Cite as: Bergman, M. K. Modular, Expandable Typologies. in *A Knowledge Representation Practionary: Guidelines Based on Charles Sanders Peirce* (ed. Bergman, M. K.) 207–226 (Springer International Publishing, 2018). doi:10.1007/978-3-319-98092-8_10

Official site: https://link.springer.com/book/10.1007/978-3-319-98092-8

Full-text: http://www.mkbergman.com/publications/akrp/chapter-10.pdf

Abstract: In this chapter, we discuss the use of types as our general classification structure, and then typologies as modular ways to further organize those types. Further, Peirce was expansive in his recognition of what kinds of objects could be classified, explicitly including ideas, with application to areas such as social classes, human-made objects, sciences, chemical elements, and living organisms. The idea of nested, hierarchical types organized into broad branches of different entity typologies also offers a flexible design for interoperating with a diversity of worldviews and degrees of specificity.

10

MODULAR, EXPANDABLE TYPOLOGIES

Typology is not a typical term within semantic technology circles, though it is used extensively in such fields as archaeology, urban planning, theology, linguistics, sociology, statistics, psychology, anthropology, and others. Based on etymology, 'typology' is the study of types.[1] However, as used in the fields noted, a 'typology' is the result of the classification of things according to their characteristics. As stated by Merriam Webster, a 'typology' is "a system used for putting things into groups according to how they are similar." Though some have attempted to make academic distinctions between typologies and similar notions such as classifications or taxonomies,[2] we think this idea of grouping by similarity is the best way to think of a typology. In this classification, each of our SuperTypes, as was introduced in the prior chapter, gets its own typology. The idea of a SuperType, in fact, is exactly equivalent to the root node of a typology, wherein we relate multiple entity types with similar essences and characteristics to one another via a natural classification.

A *typology* is a systematic classification of types according to their common or shared characteristics. A typology could be composed of all living things; all animals; all product types; or something as narrow as supervised machine learning algorithms. While *types* are classifications of *instances* according to their shared characteristics, a *typology* is a classification of *types* on a similar basis. Types enable us to "carve Nature at the joints," while typologies give us the organizational framework for coherently subsuming types under a general type. We have complete flexibility to define our general root type for a given typology as narrowly or broadly as we wish, flexibility of immense design importance.

In this chapter, we discuss the use of types as our general classification structure, and then typologies as modular ways to further organize those types. We give particular attention to the benefits that accrue from a typology design. We conclude our chapter with an overview of the typologies used in the KBpedia knowledge structure, and how its design may act as a prototype for other domain applications.

TYPES AS ORGANIZING CONSTRUCTS

Typologies assume the structural form of a subsumption hierarchy, most often in the form of a tree. Each node in that tree is a type, which by definition is itself a Thirdness or general using Peirce's universal categories. Since types are the basic building blocks of a typology, let's take a few minutes to understand them.[*]

The Type-Token Distinction

As used in knowledge representation and philosophy, *types* are the classification of natural kinds. Besides kinds, we often equate types to sets or laws (the Peircean view, using his common terms).[3] Peirce is clear that a type is a general, or collective (1908, CP 8.367). Also, in the base semantic technology language of RDF, a 'type' is what is used to declare an instance of a given class. This use is in keeping with the sense of an *instance* as a member of a type.

Toward the end of his career, Peirce proposed *mark*, *token* and *type* as the trichotomous forms of symbols corresponding, respectively, to his universal categories of Firstness, Secondness, and Thirdness (1908, CP 8.364). The examples he posed to illustrate these ideas related to language symbols. His quote on these distinctions is:

> "A common mode of estimating the amount of matter in a MS [manuscript]. or printed book is to count the number of words. There will ordinarily be about twenty *the's* on a page, and of course they count as twenty words. In another sense of the word 'word,' however, there is but one word ' the' in the English language; and it is impossible that this word should lie visibly on a page or be heard in any voice, for the reason that it is not a Single thing or Single event. It does not exist; it only determines things that do exist. Such a definitely significant Form, I propose to term a *Type*. A Single event which happens once and whose identity is limited to that one happening or a Single object or thing which is in some single place at any one instant of time, such event or thing being significant only as occurring just when and where it does, such as this or that word on a single line of a single page of a single copy of a book, I will venture to call a *Token*. An indefinite significant character such as a tone of voice can neither be called a Type nor a Token. I propose to call such a Sign a *Tone* [later *mark*]; In order that a Type may be used, it has to be embodied in a Token which shall be a sign of the Type, and thereby of the object the Type signifies. I propose to call such a Token of a Type an *Instance* of the Type. Thus, there may be twenty Instances of the Type 'the' on a page." (1906, CP 4.537)

The quote nicely illustrates the distinctions. However, in my view, a too literal reading of this passage by some linguists and semioticians has unfortunately led to a couple of misinterpretations. The first is that tokens are simply occurrences. They are not; they are instances, which may include entities and events (1903, CP 2.245). The second misinterpretation is that the distinction applies to only the symbols on the page, as opposed to the complete sign that those symbols represent. In other words, the *type-token* distinction is not one solely of the written page, a too limited interpretation, but is one of representation by symbols of any type to the particular-general distinction. Tokens and types are not limited to words, but to any instance

* From a vocabulary standpoint, we describe the role of nouns and relations in *Chapter 7*; I also speculate on parts of speech in relation to Peircean terminology in *Chapter 16* (esp. *Table 16-3*).

and general distinction that we may represent symbolically. In Peirce's 10-classification schema for signs, he defined *sinsign*, another term for token:

> "A *Sinsign* (where the syllable *sin* is taken as meaning 'being only once,' as in *single*, *simple*, Latin *semel*, etc.) is an actual existent thing or event which is a sign. It can only be so through its qualities; so that it involves a qualisign, or rather, several qualisigns. But these qualisigns are of a peculiar kind and only form a sign through being actually embodied." (1903, CP 2.245)

