, node labels may be used to

show the principles of division among a set of sibling terms (terms that share a broader term). Although their function is similar to that of broader terms, node labels are not terms, and must not be used as indexing terms. They are often typographically distinguished from terms, e.g., through the use of italics and/or enclosure in angle brackets.

Example 113: Node labels in hierarchies

- cars
 - by motive power
 - diesel cars
 - electric cars
 - by purpose
 - racing cars
 - sports cars

One can raise an eyebrow here on the apparent denigration of node labels, but that is not the real point. This approach does not quite do what is needed. There are two relevant notions to this topic in a hierarchy: being a sibling (ie having the same parent), and being on the same level (ie being the same distance from the root); siblings are on the same level, but not everything on the same level is a sibling. And the searcher might like to be able to run along a level, or partially run along a level. Consider the car example. Explorer and Focus are siblings, and so too are Camaro and Volt; but Explorer is not a sibling of Camaro, nor is Focus a sibling of Camaro. When the car dealer asks the would-be buyer ‘what model (ie species) are you interested in?’, the dealer would expect a choice from along the level ie one or more of {Explorer, Focus, Camaro, Volt}; the dealer is not thinking siblings. And when a library Patron wants to run along ‘literary form’ that Patron may well want more than the literary form of, say, just English literature. Z3.1 has other partial mechanisms eg for overlapping siblings and for exclusive siblings, etc., but none of these really fit. For levels, the notion of sibling is not enough. To state this in library-speak—you cannot handle the issue of distributed relatives with siblings alone, you need also to consider cousins. A standard text on thesaurus construction (Aitchison, et al. 2000) tells us

It is not necessary to relate (...) sibling terms. The relationship may be seen

by scanning the systemic display (...) or by checking the relevant broader term (...) which will list all the sibling terms at the same level as the sought term. (Aitchison, et al. 2000) p.62

That is exactly right (and there is no argument with it here). However, to be repetitive, when the car dealer asks ‘what model might you be interested in?’ the connective relationship is cousins (and siblings), not just siblings, and the systematic display or ‘the’ relevant broader term will not produce the requisite list. But a suitably programmed computer, aware of second order properties, could easily pop up or produce a list of models of car.

Description Logics (DLs) and OWL

Description Logics (Baader, et al. 2007; Horrocks & Sattler 2001; Lutz 2010) (ie the logic or logics of concept hierarchies) are a fragment of First Order Logic. If the fragment amounts to First Order Logic itself, then the DL in question can certainly do everything advocated here. But, in practice, the fragment is a true fragment and typically there is a restricted but effective approach to concept classification using concepts and some relations on those concepts. One feature that seems to be absent in most or all current approaches is that of allowing one concept to be an instance of another. For example, Protégé (Horridge 2009), the widely used, and very capable, ontology editor, simply does not have the ability to apply one property to another. Protégé can be configured to implement most Description Logics, it also can generate OWL (descriptions in the Web Ontology Language). What is missing from this suite are second order properties. OWL itself also just does not have them (W3C 2004).

Basic Formal Ontology (BFO) and Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE)

The Infomis instigated Basic Formal Ontology (BFO) (Ifomis 2009) also does not have second order properties. It is an Aristotelian style classification in which all the components ‘work’ ie they are all instantiated, and it is aimed at empirical science, prototypically biology, biomedicine, biodiversity etc. That it does not have second order properties is certainly strange. Its intellectual ancestors, Darwin, Linnaeus, and Aristotle regularly talked of Genus and Species, and that is exactly what BFO cannot do.

And the authors of DOLCE write

DOLCE is an ontology of particulars (...) Of course, universals do appear in an ontology of particulars, insofar they are used to organize and characterize them: simply, since they are not in the domain of discourse, they are not

themselves subject to being organized and characterized (e.g., by means of metaproperties)(Italics added) (Masolo, et al. 2001).

But for the purposes of organizing knowledge and information, universals do want to be organized and characterized.

