

# Normalisation of ontology implementations: Towards modularity, re-use, and maintainability

Alans Rector

Department of Computer Science, University of Manchester

Alan L Rector

rector@cs.man.ac.uk,

WWW home page: <http://www.cs.man.ac.uk/mig>

**Abstract.** There is a long history of standard normal forms for information models for databases. However, no analogous notion of normalisation for ontologies has yet emerged. This paper proposes initial criteria for normalisation of “rigorous formal ontologies” to promote a) domain correctness, b) utility, c) re-use, d) modularity, e) maintainability, and f) evolution. The normalisation is in two stages. For the first — “ontological normalisation” — we accept Welty and Guarino’s analysis. The second — “implementation normalisation” — is the primary focus of this paper. For “implementation normalisation” we propose an approach based on decomposing or “untangling” the ontology into independent disjoint taxonomies which are then recombined using definitions and descriptions which link relevant concepts in the decomposed taxonomies.

## 1 Introduction

The notion of normalising information models for relational databases has long been accepted and is part of most standard texts on database design. Implementations may be de-normalised for efficiency or convenience, but the development of a normalised model is a standard part of the design and analysis of most databases and information systems.

The purpose of this paper is to begin the discussion of the goals of and criteria for analogous normalisation for “rigorous formal ontologies” (to use Uschold’s phrase [28]), specifically for those implemented in description logics or related formalisms. (Without denigrating other usages, for brevity we shall use the word “ontology” solely in this sense in the remainder of this paper.) With the advent of OIL<sup>1</sup> [3] and DAML+OIL<sup>2</sup>, such ontologies are becoming widely used as a basis both for navigation on the Semantic Web<sup>3</sup>, for terminologies, and for structuring of databases and medical records in projects such as *OpenGALEN*<sup>4</sup> [16][18] and *SNOMED-RT/CT*<sup>5</sup> [26]. While much other work on ontologies, such as that of

<sup>1</sup> [www.ontoknowledge.org/oil](http://www.ontoknowledge.org/oil)

<sup>2</sup> [www.daml.org/2001/03/daml+oil-index](http://www.daml.org/2001/03/daml+oil-index)

<sup>3</sup> [www.semanticweb.org](http://www.semanticweb.org)

<sup>4</sup> [www.opengalen.org](http://www.opengalen.org)

<sup>5</sup> [www.snomed.org](http://www.snomed.org)

Guarino and Welty [5] [29], concentrates on issues of abstract meaning, this paper concentrates on practical engineering issues of robust modular implementation of the representation of that meaning in a manner that is understandable and reproducible by knowledge engineers with only modest training.

We have six goals for normalisation of ontology implementations:

**Domain correctness** — that the interpretation of the classification inferred by the ontology reasoner corresponds to what was intended by its builders

**Utility** — that the ontology adds value to the application. Ontologies do not exist for their own sake; their value must be assessed by their utility and fitness for the purposes for which they were designed.

**Modularity** — that the different parts of the ontology can be built separately and later combined without combinatorial explosion.

**Re-use** — that the ontology can be used for a range of purposes and applications, albeit always within some limited scope.

**Maintainability** — that changes do not have unanticipated side effects and do not make unrealistic demands on those maintaining the ontology.

**Evolution** — that the ontology can evolve and be extended smoothly and predictably. We assume a conceptualisation of the world which is “open” and “fractal” and therefore always subject to change and revision. No list of things in the world can ever be assumed complete (except by “fiat”); any concept can be further refined by adding more properties, and any extent further subdivided. (Ontologies for the quantum world will have to be different, but we maintain that these principles hold for our “ordinary” world including molecular biology and medicine.)

The last three goals are really special cases of the third, “modularity”. Large ontologies will only be re-usable, maintainable and evolvable if:

- The parts to be re-used can be identified and separated from parts that are irrelevant.
- The maintenance can be split up amongst different staff who can work independently with predictable effects.
- Modules can evolve independently of each other and new modules be added minimal disturbance to the existing structure.
- All information to be used computationally is explicit in the formal representation rather than being implicit in the words used to label concept.

## 2 Basic Criteria for Normalisation of Ontologies

We propose that the normalisation be divided into two stages: “ontological normalisation” and “implementation normalisation.” The first — “ontological normalisation” — depends on human understanding of the meaning of the concepts to be represented. It is independent of the technology used to implement the ontology — *e.g.* simple directed acyclic graphs, description logics, Conceptual Graphs, etc. The second — “implementation normalisation” — concerns the

details of how that meaning is represented for computational purposes and is highly dependent on the formalism used. The methods we propose here apply specifically to ontologies represented in description logics, conceptual graphs, and related logic-based formalisms in which relationships amongst concepts can be inferred by a classifier or theorem prover such as FaCT [7, 8], RACER [6] or earlier systems such as GRAIL [19] or Loom [9]. Logic based languages with classification open up a new range of choices in how any given ontology is realised because they give implementors a choice of whether to implement any given concept as a “primitive” or via a “definition”. In practice, there is such a wide choice of means to achieve any given end that guidelines are required to choose amongst them. Our suggestions for “implementation normalisation” are based on our experiences in *OpenGALEN* and related projects over a wide variety of fields but mainly focused on biomedicine.