Further, he equated the idea of *type* with the *legisign*:

> "A *Legisign* is a law that is a Sign. This law is usually established by men. Every conventional sign is a legisign [but not conversely]. It is not a single object, but a general type which, it has been agreed, shall be significant. Every legisign signifies through an instance of its application, which may be termed a *Replica* of it. Thus, the word 'the' will usually occur from fifteen to twenty-five times on a page. It is in all these occurrences one and the same word, the same legisign. Each single instance of it is a Replica. The Replica is a Sinsign. Thus, every Legisign requires Sinsigns." (1903, CP 2.246)

That Peirce saw the 'collective' type as a general, and not limited solely to language matters, also comes from his referring to his three methods of reasoning — deductive, inductive, abductive — as types. (1913, CP 8.385)

OK, so types and tokens are not limited to the written word and can apply to any general-particular distinction, respectively.[4] Types have an identity and are real, but are not an existent as defined as something with a material instantiation. "Every conventional sign is a legisign," (CP 2.246), or type, it is not an individual thing, and every type requires instances, or sinsigns. (CP 2.246)

Types and Natural Classes

In *Chapter 5* we discussed the importance of natural classes, which is related to a realistic view of the world.[5] Realism means we believe what we perceive in the world is real — it is not just a consequence of what we understand and can be aware of in our minds — and that there are forces and relationships in the world independent of us as selves. Realism is a longstanding tradition in philosophy that extends back to Aristotle and embraces, for example, the natural classification systems of living things as espoused by taxonomists such as Agassiz and Linnaeus. Adhering to realism and a natural classification is the best way to create and organize our types.

Peirce embraced this realistic philosophy but also embedded it in a belief that our understanding of the world is fallible and that we needed to test our perceptions via logic (the scientific method) and shared consensus within the community. As we have noted, his overall approach is known as pragmatism and is firmly grounded in his views of logic and his theory of signs (called semiotics or semeiotics). While absolute truth is real, it acts more as a limit, to which our seeking of additional knowledge and clarity of communication with language continuously approximates. Through the scientific method and questioning, we get closer and closer to the truth and to an ability to communicate it to one another. Still, new knowledge may change those understandings, which in any case will always remain proximate.

This intensional understanding of attributes is key to the classification of entities into categories (that is, 'types'). Further, Peirce was expansive in his recognition of what kinds of objects could be classified, specifically including ideas, with application to areas such as social classes, human-made objects, the sciences, chemical elements and living organisms.[6] Again, here are some of Peirce's own words on the classification of entities:

> "All classification, whether artificial or natural, is the arrangement of objects according to ideas. A natural classification is the arrangement of them according to those ideas from which their existence results." (1902, CP 1.231)

> "The natural classification of science must be based on the study of the history of science; and it is upon this same foundation that the alcove-classification of a library must be based." (1903, CP 1.268)

> "All natural classification is then essentially, we may almost say, an attempt to find out the true genesis of the objects classified. But by genesis must be understood, not the efficient action which produces the whole by producing the parts, but the final action which produces the parts because they are needed to make the whole. Genesis is production from ideas. It may be difficult to understand how this is true in the biological world, though there is proof enough that it is so. But in regard to science it is a proposition easily enough intelligible. A science is defined by its problem; and its problem is clearly formulated on the basis of abstracter science." (1902, CP 1.227)

A natural classification system is one, then, that logically organizes entities with shared attributes into a hierarchy of types, with each type inheriting attributes from its parents, distinguished by what Peirce calls its *final cause*, or purpose. This hierarchy of types is thus naturally termed a *typology*.

An individual that is a member of a natural class has the same kinds of attributes as other members, all of which share this essence of the final cause or purpose. We look to Peirce for the guidance in this area because his method of classification is testable, based on discernable attributes, and grounded in logic. Further, that logic is itself grounded in his theory of signs, which ties these understandings ultimately to natural language. Peirce's own words can better illustrate his perspective, some of which I have discussed elsewhere under his idea of 'natural classes' (see *Chapter 5*):

> "Thought is not necessarily connected with a brain. It appears in the work of bees, of crystals, and throughout the purely physical world; and one can no more deny that it is really there, than that the colors, the shapes, etc., of objects are really there." (1906, CP 4.551)

> "What if we try taking the term 'natural,' or 'real, class' to mean a class of which all the members owe their existence as members of the class to a common final cause? This is somewhat vague; but it is better to allow a term like this to remain vague, until we see our way to rational precision." (1902, CP 1.204)

> "... it may be quite impossible to draw a sharp line of demarcation between two classes, although they are real and natural classes in strictest truth. Namely, this will happen when the form about which the individuals of one class cluster is not so unlike the form about which individuals of another class cluster but that variations from each middling form may precisely agree... When one can lay one's finger upon the purpose to which a class of things owes its origin, then indeed abstract definition may formulate that purpose. But when one cannot do that, but one can trace the genesis of a class and ascertain how several have been derived by different lines of descent from one less specialized form, this is the best route toward an understanding of what the natural classes are." (1902, CP 1.208)

'Natural classes' thus are a testable means to organize the real objects in the world, which include both Secondness and Thirdness. Secondness consists of all extant things, namely, entities and events. We include Thirdness because generals are real. What makes these items real and classifiable into types is because they have: 1) *identity*, which means we may refer to them via symbolic names; 2) *context* related to other objects; 3) characteristic *attributes*, with some expressing the essence of what type of object it is; and 4) *realness*, since the general is not a fiction of our minds but a type recognized by others.