Conclusions

The argument here has been that second order properties are valuable for information architecture and information retrieval. The value to architecture arises as follows. The most powerful and comprehensive approach to the architecture of classification is synthetic faceted classification. But to construct a faceted classification, the component building block concepts need themselves to be given kinds or categorized. They need to be Personalities, or Energies, or Periods, or Places, or Concretes, or Processes, etc.. And that categorization is just the application of second order properties. And information retrieval, or information navigation, follows the architecture. Selecting facets to search by, for example. by Period, is making a choice among second order properties. And, independently, horizontal access in a classification hierarchy, eg search by Literary Form, invokes second order properties (namely the idea of Siblings, Cousins, Species or Levels).

So second order properties are needed, but they are not often used. Here is a conjecture as to why many information systems underrate second order properties: they are addressing Catalogs not Propaedias. When the documentation of the OWL tool Protégé instructs at length on pizza toppings, or when BFO's I-Core wants to discuss weapons in Afghanistan, they are talking about classifying things ie Catalogs. That is well and good, catalogs are important. Librarians classify things also, they classify books, and they produce Catalogs. But they also do something else. When librarians want to talk about classifying subjects; they are not classifying things, what they are doing, in part or in whole, is constructing Trees of Knowledge or Propaedias. And Propaedias are simply more abstract than Catalogs, and constructing and navigating Propaedias benefits from second order properties.

References

- Adler, M. J. (1986) *A Guidebook to Learning : for a Lifelong Pursuit of Wisdom*. Macmillan.
- Aitchison, J., Gilchrist, A., and Bawden, D. (2000) *Thesaurus construction and use: a practical manual* (4th ed). Fitzroy Dearborn.
- Austin, D. (1984) *PRECIS: a manual of concept analysis and subject indexing* (2nd ed). The British Library.
- Baader, F., Horrocks, I., and Sattler, U. (2007) *Description Logics*. In F. van Harmelen, V. Lifschitz & B. Porter (eds) *Handbook of Knowledge Representation*. Elsevier.
- Bacon, F. (1620) *The Great Instauration*. <http://www.constitution.org/bacon/instauration.htm>.
- Bates, M. J. (1998) *The Design of Browsing and Berrypicking Techniques for the Online Search Interface*. *Online Review*, 13, pp. 407-424.
- Bealer, G. (1982) *Quality and Concept*. Oxford University Press.
- Berg, J. (1982) *Aristotle's Theory of Definition*. *ATTI del Convegno Internazionale di Storia della Logica*. CLUEB, pp. 1-30.
- Blasius, K. H., Hedstuck, U., and Rollinger, C.-R. (1998) *Sorts and Types in Artificial Intelligence*. Springer-Verlag.
- Broughton, V. (2004) *Essential Classification*. Neal-Schuman.
- Broughton, V. (2006) *The need for a faceted classification as the basis of all methods of information retrieval*. *Aslib Proceedings: New Information Perspectives*, 58(1/2), pp. 4-72.
- Buchanan, B. (1997) *Theory of Library Classification*. Clive Bingley.
- Classification Research Group (1955) *The need for a faceted classification as the basis of all methods for information retrieval*. *Library Association Record*, 57(7), pp. 262-268.
- d'Alembert, J. L. R. (1751). *Preliminary Discourse* (R. N. Schwab & W. E. Rex, trans.) *The Encyclopedia of Diderot & d'Alembert Collaborative Translation Project*. Trans. of "Discourse Préliminaire," *Encyclopédie ou Dictionnaire raisonné des sciences, des arts et des métiers*, vol. 1. Paris, 1751. Ann Arbor: Scholarly Publishing Office of the University of Michigan Library 200. Web. (4/2/2010). <http://hdl.handle.net/2027/spo.did2222.0001.083>. Trans. of "Discourse Préliminaire". *Encyclopédie ou Dictionnaire raisonné des sciences, des arts et des métiers*, vol. 1. Paris, 1751.
- Encyclopaedia Britannica* (1974) *Propaedia* (15 ed).
- Foskett, A. C. (1977) *Subject approach to information* (3 ed.). Clive Bingley.
- Fugmann, R. (1993) *Subject analysis and indexing. Theoretical foundation and practical advice*. Indeks Verlag.