## 2.1 Ontological normalisation

Guarino and Welty [5, 29] provide a vocabulary and set of criteria for ontological normalisation which is essentially implementation neutral. Guarino and Welty’s criteria concern primarily the high level or “top ontology” via a series of meta properties which constrain which abstractions can subsume others. There are numerous other attempts to design high level ontologies including the OpenCyc<sup>6</sup>, the IEE Standard Upper Ontology (SUO)<sup>7</sup>. A comparison of these methods or of alternative upper ontologies is outside the scope of this paper. For purposes of this paper we shall stipulate Guarino and Welty’s approach to serve as at least an “existence proof” that the normalisation proposed can be used with methods developed independently.

## 2.2 Implementation Normalisation

**Formalisms for implementing rigorous formal ontologies** We shall assume the standard formulation and semantics of description logics and their equivalents in OIL/DAML+OIL (see *e.g.* [7]). The common principle of all such formalisms is that the hierarchical relation for the ontology is “is-kind-of” and means logical subsumption. Logical subsumption is equivalent to implication, *i.e.* to say that “B is a kind of A” is to say that “All Bs are As” or more formally in logical notation “ $\forall x.Bx \rightarrow Ax$ ”. Therefore, given a list of definitions, formal descriptions, and axioms, a theorem prover or “reasoner” can infer new subsumptions and verify that asserted subsumptions, concept definitions, and descriptions are logically consistent<sup>8</sup>.

The list of features for definitions, descriptions, and axioms varies between formalisms, as does the vocabulary used. For this paper we we shall assume that they include at least:

<sup>6</sup> [www.cyc.com/cyc-2-1/cover.html](http://www.cyc.com/cyc-2-1/cover.html)

<sup>7</sup> <http://suo.ieee.org/>

<sup>8</sup> Strictly speaking “satisfiable”, *i.e.* that it is possible to construct a model in which the subsumption, definition, and descriptions all hold without logical contradiction

**Primitive concepts** described by *necessary* conditions expressed as boolean combinations of other primitives, descriptors and defined concepts and placed in a subsumption hierarchy of other primitive concepts.

**Composite concepts**<sup>9</sup> defined by *necessary and sufficient* conditions expressed in the same way.

**Roles**<sup>10</sup> which relate concepts and which can themselves be placed in a subsumption hierarchy, and perhaps be declared equal to the inverse of other roles, functional, transitive, or symmetric.

**Descriptors**<sup>11</sup> constructed as quantified role-concept pairs, *e.g.* (*some hasLocation Leg*) meaning “located in *some leg*”, (*all eats vegetables*) meaning “eats *only* vegetables”<sup>12</sup>.

**Axioms** which declare concepts either to be disjoint or to imply other concepts (“be covered by” other concepts in OIL parlance)

Given these features, a reasoning engine can infer a subsumption hierarchy or classification of the concepts defined and described. More accurately, such reasoners infer a lattice, since it is formally closed at the top by the concept of “Everything” and at the bottom by the concept of “Nothing” (usually known simply as “Top” and “Bottom” respectively).

These features also mean that implementors have considerable choice whether to represent a given notion as a primitive concept, composite concept, or descriptor.<sup>13</sup> The goal of normalisation is to constrain these choices so that the implementation of the ontology meets the goals set out in the introduction.

**Criteria for implementation normalisation** We term that part of the ontology consisting only of the primitive concepts as the “primitive skeleton”.

We term that part of the ontology which consists only of very abstract categories such as “Structure” and “Process” which are effectively independent of the domain the “Top level ontology”, and those notions specific to a given domain the “Domain Ontology”.

The essence of our proposal for normalisation is that the primitive skeleton of the Domain Ontology should consist of disjoint homogeneous trees. We picture the skeleton taxonomy as a cascade of disjoint branches. In more detail:

1. The branches of the skeleton taxonomy should form trees, *i.e.* no primitive domain concept should have more than one primitive parent.

<sup>12</sup> In description logic and related formalisms, the meaning of the universal quantifier “all” ( $\forall$ ) is that all of the values for the role must be subsumed by the value concept. This is usually most intuitively expressed using the word “only.” For example *Vegetarian*  $\rightarrow$  (*all eats vegetables*) translates into logic as “ $\forall xy . \text{Vegetarian } x \wedge x \text{ eats } y \rightarrow \text{Vegetable } y$ ”, or in other words: “If it is eaten by a vegetarian, then it must be a vegetable” or more simply “Vegetarians eat only vegetables.”

<sup>13</sup> Note that “descriptors” themselves are first class objects and can be used alone in a definition so that implementors have a choice of whether to implement a notion such as “biological” as a simple primitive, *e.g.* *Biological*, or as a definition, *e.g.* *Biological*  $\doteq$  (*some hasBioStatus Biological*).

2. Each branch of the skeleton taxonomy should be homogeneous and logical, *i.e.* the principle of specialisation should be subsumption (as opposed, for example, to partonomy) and should be based on the same, or progressively narrower criteria, throughout. For example, even if it were true that all vascular structures were part of the circulatory system, placing the primitive “vascular structure” under the primitive “circulatory system structure” would be inhomogeneous because the differentiating notion in one case is structural and in the the other case is functional.
3. The primitive skeleton should clearly distinguish:
  - (a) “Self-standing” concepts<sup>14</sup>, which should form open disjoint taxonomies, *i.e.* the primitive children of each primitive concept should be disjoint, but the list of primitive children should not be considered exhaustive (should not “cover” the parent). Examples of “self-standing” concepts include most “things” in the physical and conceptual world – animals, body parts, people, organisations, ideas, processes, events, etc. They also include less tangible notions such as “style”, “colour,” “risk”, “family history” etc. which can be described in their own right. (See also Section 3.1)
  - (b) “Partitioning” or “Refining concepts”, also known as “value types” and “values”,<sup>15</sup> which should form possibly overlapping, closed taxonomies. That is: a) there should be a taxonomy of primitive “value types” which may or may not be disjoint; b) the primitive children of each value type should form a disjoint exhaustive taxonomy, *i.e.* each value type should be “covered by” its values. Examples of “closed” concepts include most “properties”, e.g. “small, medium, large”, “mild, moderate, severe”, etc. The values themselves are disjoint, because something cannot be, for example, both “large” and “small” (at least in the same context), but the value types are not disjoint because something can be both “small and serious.”