Natural classifications may apply to truly 'natural' things, like organisms and matter, but also to human-made objects and social movements and ideas. The key argument is that shared attributes, including a defining kind of 'essence' (Aristotle) or 'final cause' (Peirce), help define the specific class or type to which an object may belong. For Peirce, what science has to tell us, or what social consensus settles upon, holds sway. If accomplished well, natural classification systems lend themselves to hierarchical structures that may be reasoned over. Further, if we make natural splits between typologies, then it is also possible to establish non-overlapping ('*disjoint*') relationships between typologies that provide powerful restriction and selection capabilities across the knowledge structure. We believe KBpedia already achieves these objectives, though we continue to refine the structure based on our mappings to other external systems and other logical tests.

Very Fine-Grained Entity Types

Entity recognition or extraction is a key task in natural language processing (NLP) and one of the most common uses for knowledge bases. Entities are the unique, individual things in the world. Entities typically account for 90% or so of the items in a knowledge base that we may type, though we may also type events, attributes, relations, ideas, and concepts. Context plays an essential role in entity recognition. We have come to define types of finer and finer bases over time.

The 'official' practice of named entity recognition used within NLP began with the Message Understanding Conferences, especially MUC-6 and MUC-7, in 1995 and 1997. These conferences began competitions for finding '*named entities*' within candidate texts as well as the practice of in-line tagging.[7] Many named entities signal their status via capitalization, such as *Rome* or *John F. Kennedy*. Sometimes named entities are also written in lower case, with examples such as rocks ('gneiss') or common animals or plants ('daisy') or chemicals ('ozone') or minerals ('mica') or drugs ('aspirin') or foods ('sushi'). We give some deference to Kripke's idea of 'rigid designators' for how to identify entities; rigid designators include proper names as well as natural kinds of terms like biological species and substances. Because of these blurrings, the nomenclature of 'named entities' began to fade away, though some practitioners still use the term. Much has changed in the twenty years since the seminal MUC conferences regarding entity recognition and characterization. We are learning to adopt a very fine-grained approach to entity types. What we see evolve with fine-grained entity types has led us, in part, to the logic of our typology design.

The original MUC conferences only recognized three initial entity types: `person`, `organization`, and `location` names. However, it did not take long for various groups and researchers to want more entity types, more distinctions. BBN categories, proposed in 2002, were used for question answering and consisted of 29 types and 64 subtypes.[8] Sekine put forward and refined over many years his Extended Entity Types, which grew to about 200 types.[9] These ideas of extended entity types helped inform a variety of tagging services over the past decade, notably including OpenCalais, Zemanta, AlchemyAPI, and OpenAmplify, among others. Moreover the research community also expanded its efforts into more and more entity types, or what we now term fine-grained entities.[10] Ling and Weld proposed 112 entity types in 2012.[11] Another one, from Gillick *et al.* in 2014 proposed 86 entity types,[12] organized, in part, according to the same `person`, `organization`, and `location` types from the earliest MUC conferences. A report in 2017 pointed to 1941 entity types drawn from both Wikipedia and Wordnet.[13] In KBpedia, at the time of this writing, we provide mappings to a large number of entity types in external knowledge bases, including the DBpedia ontology (738 entity types), schema.org (636 types) and GeoNames (654 types).

This growth in entity types comes from wanting to describe and organize things with more precision. Tagging and extracting structured information from text are obviously a key driver. For a given enterprise, what is of interest — and at what depth — for a particular task varies widely. The desired depth, or degree of fine-graininess, increases for entity types within our domains of inquiry. For example, let's take a general thing such as a *camera*. A photographer may want finer-grained distinctions such as *SLR cameras* or further sub-types like *digital SLR cameras* or even specific models like the *Canon EOS 7D Mark II*, or even the name of the photographer's favorite camera, such as '*Shutter Sue*.' Capitalized names (common for named entity recognition) often signal we are dealing with an actual individual entity, but again, depending on context, a named automobile such as Chevy Malibu may refer to a specific car or the entire class of Malibu cars. If our domain of interest is transportation in general, treating the Chevy Malibu as an instance of a Chevy may be sufficient; but if our domain of inquiry includes Chevrolet automobiles, we probably want details including specific years and models of Malibus. This kind of hierarchical organization provides paths for inferencing, as well as user interface benefits (see *Chapter 11*). We can visualize this hierarchy of types something like what we show in *Table 10-1*.

Recent efforts in fine-grained entity recognition are notable because machine learners have been trained to recognize all of the various types indicated. Which entity types to include, the different conceptions of the world, and how to organize entity types varies broadly across these references.

Thing
- Product
  - Camera
    - Digital Camera
      - SLR Digital Camera
        - Olympus Evolt E520

Table 10-1: Hierarchical Nature of Typologies

Perhaps 40,000 entity types are included in the baseline KBpedia knowledge structure to accommodate such fine-grained entity recognition. Over the past two decades we see logarithmic growth in recognition of entity types:

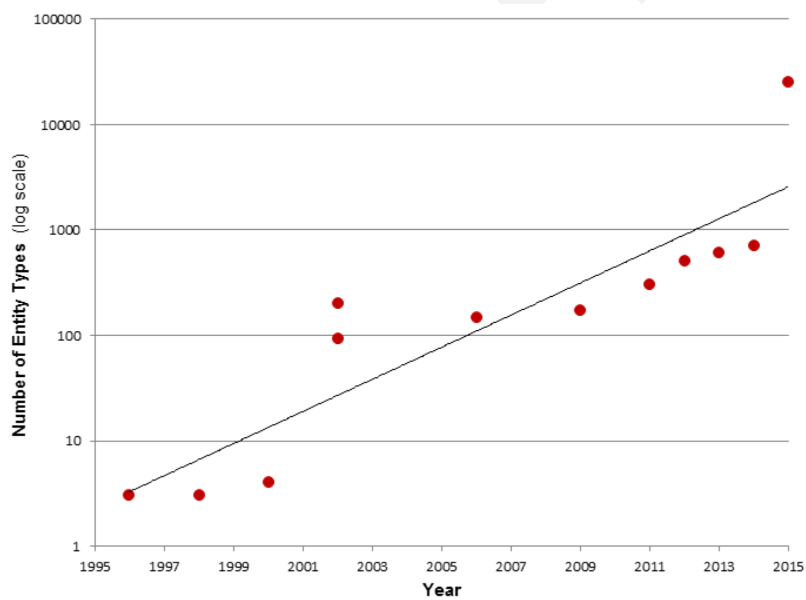

Figure 10-1. Growth in Recognition of Entity Types

Each type has a basis — ranging from attributes and characteristics to hierarchical placement and relationship to other types — that can inform computability and logic tests, potentially including neighbor concepts. We base supervised machine learners on these features. Linking to knowledge bases helps provide the instance data to drive these learners. Ensuring that type placements are accurate and meet these tests means that we may use the now-placed types and their attributes to test the placement and logic of subsequent candidates. The candidates need not be only internal typology types, but may also be used against external sources for classification, tagging or mapping.

It is no longer sufficient to satisfy our domain queries with entity types classified at the level of `person`, `organization`, and `location` names. We want more precision; more detail; and more relevance. The fact that knowledge bases, such as Wikipedia, but also our own data stores and domain-specific knowledge bases as well, can provide entity-level instance information for literally thousands of entity types, means that rich information is now available for driving the finest of fine-grained entity extractors. It is essential to have a grounded understanding of what an entity is, how to organize them into logical types, and an intensional understanding of the attributes and characteristics that allow us to conduct inferencing over these types. These understandings, in turn, point to the features that are useful to machine learners for artificial intelligence. These understandings also can inform a flexible design for accommodating entity types from coarse- to fine-grained, with variable depth depending on the domain of interest.

A FLEXIBLE TYPOLOGY DESIGN

The only sane way to tackle knowledge bases for knowledge representation and management is to provide consistent design patterns that are easier to test, maintain, and update. Open world systems must embrace repeatable and mostly automated workflow processes, plus a commitment to timely updates, to deal with the constant change in knowledge. We also want natural workflows for the knowledge workers who use them, else quality checks and frequent updates will suffer. We thus seek semi-automatic methods for constant knowledge updates.

The *typology* structure is not only a natural organization of natural classes, but it enables flexible interaction points with inferencing and mapping across its design. The typology design is the result of the classification of things according to their shared attributes and essences. The idea is that we divide the world into real, discontinuous and immutable 'kinds.' If using statistical terminology, a typology is a composite measure that involves the classification of observations using attributes treated as variables.

Our typology design has arisen from the intersection of 1) our efforts with SuperTypes, and creating a computable structure that uses powerful disjoint assertions; 2) an appreciation of the importance of entity types as a focus of knowledge base terminology; and 3) our efforts to segregate entities from other constructs of knowledge bases, including attributes, relations, and annotations. Though these insights may have resulted from serendipity and practical use, they have brought a new understanding of how best to organize knowledge bases for artificial intelligence uses. It just so happens that these splits are in complete accordance with Peirce's writings.

The simple bounding and structure of the typology design make each typology understandable merely by inspecting its structure. The typologies can also be read into programs such as Protégé to examine or check complete specifications and relationships. Because each typology attempts to have a coherent, modular, and consistent design, new concepts or structures may be related to any part of its hierarchical design. The organization of entity types also has a different structure than the more graph-like organization of higher-level conceptual schema or knowledge graphs. In the cases of broad knowledge bases, such as Wikipedia, where 70 percent or more of the overall schema is related to entity types, more attention can now be devoted to the remaining 30 percent, to extend insights into type placements based on relationships and attributes. The combination of logical coherence with a flexible, accordion-like structure gives typologies a unique set of design benefits.

Construction of the Hierarchical Typologies

We develop the initial typology by first gathering the relevant types (concepts) and automatically evaluating them for orphans (unconnected concepts) and fragments (connected portions missing intermediary parental concepts). We allow no instances in the typology, only types.* We typically see multiple roots, multiple fragments, and numerous orphans in the first builds. We test and refine until we fix these problems, resulting in a single root to connect all concepts in the typology. We query source knowledge bases for missing concepts and evaluate again in a recursive manner. We then write candidate placements to CSV files and evaluate them with various utilities, including crucially manual inspection and vetting. (Because the system bootstraps what is already known and structured in the system, it is mandatory to build the structure with coherent concepts and relations.)

We should include all types related to a given typology as a sub-module. This design means we may maintain and inspect each typology separately. We may share some types across typologies (due to multiple inheritances), and when many or all typologies are present, the entire knowledge system assumes the form of an interconnected graph. External structures, especially those based on SKOS,[14] are well-suited for direct incorporation as typologies.

Once we complete the overall candidate structure, we then analyze it against prior assignments in the knowledge base. We create CSV files that we may view and evaluate with various utilities for such tasks as SuperType disjoint analysis, coherent inferencing, and logical placement tests. Again, however, to retain the integrity of the structure, we manually vet final assignments. The objective of the build process is a connected typology that passes all coherency, consistency, completeness and logic tests. If we discover subsequent errors, we rerun the build process with updates to the processing scripts. Upon acceptance, we should ensure each new type added to a typology is complete by including a definition, *semset*, guideline annotations, and connections. We write out the completed typology in both RDF and CSV formats.

* However, like the Chevy Malibu case described earlier, items that appear as instances in the putative typology may be expanded to become an eventual class (type), with its own instances, akin to the *punning* discussion in the prior chapter.

