Data & Knowledge Engineering 66 (2008) 147–162

Contents lists available at ScienceDirect

Data & Knowledge Engineering

journal homepage: www.elsevier.com/locate/datak

In situ migration of handcrafted ontologies to reason-able forms

Mikel Egaña Aranguren *, Chris Wroe, Carole Goble, Robert Stevens

School of Computer Science, University of Manchester, Oxford Road, Manchester M13 9PL, UK

ARTICLE INFO

Article history:
Received 18 October 2007
Accepted 21 February 2008
Available online 29 February 2008

Keywords:
DAG
OWL
Ontology migration
Ontology enrichment
Ontology untangling
Feature escalator
Gene ontology

ABSTRACT

A methodology for in situ migration of a handcrafted Directed Acyclic Graph (DAG), to a formal and expressive OWL version is presented. Well-known untangling methodologies recommend wholesale re-coding. Unable to do this, we have tackled portions of the DAG, lexically dissecting term names to property-based descriptions in OWL. The different levels of expressivity are presented in a model called the "feature escalator", where the user can choose the level needed for the application and the expressivity that delivers requirement. The results of applying the methodology to some areas of the gene ontology (GO) demonstrate the validity of the methodology.

1. Introduction

In ontology development there is often a tension between large-scale production of relatively simple representations of knowledge and small-scale, rich and intricate representations of some of the same knowledge. This tension can be particularly great when it is the knowledge holders themselves who build the ontology. Lacking the skills in knowledge representation (KR), a simple representation form is adopted. Over time, however, the need for richer representations will arise, but the ontology is stuck in its simpler, *lean* form.

In this paper, we propose a methodology that exploits the range of expressivity in the web ontology language (OWL) [1] to migrate from a simple, lean hierarchical representation to one that uses a greater proportion of the *richer* facilities of the description logic (DL) version of OWL (OWL-DL). Each stage in between has its benefits and our methodology advocates that movement to the next stage should be based on these benefits so that judgement can be used in deciding how to migrate to another level of expressivity in a representational form.

Ontologies should provide a shared understanding of a domain to facilitate the communication and integration of data between people and machines [2]. This is achieved by offering a shared conceptualisation of a domain of interest for both humans and computers [3]. Classes of instances in the domain of interest are representations of the concepts that capture the understanding of that domain. The ontology can also capture relationships between instances of those classes. Representations of concepts (classes and individuals), as well as relationships, can be labelled to provide a collection of terms or a vocabulary with which to describe that domain of interest.

The conceptualisation that captures the community knowledge in an ontology needs to be encoded or represented so that it can be understood and communicated to those sharing the understanding it captures. This is the job of the chosen KR language [4]. As with programming languages, different KR languages have different capabilities: semantics, expressivity and reasoning support. Ideally, the language chosen should reflect the purpose to which the ontology will be put and the lan-

* Corresponding author.
E-mail address: mikel.eganaaranguren@cs.man.ac.uk (M. Egaña Aranguren).

0169-023X/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.datak.2008.02.002

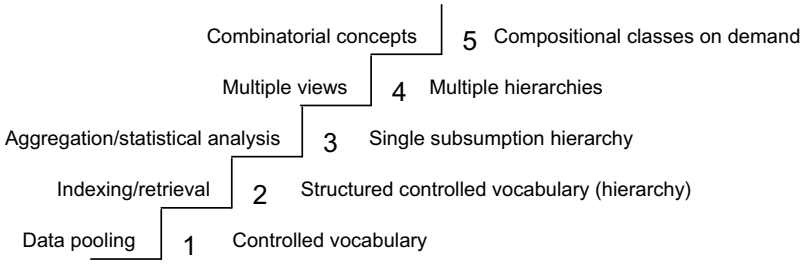

Fig. 1. The feature escalator—as sophistication increases range of use widens.

guage should support these application needs. Application areas can be divided into three broad categories [5]: (1) neutral authoring, (2) common access to information, and (3) indexing for search.

The combination of application area and domain of interest will determine what is represented in the ontology and how complex the encoding should be in a particular representation. As a result, though there is much argument about whether some artefacts actually are ontologies, they span a wide range of styles of encoding, even within a particular KR language. This span moves from a simple list of terms (with or without definition); through trees of terms; multiple hierarchies; frames based representations; to DL [4,6]. The needs and scope of the application and ontology can, however, change and the need for added complexity increase as requirements grow and change. We use the concept of a "feature escalator" to capture this idea of what a particular representational feature brings in terms of application benefits (Section 2).

When a community sets out to build such a ontology to promote understanding and uptake, it is necessary to constrain the complexity and scope of the ontology. As the use of the ontology becomes more sophisticated, so must its representation. How can a community start simple, but leave the door open for more complexity in the future, without having to start again? This paper outlines a methodology to migrate a simple *lean* ontology into a more formal, *richer* ontology using a more complex and explicit representation.