In practice, we recommend that the distinction between “self-standing” and “partitioning” concepts be made a top level division in the upper ontology. However, to avoid commitment to any one upper ontology, we suggest only the weaker requirement for normalisation, *i.e.* that the distinction be made clear by some mechanism.

**Consequences** The first set of consequences of criteria 1 and 3 is that all multiple classification is inferred as the result of definitions and axioms. Further, axioms should never result in subsumption of one primitive concept by another, since this would “denormalise” the ontology. For any two “self-standing” concepts, therefore, either one subsumes the other or they are disjoint. It follows

<sup>14</sup> The naming of this category is problematic. In Guarino and Welty’s terms, they correspond to “types”, “quasi-types”, and certain concepts used in constructing our representation of “formal and material roles”, see section 3.2

<sup>15</sup> in Guarino and Welty’s terms, the values for constructing “attributions” and “mix-ins”

that there must be exactly one most specific self-standing concept of which any self-standing individual is an instance.

A second set of consequences of criteria 1 and 3 is that a) descriptions of primitives should consist of conjunctions of exactly one primitive (excluding *Top*<sup>16</sup>) and zero or more descriptors, b) every primitive open concept should be part of a disjointness axiom with its siblings, and c) every primitive value should be part of a disjoint covering axiom with its siblings.

A final pair of consequences is that if a primitive concept is disjoint from its siblings, its children must also be disjoint, and if a primitive concept is part of a partition, its children must also form a partition. This confines “tangles” to the high level ontology; the skeleton of domain concepts should be “untangled.”

### 3 Rationale and illustrations

#### 3.1 Independence, homogeneity, and “untangling”

**Illustrations from *OpenGALEN*** The great power of “rigorously formal ontologies” is that they can combine separate taxonomies logically to produce classifications which are more complex than could be maintained manually. However, this can only be done correctly if all of the information required for classification is correct, explicit and available to the reasoner rather than tacit in the assumptions behind primitive subsumptions.

Many, perhaps most, large ontologies have taken existing classifications as the starting point for their development. These classifications usually have a long history and have been designed to be interpreted by people rather than used as the basis of computation by machines. This is recognised by librarians’ use of “broader than” and “narrower than” to name the hierarchical relations in their thesauri. Such classifications are usually tangled and inhomogeneous, often mixing subsumption and parthood, *e.g.* the International Classification of Diseases, the British National Formulary’s classification of drugs<sup>17</sup>, the Gene Ontology [1, 2] or the Medical Subject Headings (MeSH) used in Medline and Pubmed<sup>18</sup> [12].

However useful for human interpretation, such tangled classifications cannot be interpreted or maintained formally. For example although there is a named branch for steroids in the British National Formulary classification, steroids also occur in at least five of the fifteen major branches. New classes for new purposes are therefore difficult or impossible to define and maintain — *e.g.* there is no class for identifying notions such as “drugs affecting the mood” (which includes classes of steroids) although this is a key notion for checking contraindications and interactions. The first process in normalisation is therefore to untangle their structure and make it homogeneous.

<sup>16</sup> Adding *Top*, or “everything” as a conjunct to a definition does not restrict the meaning and so is redundant. In other words, *Top* is an identity element in definitions

<sup>17</sup> published bi-annually by the Royal Pharmaceutical Society and available on-line at [www.bnf.org](http://www.bnf.org)

<sup>18</sup> [www.ncbi.nlm.nih.gov/entrez/query.fcgi](http://www.ncbi.nlm.nih.gov/entrez/query.fcgi)