Build, testing and maintenance routines, scripts, and documentation must be integral to the design. Knowledge bases are inherently open world, which means that the entities and their relationships and characteristics are continually growing and changing due to new knowledge underlying the domain at hand. Such continuous processing and keeping track of the tests, learnings and workflow steps place a real premium on literate programming. We discuss the build process in *Chapter 14*.

Typologies are Modular

Since each typology has a single root, it is readily plugged into or removed from the broader structure. Each typology is rather simple in scope and structure, given its hierarchical nature. We can readily build, test and maintain each typology. Typologies pose relatively small ontological commitments. This isolated design means the scale and scope of the overall system may be easily adjusted, and we may use the existing structure as a source for extensions (see next). Unlike more interconnected knowledge graphs (which can have many network linkages), typologies are organized strictly along these lines of shared attributes, which is both simpler and also provides an orthogonal means for investigating type-class membership.

Our learning path to a typology design began with our early experience with SuperTypes in UMBEL. We started to explore typologies (though we did not call them that at the time) because we observed about 90% of the concepts in UMBEL were *disjoint* from one another. Disjoint assertions are computationally efficient and help better organize a knowledge graph. Besides computational efficiency and its potential for logical operations, we also observed that these SuperTypes could also aid our ability to drive display widgets (such as being able to display geolocational attributes for geolocational types on maps). As we looked over the tens of thousands of concepts in UMBEL, we began to see we could organize them into a tractable number of SuperTypes. The SuperType tagging and possible segregation of STs into individual modules led us to review other separations and tags. Given that the SuperTypes were all geared to entities and entity types — and further represented about 90% of all concepts at hand — we began to look at entities as a category with more care and attention. This analysis took us back to the beginnings of entity recognition and tagging in natural language processing. We saw the progression of understanding from named entities and just a few entity types, to the more recent efforts in so-called fine-grained entity recognition, as we reviewed above.

What was blatantly obvious, but which other researchers and we had previously overlooked, was that most knowledge graphs (or upper ontologies) were themselves made up of mostly entity types. In retrospect, this should not have been surprising. Most knowledge graphs deal with real things in the world, which are most often entities. Entities are the observable, often nameable, things in the world around us. How we organize and refer to those entities — that is, the entity types — constitutes the bulk of the vocabulary for a knowledge graph. The trick to the transition is moving from the idea of discrete numbers of entity types to a system and design that supports continuous interoperability through a generalized, modular typology structure.

The '*type*-orientation' of a typology is also attractive because it offers a construct that we can apply to all other (non-entity) parts of the knowledge base. We can also type actions, attributes, roles, events, and relations. A mindset around natural kinds and types helps define the *speculative grammar* we need to do *knowledge-based artificial intelligence*. Because the essential attributes or characteristics across typologies in an entire domain can differ broadly — such as entities *v* attributes, living *v* inanimate things, natural things *v* human-made things, ideas *v* physical objects — it is possible to make disjointedness assertions between entire groupings of natural classes. Disjoint assertions, combined with logical organization and inference, further provides a typology design that lends itself to reasoning and tractability. The internal process to create these typologies also has the beneficial effect of testing placements in the knowledge graph and identifying gaps in the structure, as informed by fragments and orphans. This computability of the structure is its foundational benefit since it determines the accuracy of the typology itself and drives all other uses and services.

Typologies are Expandable

A typology design for organizing entities can thus be visualized as a kind of accordion or squeezebox, expandable when detail requires, or collapsible to more coarse-grained when relating to broader views. Each class (type) within the typology can become a tie-in point for external information, providing a collapsible or expandable scaffolding (the 'accordion' design). Via inferencing, multiple external sources may be related to the same typology, even though at different levels of specificity. Further, we may accommodate very detailed class structures in this design for domain-specific purposes. Moreover, because of the single tie-in point for each typology at its root, it is also possible to swap out entire typology structures at once, should design needs require this flexibility.

The idea of nested, hierarchical types organized into broad branches of different entity typologies also provides a very flexible design for interoperating with a diversity of worldviews and degrees of specificity. A typology design, logically organized and placed into a consistent grounding of attributes, can readily interoperate with these different worldviews. The photographer, as discussed above, is interested in different camera types and even how specific cameras can relate to a detailed entity typing structure. Another party more interested in products across the board may have a view to greater breadth, but lesser depth, about cameras and related equipment. A typology design, logically organized and placed into a consistent grounding of attributes, can readily interoperate with these different worldviews. Typologies for attributes and relations, as we have implemented in KBpedia, also extend this basis to include full data interoperability of `attribute:value` pairs.

KBPEDIA'S TYPOLOGIES

So, to understand this typology design in action, it is worth inspecting the KBpedia knowledge structure. I provide the general vocabulary for KBpedia in *Chapter 8*. *Appendix B* is a broad overview of KBpedia. Further, the KBpedia Web site offers access to KBpedia's overall upper structure (KKO),[*] plus the approximately 70 typologies with formal type listings. You may inspect the KKO files and the typologies in an ontology editor to glean additional details.[†] As noted, nearly 90% of the classification structure of KBpedia resides in the *Generals* branch of KKO, which is also the location for all KBpedia types and typologies.

Full Listing of Typologies

Unlike the KKO upper structure, we do not necessarily organize each typology according to Peirce's triadic logic. That is because we are dealing with objects of a more-or-less uniform character (such as *animals* or *products* or *atomic elements*). About 85 such typologies exist in the KBpedia structure, 70 of which with formal typologies ('◊'), and about 30 of which are deemed 'core' ('‣'), meaning they capture the bulk of the classificatory system.