- Hjørland, B. (2002) The method of constructing classification schemes: a discussion of the state-of-the-art. Proceedings of the the 7th International ISKO Conference.
- Hjørland, B. (2008) Lifeboat for Knowledge Organization. http://www.iva.dk/bh/li-feboat_ko/home.htm.
- Horridge, M. (2009) A Practical Guide To Building OWL Ontologies Using Protégé 4 and CO-ODE Tools (1.2 ed).
- Horrocks, I. and Sattler, U. (2001) Description Logics-Basics, Applications, and More. <http://www.cs.manchester.ac.uk/~horrocks/Slides/IJCAR-tutorial/Print/p1-introduction.pdf>.
- Ifomis. (2009) Basic Formal Ontology (BFO). <http://www.ifomis.org/bfo>.
- Kaiser, J. (1911) Systematic Indexing. Pitman.
- Kriesel, G., and Krivine, J. L. (1971) Elements of Mathematical Logic. North-Holland.
- Lambe, P. (2007) Organising Knowledge: Taxonomies, Knowledge and Organisational Effectiveness. Chandos Publishing.
- Lutz, C. (2010) Description Logics. <http://dl.kr.org/>.
- Manzano, M. (1993) Introduction to many-sorted logic: Many-sorted logic and its applications. Pp. 3-86. John Wiley & Sons.
- Manzano, M. (1996) Extensions of First-Order Logic. Cambridge University Press.
- Masolo, C., Borgo, S., Gangemi, A., Guarino, N., and Oltramari, A. (2001) WonderWeb Deliverable D18.
- Mills, J. and Broughton, V. (1977) Bliss Bibliographic Classification. Second Edition. Introduction and Auxiliary Schedules. Butterworths.
- NISO. (2005) NISO Standards: Z3.1. <http://www.niso.org/>.
- Oberschelp, A. (1998) Order-Sorted Predicate Logic. In K. H. Blasius, U. Hedstuck and C.-R. Rollinger (eds) Sorts and Types in Artificial Intelligence.
- Ranganathan, S. R. (1937) Prolegomena to Library Classification (3rd ed: 1967; 1st ed: 1937). The Madras Library Association.
- Ranganathan, S. R. (1951) Philosophy of Library Classification. Munksgaard.
- Ranganathan, S. R. (1960) Colon classification (6 ed.). Asia Publishing House.
- Rosenfeld, L. and Morville, P. (2006) Information Architecture for the World Wide Web. O'Reilly.
- Sowa, J. F. (2000) Knowledge Representation: Logical, Philosophical, and Computational Foundations. Brooks/Cole.
- UDC Consortium (2010) UDC Consortium. <http://www.udcc.org/>.
- W3C (2004) OWL Web Ontology Language: Overview. <http://www.w3.org/TR/owl-features/>.
- Wang, H. (1952) Logic of many-sorted theories. Journal of Symbolic Logic, 17(2),

105-116.

Willets, M. (1975) An investigation of the nature of the relation between terms in thesauri. *Journal of Documentation*, 31(3), pp. 158-184.

Yeo, R. R. (1996). Ephraim Chambers's *Cyclopaedia* (1728) and the Tradition of Commonplaces. *Journal of the History of Ideas*, 57(1), pp. 157-175.

Yeo, R. R. (2001a) The Best Book in the Universe: Ephraim Chambers' *Cyclopaedia*. *Encyclopaedic Visions: Scientific Dictionaries and Enlightenment Culture*, pp. 120-16. Cambridge University Press.

Yeo, R. R. (2001b) *Encyclopaedic Visions: Scientific Dictionaries and Enlightenment Culture*. Cambridge University Press.

Zeng, M. L. (2005) *Construction of Controlled Vocabularies, A Primer*. <http://www.slis.kent.edu/~mzeng/Z31/index.htm>.

Cite as

Frické, M. (2010) Classification, Facets, and Metaproperties. *Journal of Information Architecture*. Vol. 02. Iss. 02. Pp. 43-65. <http://journalofia.org/volume2/issue2/04-fricke/>.