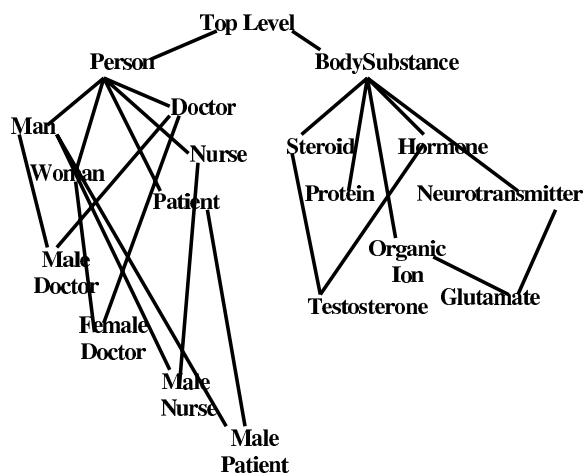


Fig. 1. Fragment of a tangled ontology

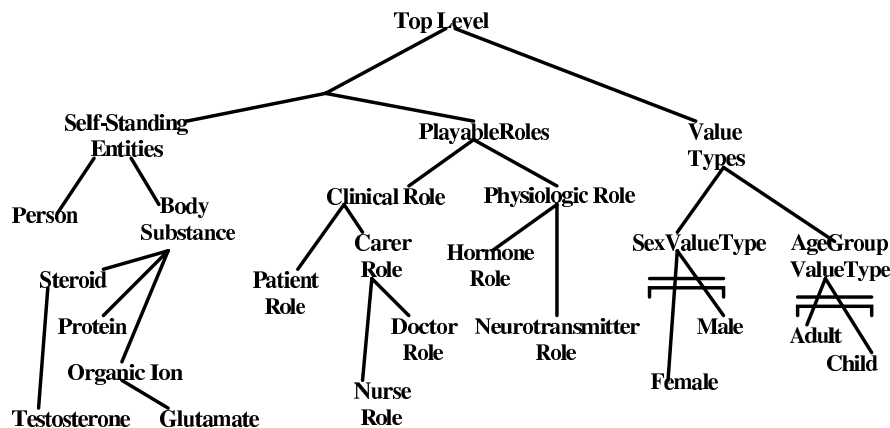
Figure 1 is a simplified extract from two existing classifications which have been used as the starting point for developing ontologies. The left-hand branch is adapted from models used in the NHS health service, the right-hand branch is from an old version of SNOMED. Figure 2 shows the same notions in an ontology normalised for implementation in a description logic formalism using a skeleton of primitives plus a set of definitions. In both versions, *Person* can be classified by age, sex, occupation, etc. *BodySubstance* can be classified by structure, function, origin, etc.

Normalisation as shown in Figure 2, requires separating the different axes into separate taxonomies — of *Age Group*, *Sex*, and *Clinical Role* for *Person*; of chemical structure and *Physiologic Role* for *BodySubstance*.

Separated as shown in Figure 2, each branch of the taxonomy can be modified independently. For example, a taxonomy of *Hormone Roles* could be elaborated, or *Sex* could be separated into *Administrative sex*, *Phenotypic sex*, *Genotypic sex*, etc. and all combinations supported (a practical requirement in some biomedical applications).

The choice of which aspect to make the basis of the primitive skeleton taxonomy is to some degree arbitrary, but it must be a) ontologically “rigid”—*i.e.* be an intrinsic property unchanging during the life of any given instance of the concept—and b) pragmatically “stable”—*i.e.* unlikely to be reformulated in the life of the ontology. For *Person* we chose to take no aspect as primitive and constructed all subconcepts by definition, since no aspect, even *sex* fit all these criteria. For *BodySubstance*, we chose structure as being the more intrinsic and “rigid” — the same substance can have many functions but only one structure.<sup>19</sup>

<sup>19</sup> at least at this granularity. If we were dealing with the folding of proteins, this choice would have to be revised.



Man  $\doteq$  (and Person (some hasSex Male))  
 MaleDoctor  $\doteq$  (and Man (some playsRole DoctorRole))  
 Hormone  $\doteq$  (and BodySubstance (some playsRole HormoneRole))  
 Testosterone  $\sqsubseteq$  (and Steroid (some playsRole HormoneRole))  
 Glutamate  $\sqsubseteq$  (and OrganicIon (some playsRole NeurotransmitterRole))

Fig. 2. Normalised skeleton from Fig 1 with marked coverings for value types and example definitions and descriptions

**Illustration of implementation normalisation as an extension to Guarino and Welty’s ontology “cleaning”** As a further illustration, this style of untangling by extracting the primitive skeleton can be thought of as the next step towards implementation from Guarino and Welty’s notion of ontology “cleaning.” For example, in Figure 3, adapted from [5] (fig 6), *Food* is listed as part of the back bone, whereas after the next step of normalisation, *Food* would almost certainly be represented as a defined concept as shown in Figure 4, since otherwise an instance of *Food* would be an instance of two primitives: *Food* and *Amount of matter*. Similarly, *Group of people* appears in the in Figure 3 as a single entity, but we would implement it as a defined concept as shown in Figure 4. This would allowing separate taxonomies for *Person* and for different variants of *Group*, e.g. for groups with identity not exclusively based on membership such as “flocks.”

Relations which would otherwise have to be captured later, such as that between *Group* and *Person* and between *Food* and *Eating*, are captured directly in the respective definitions. Furthermore, having to define *GroupOfPeople* forces the ontology author to clarify ambiguities — does *GroupOfPeople* mean a group *only* of people? Of *some* people along perhaps with some other things? Or a group *mostly* of people (a notion not easily expressed in description logics at all)? Does a “group” have to have at least one member? More than one? (We have chosen a definition here representing “at least one member and only person members” but the original leaves this open.)

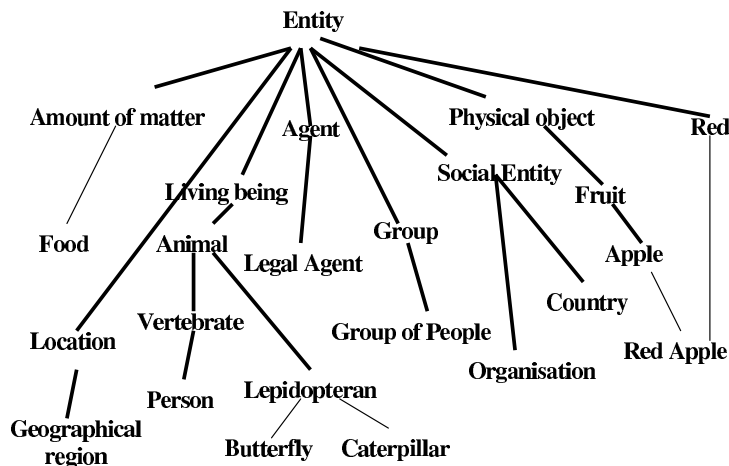


Fig. 3. Guarino and Welty’s version (adapted). The heavy lines indicate the “backbone”