The best perspective to see the full listing of the typologies in KBpedia is to inspect the *Generals* branch of the KKO knowledge graph, which also includes *Predications* types and contains about 85 SuperTypes. *Table 10-2* below provides this *Generals* branch organization, as first organized around the Peircean universal categories of Firstness (1), Secondness (2), and Thirdness (3). Also, recall that the *Generals* branch is itself the Thirdness (3) branch of the broader KBpedia Knowledge Ontology (KKO)[‡]:

- 3-Generals (SuperTypes)
  - 1-Constituents ◊
    - 1-Natural Phenomena ‣
    - 2-Time Types ◊
      - Times ‣
      - Event Types ‣
    - 3-Space Types ◊
      - 1-Shapes ‣
      - 2-Places ‣
        - Area Region ‣
        - Location Place ‣
      - 3-Forms ‣
  - 2-Predications ◊
    - 1-Attribute Types ◊
      - 1-Intrinsic Attributes ◊
        - 1-Qualities
        - 2-Components

* See http://kbpedia.org

† For example, using the open source Protégé ontology development environment (https://protege.stanford.edu/).

‡ The remaining portions of the upper KKO are shown in *Chapter 8*.

            - 3-Forms (configurations)
        - 2-Adjunctual Attributes ◊
            - 1-Quantities
            - 2-Eventuals
            - 3-Extrinsics
        - 3-Contextual Attributes ◊
            - 1-Situants
            - 2-Ratings
            - 3-Classifications
    - 2-External Relations Types ◊
        - 1-Direct Relations ◊
            - 1-Equivalences
            - 2-Parts
            - 3-Descendents
        - 2-Copulative Relations ◊
            - 1-Identities
            - 2-Action Types ◊
            - 3-Conjoins
        - 3-Mediative Relations ◊
            - 1-Comparisons
            - 2-Situation Types ▸
            - 3-Cognitives
    - 3-Representation Types ◊
        - 1-Denotatives ◊
        - 2-Indexes ◊
        - 3-Associatives ◊
- 3-Manifestations ◊
    - 1-Natural Matter ◊
        - 1-Atoms Elements ▸
        - 2-Natural Substances ▸
        - 3-Chemistry ▸
    - 2-Organic Matter ◊
        - 1-Organic Chemistry ▸
            - Biological Processes ▸
        - 2-Living Things ◊
            - 1-Prokaryotes ▸
            - 2-Eukaryotes ◊
                - 1-Protists Fungus ▸
                - 2-Plants ▸
                - 3-Animals ▸
            - 3-Diseases ▸
        - 3-Agents ◊
            - 1-Persons ▸
            - 2-Organizations ▸
            - 3-Geopolitical ▸
    - 3-Symbolic ◊
        - 1-Information ◊

- 1-AV Info ◊
  - Audio Info ‣
  - Visual Info ‣
- 2-Written Info ‣
- 3-Structured Info ‣

2-Artifacts ◊
- FoodDrink ‣
- Drugs ‣
- Products ‣
- Facilities ‣

3-Systems ◊
- 1-Conceptual Systems ◊
  - 1-Concepts ◊
  - 2-Topics Categories ◊
  - 3-Learning Processes ◊
- 2-Social Systems ◊
  - Society ‣
  - Economic Systems ‣
- 3-Methodeutic ◊
  - 1-Inquiry Methods ◊
  - 2-Knowledge Domains ◊
  - 3-Emergent Knowledge ◊

Table 10-2: Full, Upper Hierarchy of KBpedia Generals

In *Table 10-2* the mark '◊' indicates a formal typology for the entry in KBpedia, which means a corresponding file for inspecting it exists. The mark '‣' indicates the formal typology is also one of the 'core' KBpedia typologies, meaning it contains a more substantial number of types with possible disjointedness assertions to other typologies.[15] If time is limited, those typologies are the most fruitful to inspect. The largest files, of course, are the ones with the largest number of types.

'Core' Typologies

So, let's take a bit deeper look at these 30 'core' ('‣') typologies. Here are those 30, with a definition of the type coverage for each:

Constituents	**Natural Phenomena**	This typology includes natural phenomena and natural processes such as weather, weathering, erosion, fires, lightning, earthquakes, tectonics, etc. We explicitly include clouds and weather processes. Also, it covers climate cycles and general natural events (such as hurricanes) that are not specifically named. Biochemical processes and pathways are expressly excluded, occurring under a different typology.