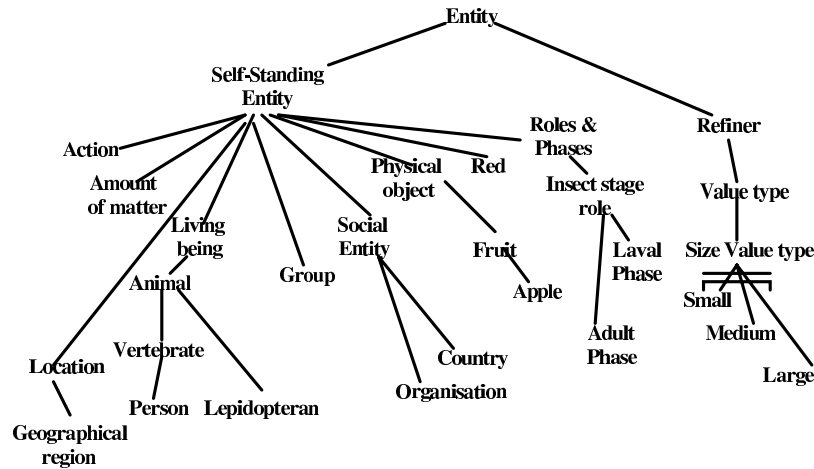
Extracting the skeleton in this way strongly encourages developers to follow many of Guarino and Welty’s precepts. For example the “being food”<sup>20</sup>—(some *isEatenBy Animal*)—is clearly distinguished from the “substance being eaten”<sup>21</sup>—(and *AmountOfMatter (some isEatenBy Animal)*).<sup>22</sup> Similarly no engineering problem is posed by what Guarino and Welty term “phased sortals”—*i.e.* things which fall into different types at different phases in their lifetime *e.g.* “butterfly” and “caterpillar”. Phased sortals are represented as compositions analogous to *Caterpillar* in Figure 4.

**Fundamental rationale: minimising implicit information** Put another way, this approach seeks to minimise the implicit information in the implementation and avoid unintended consequences resulting from it. We cannot define everything. Some things must be primitive. Some concepts are best seen as natural kinds, and some concepts will be left primitive in any given implementation because it is not worth the effort to define them within the intended scope and use of the ontology. In effect, with or without normalisation, for each primitive concept there is an implicit notion which differentiates it from each of its primitive parents (the Aristotelian “differentia” if you will). However, since since these notions are implicit, they are invisible to the human developer and mechanical reasoner alike. They are therefore likely to cause confusion to developers and result in missed or unintended inferences in the reasoner. The requirement for

<sup>20</sup> a “material role” in Guarino and Welty’s terms

<sup>21</sup> a “quasi-type” in Guarino and Welty’s terms

<sup>22</sup> It is explicit in Guarino and Welty’s text that by “Agent” they mean the formal role “being an agent” as shown. Perhaps deliberately, they left *Food* ambiguous and so we chose to use it here for illustration



Agent  $\doteq$  (some agentFor Action)  
 Caterpillar  $\doteq$  (and Lepidoteran (some inPhase LarvalPhase))  
 RedApple  $\doteq$  (and Apple (some hasColour Red))  
 Food  $\doteq$  (and AmountOfMatter (some isEatenBy Animal)))  
 BigApple  $\doteq$  (and Apple (some hasSize Large))  
 GroupOfPeople  $\doteq$  (and Group (all hasMember Person)(some hasMember Person))

Fig. 4. Extracted normalised skeleton plus example value type and definitions

independent homogeneous taxonomies amounts to a requirement that a) there be exactly one differentiating notion per primitive concept which differentiates it from a single primitive parent which must be unique; b) all differentiating notions in a given part of the taxonomy be of the same sort – *e.g.* all structural, all function, etc. Given this requirement, confusion and unintended inferences are minimised.

**“Flavours” of is-kind-of** This technique can also be seen as a means of satisfying common request from knowledge engineers — to be able to have different “flavours” of *is-kind-of* *i.e.* to be able to label the “is-kind-of” links to show their significance. In effect, the normalisation technique proposed here allows exactly one unlabelled *is-kind-of* link. All others are inferred from descriptors and axioms. The majority of inferred “is-kind-of” links arise from simple existential descriptors and may be explained as being equivalent to “labeled” “is-kind-of” links. (Those that arise from more complex inferences require more sophisticated explanation facilities beyond the scope of this paper.)

### 3.2 The fractal nature of knowledge and open taxonomies

Knowledge is fractal, at least in biomedicine. Any self-standing concept — *e.g.* bones, diseases, species, organisations, etc.— can be further refined. Any such list

of kinds is open both to new knowledge and to the addition of items considered too trivial to include originally. Put another way, it is never safe to assume that a concept will remain a leaf node in future versions of the ontology.

Since the lists cannot be complete, a closed world cannot be assumed, *i.e.* it is not generally safe from “A and not some known kinds of A” to infer “some other known kinds of A”. More formally, given  $A \sqsupseteq B_1, \dots, B_{n-1}, B_n$ , from  $A \wedge \neg B_1 \wedge \dots \wedge \neg B_{n-1}$  it is not safe to conclude  $B_n$ . By contrast, when dealing with value types — for example a *SizeValueType* subsuming *small, medium, large* — the value types are defined to be exhaustive, at least for the purposes of a given ontology. Therefore such inferences are safe.

There are a few qualities that seem to defy this pattern, in particular colour. Since they are perceived directly and the list of colours cannot usually be considered exhaustive, we argue that they should be treated as “self-standing concepts” of the same status as other representations of things in the world.

Furthermore, as Barry Smith points out [24] many other features of our conceptual world depend on “flat boundaries” — arbitrary distinctions in some continuum. For some such notions, *e.g.* the official boundaries of legal entities, there may be a closed finite list — *e.g.* the list of countries in the European Union or states in the United States. However, in engineering terms, representing such lists as “closed” has few advantages and several disadvantages. Firstly, such lists change over time. Secondly, in applications in which the such notions are peripheral, as in the case of nations in a health and bioscience ontology, there are few circumstances in which we would wish to make inference from the closed list such as “Not England implies France or Germany or ...”. Thirdly, the ability to make such inferences has expensive and global computational costs — *i.e.* inferences we want to make commonly become much more expensive in order to make inferences we wish to make rarely, if ever.

It is considerations such as these which have led us to use the labels of “self-standing” vs “partitioning” concepts. Alternatives such as “independent” vs “dependent” or “open” vs “closed” either carry too many overloaded meanings from philosophy and/or database usage or miss the desired meaning or both.

## 4 Experience and relation to other methods

Experience and several experiments support our contention that these techniques are a major assistance in achieving the six goals set out in Section 1: correctness, utility, modularity, re-use, maintainability, and evolution.

The approach to “normalisation” set out here, which we also refer to as “untangling”, has been used throughout *OpenGALEN* and related ontologies over a period of nearly fifteen years [18]. In fact, many features of the *GALEN Representation and Implementation Language* (GRAIL) were designed around these precepts. In GRAIL: a) all primitive concepts are disjoint unless otherwise specified; b) all definitions consist of a single “base concept” followed by zero or more descriptors<sup>23</sup>.

<sup>23</sup> termed “criteria” in GRAIL’s parlance

The approach to normalisation or “untangling” has proved easy to explain to new ontology developers and has been one of the key strategies for supporting loosely coupled distributed development. (The other is the use of “intermediate representations” for knowledge acquisition. [16][17])

The *OpenGALEN* related ontologies now total on the order of 35,000 primitive and defined concepts and perhaps 100,000 descriptors linking those concepts into a densely interconnected network. The two primary topics have been surgical procedures [20,15] and the ontology and knowledge base of drug information to underpin the Prodigy decision support system for prescribing in UK general practice [25]. The ontologies have been described informally in [13][21] and more formally in [14].

Throughout this experience, we have found no situation in which the suggested normalisation could not be performed. The requirement to limit the “primitive skeleton” to simple disjoint trees may seem restrictive, but it does not actually reduce expressiveness. In our experience violation of this principle almost always indicates that tacit information is concealed which makes later extension and maintenance difficult<sup>24</sup>. Interestingly Gu and her colleagues have independently proposed *post hoc* decomposition into trees of ontologies represented as directed a cyclic graphs in frame-like systems as a means to improve the maintainability of large ontologies [4].

In addition to the internal tests within the GALEN project, the GALEN procedure ontology was compared against the manually constructed ontology from the Clinical Terms (Read Codes) Version 3 (CTv3) [22]. Overall, there was a high degree of concurrence, but three main types of discrepancy were found. Firstly, *OpenGALEN*’s codes were consistent with respect to usages which were used variably in CTv3. For example, in CTv3, “Excisions” were sometimes kinds of “Removals” and sometimes *vice versa*. Secondly, when changes had to be made, they could always be made in just one place in *OpenGALEN*, whereas many occurrences had to be changed in CTv3. For example, which structures are within vague anatomic regions such as the axilla or perineum differed systematically between the two models, but changing the *OpenGALEN* model was much easier. Thirdly, the *OpenGALEN* ontology was much smaller since many notions which had been expanded combinatorially in the CTv3 version needed to be defined in *OpenGALEN* only when required by an application.

Further evidence for the robustness of the approach comes from the use of the *OpenGALEN* ontology as the basis for the drug information knowledge base underpinning the UK Prodigy Project[25]. Untangling the existing source classifications has made expressing new abstract concepts needed by applications simple and reliable whereas previously it was tedious and error prone.

Perhaps the most dramatic example justifying the methodology was recent work on the “simple” problem of forms, routes, and preparations for drugs. Although there are only a few hundred concepts, classification had resisted con-

---

<sup>24</sup> The only exception is the use of additional parents to simulate disjunctions which are not otherwise supported in GRAIL. This usage is carefully controlled and has been made made obsolete by more recent languages such as DAML+OIL.

certed efforts by standards bodies for over two years because of severe “tangling.” Restructuring as a normalised ontology solved the problem in weeks [30].

More recently, we have applied the technique to formalising the concepts in the Digital Anatomist Foundational Model of Anatomy[10][23]<sup>25</sup> where they captured naturally the careful definitions of “organ”, “organ part”, etc. that form the foundation of that ontology. This work illustrates the importance of “untangling” in promoting modularity. Preliminary results indicate that it would be straightforward to extend the Foundational Model with functional and clinical notions precisely because its classification is based solely on structure. Had its classification included functional elements, then there would be no automatic way to determine whether new notions of function conflicted with existing notions.

Still more recently in the Gene Ontology Next Generation (GONG) project, the requirement to break out the chemical structure information from the overall ontology and reformulate it as a series of independent trees for modularity and explicitness has become clear [32].