	Area or Region	The AreaRegion typology includes all nameable or definable areas or regions that we may find within 'space.' Though the distinction is not sharp, this typology is meant to be distinct from specific points of interest (POIs) that may be mapped (often displayed as a thumbtack). We may show areas or regions on a map as a polygon (area) or path (polyline).
	Location or Place	The LocationPlace typology is for bounded and defined points in 'space,' which can be positioned via some form of coordinate system and we often show as points of interest (POIs) on a map. This typology is distinguished by areas or locations, which are often best displayed as polygons or polylines on a map.
	Shapes	The Shapes typology captures all 1D, 2D and 3D shapes, regular or irregular. Most shapes are geometrically describable things. Shapes have only a minor disjointedness role, with more than half of KKO reference concepts having some aspect of a Shapes specification.
	Forms	This typology category includes all aspects of the shapes that objects take in space; Forms is thus closely related to Shapes. The Forms typology is also the collection of natural cartographic features that occur on the surface of the Earth or other planetary bodies, as well as the form shapes that naturally occurring matter may assume. Positive examples include Mountain, Ocean, and Mesa. We exclude artificial features such as canals. Most instances of these natural features have a fixed location in space.
Time-related	**Events**	These are nameable occasions, games, sports events, conferences, natural phenomena, natural disasters, wars, incidents, anniversaries, holidays, or notable moments or periods of time. Events have a finite duration, with a beginning and end. Individual events (such as wars, disasters, newsworthy occasions) may also have names.
	Times	This typology is for specific time or date or period (such as eras, or days, weeks, months type intervals) references in various formats.
Natural Matter	**Atoms and Elements**	The Atoms and Elements typology contains all known chemical elements and the constituents of atoms.
	Natural Substances	The Natural Substances typology are minerals, compounds, chemicals, or physical objects that are not living matter, not the outcome of purposeful human effort, but are found naturally occurring. We also place other natural objects (such as rock, fossil, etc.) in this typology. Chemicals can be Natural Substances, but only if they are naturally occurring, such as limestone or salt.
	Chemistry	This typology covers chemical bonds, chemical composition groups, and the like. It excludes natural substances or living thing (organic) substances. Organic Chemistry and Biological Processes are, by definition, separate typologies. The Chemistry typology thus includes inorganic chemistry, physical chemistry, analytical chemistry, materials chemistry, nuclear chemistry, and theoretical chemistry.
Organic Matter	**Organic Chemistry**	The Organic Chemistry typology is for all chemistry involving carbon, including the biochemistry of living organisms and the materials chemistry (and polymers) of organic compounds such as fossil fuels.
	Biochemical Processes	The Biochemical Processes typology is for all sequences of reactions and chemical pathways associated with living things.
Living Things	**Prokaryotes**	The Prokaryotes include all prokaryotic organisms, including the Monera, Archaebacteria, Bacteria, and Blue-green algae. Also included in this typology are viruses and prions.
	Protists and Fungus	This typology is for the remaining cluster of eukaryotic organisms, explicitly including the fungus and the protista (protozoans and slime molds).
	Plants	This typology includes all plant types and flora, including flowering plants, algae, non-flowering plants, gymnosperms, cycads, and plant parts and body types. Note that we also include all Plant parts.

	Animals	This large typology includes all animal types, including specific animal types and vertebrates, invertebrates, insects, crustaceans, fish, reptiles, amphibia, birds, mammals, and animal body parts. We also include all Animal parts. Also, we cover the groupings of such animals (such as herds, flocks). We include Humans, as an animal, but exclude individual Persons. We specifically exclude Diseases. Animals have many of the similar overlaps to Plants. However, there are more terms for animal groups, animal parts, animal secretions, among others. Also, Animals can include some human traits (posture, dead animal).
	Diseases	Diseases are atypical or unusual or unhealthy conditions for (mostly human) living things, generally known as conditions, disorders, infections, diseases or syndromes. Diseases only affect living things and sometimes are caused by living things. This typology also includes impairments, disease vectors, wounds and injuries, and poisoning.
Agents	**Persons**	The appropriate typology for all named, individual human beings. This typology also includes the assignment of formal, honorific or cultural titles given to specific human individuals. It further contains names given to humans who conduct particular jobs or activities (we know the latter as an avocation). Examples include steelworker, waitress, lawyer, plumber, artisan. We specifically include Ethnic groups. Note, we include Persons as living animals under the Animals typology.
	Organizations	Organization is a broad typology and includes formal collections of humans, sometimes by legal means, charter, agreement or some mode of formal understanding. Examples these organizations include geopolitical entities such as nations, municipalities or countries; or companies, institutes, governments, universities, militaries, political parties, game groups, international organizations, trade associations, etc. All institutions, for example, are organizations. Also included are informal collections of humans. Informal or less defined groupings of humans may result from ethnicity or tribes or nationality or shared interests (such as social networks or mailing lists) or expertise ('communities of practice'). This dimension also includes the notion of identifiable human groups with set members at any given point in time. Examples include music groups, cast members of a play, directors on a corporate Board, TV show members, gangs, teams, mobs, juries, generations, minorities, etc.
	Geopolitical	Named places that have some informal or formal political (authorized) component. Notable subcollections include Country, IndependentCountry, State_Geopolitical, City, and Province.
Artifacts	**Products**	The Products typology includes any instance offered for sale or barter or performed as a commercial service. A Product is often a physical object made by humans that is not a conceptual work or a facility (which have their own typologies), such as vehicles, cars, trains, aircraft, spaceships, ships, foods, beverages, clothes, drugs, weapons.
	Food or Drink	This typology is any edible substance grown, made or harvested by humans. The category also includes the concept of cuisines explicitly.
	Drugs	This typology is a drug, medication or addictive substance, or a toxin or poison.
	Facilities	Facilities are physical places or buildings constructed by humans, such as schools, public institutions, markets, museums, amusement parks, worship places, stations, airports, ports, car stops, lines, railroads, roads, waterways, tunnels, bridges, parks, sports facilities, monuments. All can be geospatially located. Facilities also include animal pens and enclosures and general human 'activity' areas (golf course, archeology sites, etc.). Importantly Facilities include infrastructure systems such as roadways and physical networks. Facilities also include the components that go into making them (such as foundations, doors, windows, roofs, etc.). Facilities can also include natural structures that have been converted or used for human activities, such as occupied caves or agricultural facilities. Finally, facilities also include workplaces. Workplaces are areas of human activities, ranging from single person workstations to large aggregations of people (but which are not formal political entities).
Information	**Audio Info**	This typology is for any audio-only human work. Examples include live music performances, record albums, or radio shows or individual radio broadcasts

	Visual Info	The Visual Info typology is for any still image or picture or streaming video human work, with or without audio. Examples include graphics, pictures, movies, TV shows, individual shows from a TV show, etc.
	Written Info	This typology includes any general material written by humans including books, blogs, articles, manuscripts, but any written information conveyed via text.
	Structured Info	This information typology is for all kinds of structured information and datasets, including computer programs, databases, files, Web pages and structured data that can be presented in tabular form.
Social	**Finance and Economy**	This typology pertains to all things financial and concerning the economy, including chartable company performance, stock index entities, money, local currencies, taxes, incomes, accounts and accounting, mortgages and property.
	Society	This category includes concepts related to political systems, laws, rules or cultural mores governing societal or community behavior, or doctrinal, faith or religious bases or entities (such as gods, angels, totems) governing human spiritual matters. We include culture, Issues, beliefs and various activisms (most -isms).