Utility is the most difficult of the criteria to measure. However, indexing information is a key function, perhaps the key function, of ontologies for the Semantic Web.

In indexing information with an description logic based ontology, the lattice computed by the description logic reasoner is used as a “conceptual coat rack” on which to hang pointers to other information, precisely in the way that information is put onto the class structure of a frame system. The general operation is not a query to the description logic itself but rather to attach pointers to the subsumption lattice computed by the reasoner. The retrieval operation is then to find the set of most specific pointers of a given type for the target concept using the standard Touretzky criteria[27]. Although a single concept may have many inferred parents — an average of just over three in *OpenGALEN* with a maximum of over twelve — each parent is of a clearly different type. Consequently the set of properties or pointers attached to each parent — what is said about each each parent — is likely to be different. For example parents inferred because of a drug’s chemical structure will have different properties from those acquired because of its physiological action. As a result, in most cases the set of most specific pointers has only one member. In cases where there the set does contain more than one member, relatively simple additional reasoning, outside the description logic framework, usually suffices to manage the result. (See [16]) (Put another way, “Nixon diamonds” involving such indexing pointers are rare in normalised ontologies.)

*OpenGALEN* based ontologies have been used in three applications involving indexing. In each case the process of “normalising” or “untangling” has brought dramatic gains. The first is the PEN&PAD user interface for UK General Practitioners which has been well validated in repeated studies and was eventually commercialised [11]. PEN&PAD is based on assembling fragments of data entry forms indexed on the ontology to construct a complete data input form adapted

<sup>25</sup> <http://sig.biostr.washington.edu/projects/da/>

to the particular disease, clinical setting, and user preference. The second is the UK Prodigy drug ontology in which untangling greatly facilitated indexing indications, contraindications, and other related information [25][31]. The third is the mapping from *OpenGALEN* to the International Classification of Diseases (ICD) whose structure includes many rubrics of the form “Hypertension excluding pregnancy”. No case has been found where an “excluding” clause was not catered for automatically by the standard indexing procedure. In all such cases there was a rubric elsewhere in ICD such as “Hypertension in Pregnancy” which when classified in *OpenGALEN* intervened between the parent concept and the target concept so that the strategy of taking the most specific mapping indexed by the ontology produces a unique result [18].

## 5 Conclusion

The ability of logical reasoners to link independent taxonomies to re-use them to form new richer ontologies is extremely powerful, but it only works if the source taxonomies are logically correct and all information is explicit so as to be available to the reasoner. The requirements for homogeneity and independence—“untangling”—are designed to promote explicitness and correctness.

The approach presented here is based on fifteen years’ experience in the development and use of large (>35,000 concept) biomedical ontologies including *OpenGALEN* [18] and more recently the UK Drug Ontology [25].

Nonetheless, these are merely initial suggestions for criteria for normalising ontologies. Other workers will no doubt propose further criteria or challenge these. Some may even object to the use of the word “Normalisation” as being too grand for this stage of development. However, the large range of options provided by description logic based formalisms means that knowledge engineers and tool designers need strong guidance on how to use these formalisms to implement robust, maintainable, ontologies. We believe that if the potential of ontologies implemented in description logics is to be realised, then normalising of their implementation needs to become as well defined and routine as normalising information models for database design.

## References

1. The Gene Ontology Consortium. Gene ontology: tool for the unification of biology. *Nature Genetics*, 25:25–29, 2000.
2. The Gene Ontology Consortium. Creating the gene ontology resource: design and implementation. *Genome Research*, 11:1425–1433, 2001.
3. D. Fensel, F. van Harmelen, I. Horrocks, D. McGuinness, and P. F. Patel-Schneider. OIL: An ontology infrastructure for the semantic web. *IEEE Intelligent Systems*, 16(2):38–45, 2001.
4. H. H. Gu, Y. Perl, J. Geller, M. Halper, and M. Singh. A methodology for partitioning a vocabulary hierarchy into trees. *Artificial Intelligence in Medicine*, 15(1):77–98, 1999.