Table 10-3: 'Core' KBpedia Typologies

Because *Table 10-3* does not show all of the typologies, we collapse some of the hierarchical aggregations a bit. Note that the typologies that are not part of this 'core' listing also have complete descriptions within the online ontology files, as well as, of course, other specifications related to their roles in the knowledge graph.

Tailoring Your Own Typologies

The open source nature of KBpedia is such that you may use as little or as much of this structure as you would like to build your own domain knowledge representations. The basic KKO structure, plus expansions or constrictions of existing KBpedia typologies, provides a consistent scaffolding, with some promise of interoperability with external systems, for your knowledge efforts.

The quickest way to leverage KBpedia is to create and add your domain typologies. As needed, these may be large expansions of new detail and scope. Some areas may only require sporadic extensions or attention to the types already in KBpedia. I noted earlier the importance of addressing orphans and fragments as you build these typologies. You may need to create some new branches, including perhaps major ones, to capture the new domain scope. Once you are done revising KKO and its relevant typologies, you should turn your attention to integrating relevant instance data from local datastores or knowledge bases appropriate to the domain. Once fueled by instance data, including attributes and descriptive text, your knowledge system will be a valuable basis for knowledge supervision in machine learning. The outcomes of such learners can usefully aid many knowledge management tasks, importantly including tagging and categorization, and continued growth of your knowledge structures.

Chapter Notes

1. Some material in this chapter was drawn from the author's prior articles at the *AI3:::Adaptive Information*

blog: “Climbing the Data Federation Pyramid" (May 2006); “‘Structs’: Naïve Data Formats and the ABox" (Jan 2009); “Advantages and Myths of RDF" (Apr 2009); “structWSF: A Framework for Data Mixing" (Jun 2009); “Big Structure and Data Interoperability" (Aug 2014); “Logical Implications of Interoperability" (Jun 2015); “How Fine Grained Can Entity Types Get?" (Mar 2016); “Rationales for Typology Designs in Knowledge Bases" (Jun 2016); “Threes All of the Way Down to Typologies" (Oct 2016).

2. Marradi, A., “Classification, Typology, Taxonomy,” *Quality & Quantity*, 1990, pp. 129–157.
3. Wetzel, L., “Types and Tokens,” *The Stanford Encyclopedia of Philosophy*, 2014.
4. Aspects of Peirce’s definition of types have some interesting parallels to type theory (https://en.wikipedia.org/wiki/Type_theory), especially homotopy type theory (https://en.wikipedia.org/wiki/Homotopy_type_theory), that we do not have time to pursue further here. In type theory, well-founded types are ones where we can define objects by primitive recursion and prove properties by induction. (See Thompson, S., *Type Theory and Functional Programming*, Addison Wesley, 1991.) Primitive recursion over boolean properties (which is why dichotomous keys for classification are so useful) is an interesting link to type theory, as are type families and creating new types. Further, some proposed resolutions to improve the representation of subsets in type theories involve representing propositions distinct from types or as types.
5. Philosophers often contrast realism to idealism, nominalism or conceptualism, wherein how the world exists is a function of how we think about or name things. Descartes, for example, summarized his conceptualist view with his aphorism “I think, therefore I am.”
6. Hulswit, M., “Natural Classes and Causation,” *the online Digital Encyclopedia of*, Charles S. Peirce, 2000.
7. Chinchor, N., “Overview of MUC-7,” *MUC-7 Proceedings*, 1997.
8. Brunstein, A., *Annotation Guidelines for Answer Types*, Linguistic Data Consortium, 2002.
9. Sekine, S., “Extended Named Entity Ontology with Attribute Information,” *Proceedings of the Sixth International Language*, 2008, pp. 52–57.
10. For example, try this query, https://scholar.google.com/scholar?q=”fine-grained+entity”, also without quotes.
11. Ling, X., and Weld, D. S., “Fine-Grained Entity Recognition,” *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, 2012.
12. Gillick, D., Lazic, N., Ganchev, K., Kirchner, J., and Huynh, D., “Context-Dependent Fine-Grained Entity Type Tagging,” *arXiv:1412.1820 [cs]*, Dec. 2014.
13. Murty, S., Verga, P., Vilnis, L., and McCallum, A., “Finer Grained Entity Typing with TypeNet,” *arXiv:1711.05795 [cs]*, Nov. 2017.
14. For best interoperability with KBpedia, the SKOS reference should include the SKOS DL version; see M.K. Bergman, “SKOS Now Interoperates with OWL 2”, *AI3:::Adaptive Information* blog, February 10, 2011.
15. Jack Park has questioned why chemistry appears in this schema, while physics and quantum phenomena do not. I agree those topics are worthy, likely under the Natural Matter node at the interface between Firstness and Secondness. Peirce does address these ideas a bit, and even posited something like the Big Bang. (1888, CP 1.411-2) These fundamental perspectives on matter are an active area of research for me, though there are not many crumbs from Peirce on these topics. Still, as we learn more, I can readily see including such topics in the schema. As for chemistry and organic chemistry, we better understand them at present and they importantly demark the transition from natural matter to life. Chemistry is the laws or “habits” (Peirce’s term) for how matter interacts and what products (compounds) may result, so is a natural Thirdness with respect to Matter. Organic chemistry provides the building blocks or possible compounds or substrates to life, so is equivalent to a Firstness regarding organic matter and life.