5. N. Guarino and C. Welty. Towards a methodology for ontology-based model engineering. In *ECOOP-2000 Workshop on Model Engineering*, Cannes, France, 2000.
6. V. Haarslev and R. Moeller. Expressive abox reasoning with number restrictions, role hierarchies, and transitively closed roles. In AG Cohn, F Giunchiglia, and B Selman, editors, *Proceedings of the Seventh International Conference on Knowledge Representation and Reasoning (KR2000)*, pages 273–284, San Francisco, CA, 2000. Morgan Kaufmann.
7. I. Horrocks. Using an expressive description logic: Fact or fiction. In A G Cohn, L K Schubert, and S C Shapiro, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Sixth International Conference on Knowledge Representation (KR 98)*, pages 634–647, San Francisco, CA, 1998. Morgan Kaufmann.
8. I. Horrocks, U. Sattler, and S. Tobies. Practical reasoning for very expressive description logics. *Journal of the Interest Group in Pure and Applied Logics (IGPL)*, 8(3):293–323, 2000.
9. R. MacGregor. Inside the loom description classifier. *SIGART Bulletin*, 2(3):88–92, 1991.
10. J. L. V. Mejino and C. Rosse. Conceptualization of anatomical spatial entities in the digital anatomist foundation model. *J. of the American Medical Informatics Association*, (Fall Symposium Special Issue):112–116, 1999.
11. W. Nowlan, A. Rector, S. Kay, B. Horan, and A. Wilson. A patient care workstation based on a user centred design and a formal theory of medical terminology: PEN&PAD and the SMK formalism. In *Proc. of the 15th Annual Symposium on Computer Applications in Medical Care (SCAMC'91)*, pages 855–857, 1991.
12. A. Rector. Thesauri and formal classifications: Terminologies for people and machines. *Methods of Information in Medicine*, 37(4–5):501–509, 1998.
13. A. Rector, A. Gangemi, E. Galeazzi, A. Glowinski, and A. Rossi-Mori. The GALEN CORE Model schemata for anatomy: Towards a re-usable application-independent model of medical concepts. In P Barahona, M Veloso, and J Bryant, editors, *Twelfth International Congress of the European Federation for Medical Informatics, MIE-94*, pages 229–233, Lisbon, Portugal, 1994.
14. A. Rector and J. Rogers. Ontological issues in using a description logic to represent medical concepts: Experience from galen. *Methods of Information in Medicine*, (in press), 2002.
15. A. Rector, A. Rossi Mori, F. Consorti, and P. Zanstra. Practical development of re-usable terminologies: Galen-in-use and the galen organisation. *International Journal of Medical Informatics*, 48(1-3):71–84, 1998.
16. A. Rector, C. Wroe, J. Rogers, and A. Roberts. Untangling taxonomies and relationships: Personal and practical problems in loosely coupled development of large ontologies. In Y. Gil, M. Musen, and J. Shavlik, editors, *Proc. of the 1st Int. Conf. on Knowledge Capture (K-CAP 2001)*, pages 139–146. ACM, 2001.
17. A. L. Rector, P. E. Zanstra, W. D. Solomon, J. E. Rogers, R. Baud, W. Ceusters, W. Claassen, J. Kirby, J.-M. Rodrigues, A. R. Mori, E. Haring, and J. Wagner. Reconciling users' needs and formal requirements: Issues in developing a re-usable ontology for medicine. *IEEE Transactions on Information Technology in BioMedicine*, 2(4):229–242, 1999.
18. A.L. Rector. Clinical terminology: Why is it so hard? *Methods of Information in Medicine*, 38:239–252, 1999.
19. A.L. Rector, S. Bechhofer, C.A. Goble, Nowlan W.A. Horrocks, I., and W.D. Solomon. The grail concept modelling language for medical terminology. *Artificial Intelligence in Medicine*, 9:139–171, 1997.

20. J. M. Rodrigues, B. Trombert-Paviot, R. Baud, J. Wagner, P. Rusch, and F. Meusnier. Galen-in-use: An eu project applied to the development of a new national coding system for surgical procedures: Ncam. In C Pappas, M Maglavera, and J R Scherrer, editors, *Medical Informatics Europe '97*, pages 897–901, Porto Carras, Greece, 1997. IOS Press.
21. J. Rogers and A. Rector. The GALEN ontology. In J Brender, JP Christensen, J-R Scherrer, and P McNair, editors, *Medical Informatics Europe (MIE 96)*, volume A, pages 174–178, Copenhagen, 1996. IOS Press.
22. J. E. Rogers, C. Price, A. L. Rector, W. D. Solomon, and N. Smejko. Validating clinical terminology structures: Integration and cross-validation of read thesaurus and GALEN. In *Proc. American Medical Informatics Association Annual Symposium 1998 (AMIA-1998)*, pages 845–849, 1998.
23. C. Rosse, I. G. Shapiro, and J. F. Brinkley. The digital anatomist foundational model: Principles for defining and structuring its concept domain. In *Proc. American Medical Informatics Association Annual Symposium 1998 (AMIA-1998)*, pages 820–824, 1998.
24. B. Smith and A. Varzi. Fiat and bona fide boundaries. *Philosophy and Phenomenological Research*, 60(2):401–420, 2001.
25. D. S. Solomon, C. Wroe, J. E. Rogers, and A. Rector. A reference terminology for drugs. In *Proc. American Medical Informatics Association Annual Symposium 1999 (AMIA-1999)*, pages 152–155, 1999.
26. K.A. Spackman, K.E. Campbell, and R.A. Côté. SNOMED-RT: A reference terminology for health care. In *Proc. American Medical Informatics Association Annual Symposium 1997 (AMIA-1997)*, pages 640–644, 1997.
27. DS Touretzky. *The Mathematics of Inheritance Systems*. Morgan Kaufmann, Los Altos, CA, 1986.
28. M. Uschold and M. Gruninger. Ontologies: principles, methods and applications. *Knowledge Engineering Review*, 11(2), 1996.
29. C. Welty and N. Guarino. Supporting ontological analysis of taxonomic relationships. *Data and Knowledge Engineering*, 2001.
30. C. Wroe and J. Cimino. Using OpenGALEN techniques to develop the HL7 drug formulation vocabulary. In *Proc. American Medical Informatics Association Annual Symposium 2001 (AMIA-2001)*, pages 766–770, 2001.
31. C. Wroe, W. Solomon, A. Rector, and J. Rogers. Inheritance of drug information. In *Proc. American Medical Informatics Association Annual Symposium 2000 (AMIA-2000)*, page 1158, 2000.
32. CJ Wroe, R Stevens, C.A Goble, and M Ashburner. An evolutionary methodology to migrate the gene ontology to a description logic environment using daml+oil. In *Proceedings of the 8th Pacific Symposium on Biocomputing (PSB)*, page (in press), Hawaii, 2003.