The methodology is illustrated by the Gene Ontology Next Generation (GONG) project,[1] which has applied this methodology to the migration of the current Gene Ontology (GO) [7] (see Section 3) into a more expressive DL environment [8]. GONG was a two year DARPA DAML programme project to investigate how an existing ontology could be migrated to a more expressive, semantically strict language, such as OWL and what benefits that move would bring (see Section 2). The GONG methodology aims to mitigate the costs of moving to a more strict representation and richer encoding by providing an evolutionary approach in which the process can be staged both in coverage and complexity of definitions. The GONG methodology itself is described in Section 4 and its implementation in Section 5. Results from the case study can be seen in Section 6. Finally, the outcomes and wider implications of this work are discussed in Section 7.

2. The range of terminology representations and their uses

If we are to make a judgement about how and why to migrate from a lean to rich ontology by capitalizing on the expressivity of a language such as OWL-DL, it is important to understand the wide spectrum of ontology-like resources that can be created by OWL and other KR languages. These resources can be arranged on a "feature escalator" (see Fig. 1). The feature escalator resembles Welty et al.'s "ontology spectrum" [9], but differs significantly in its motivation. Welty et al.'s purpose was to show that different styles of knowledge artefact, from term catalogue to logic ontology, had different representational needs and requirements. Placing a line on this spectrum to distinguish "ontology" from "non-ontology" is very much in the eye of the beholder. In the feature escalator, in contrast to the ontology spectrum, the concern is not with representation and what counts as an ontology. Instead the concern of the escalator is with what the basic application needs are at each stage on the escalator.

Moving upwards on the escalator is triggered by changing requirements and results in the addition of novel features to support those requirements. The goal of an ontology developer is to choose a level that supports their community's needs. The steps on the escalator are:

(1) A controlled vocabulary;
(2) A structured controlled vocabulary (hierarchy);
(3) A single subsumption hierarchy (single inheritance);
(4) Multiple hierarchies (multiple inheritance);
(5) Compositional classes on demand.

[1] http://www.gong.manchester.ac.uk.

Step 1: A controlled vocabulary. A controlled vocabulary is a constrained list of terms used to describe qualitative data. When a community agrees on such a list for aspects of their data, it is possible to pool data across the community.

Step 2: A structured controlled vocabulary. If the number of vocabulary terms grows, there is usually a move to organise them into related groups to form some kind of hierarchy. In a thesaurus-like ontology the relationship between parent and child terms is one of a vague `narrower than` or `broader than`. The shape of the tree is designed to (i) assist manual navigation around the tree and (ii) help retrieval of items associated with terms.

The medical subject headings (MeSH) [10] is an example of a controlled vocabulary in a thesaurus structure used to assign keywords to life science publications. A parent–child relationship is added if a search for documents with the parent term should return documents annotated with the child term. Therefore, `Accident Prevention` (`G03.850.110.060`) is a child of `Accidents` (`G03.850.110`), despite not being a subclass of the parent term. This provides few problems either if the hierarchy is used for retrieval only or is to be interpreted only by humans.

Step 3: A single subsumption hierarchy. Many ontology-like resources have been used, not just for manual navigation and retrieval, but for statistical aggregation of data.

The London Bills of Mortality[2] were created as far back as the 17th Century, to record the cause of death of Londoners, and used a controlled vocabulary to do so. Its modern day counterpart, the International Classification of Diseases[3] (ICD-version 10) arranges *circa* 22,000 terms in a hierarchy where the pure subsumption (class–subclass) relationships are more uniformly distributed than the parent–child relationships found in MeSH.

In a subsumption relationship the child implies the parent. All instances denoted by the child are also instances of the subsuming (parent) class. Providing statistics for accidents by aggregating the frequency of child concepts depends on this subsumption relationship. It would not be correct to include `Accident Prevention` events in the statistics of `Accidents` as would happen with the relationship described in the MeSH terms above. So, with a strict subsumption hierarchy, where the relationship forming the backbone of the ontology has a formal definition, the representation has a specific meaning that can be exploited when asking queries.

Step 4: Multiple subsumption hierarchies. When the nature of concepts in the ontology becomes complex, there are multiple ways in which they can be classified using just subsumption relationships. For example a classification of diseases could be organised by their cause or the system in which they occur in the body. Each parent–child relationship implicitly captures an aspect of the term's definition. For example, `Stomach cancer` is a compositional term; it is made up of an *anatomical* term and a *disease* term. Each term can be classified in its own right: `Stomach` in an `Anatomy` axis and `Cancer` in a `Disease` axis. Providing these multiple subsumption relationships (multiple inheritances) allows the user to either search for or aggregate data along different axes.

This added functionality, however, comes at a cost to the maintainers of the ontology. Maintaining an exhaustive multiple subsumption hierarchy has been shown to be difficult, leading to a significant rate of omitted subsumption relationships [11,8].

Human users may have a limit to the number of views they require, but application developers providing decision support can vastly inflate the complexity. These application developers write rules of the form "if situation X occurs suggest action Y". An example, from medicine, is "If patient has angina, suggest prescribing a betablocker". To minimize the number of rules, they write them at the highest level of abstraction possible. "X" and "Y" can be implemented as abstract concepts in an ontology (such as `angina` in a disease ontology, and `betablocker` in a drug ontology). These must then be linked via subsumption relationships to more specific classes of that situation (more specific class of `disease`-s or `drug`-s). The number of such abstractions can grow enormously, giving rise to a structure that is too complex to navigate around by humans but is always being extended to support the next decision support rule.

Step 5: Classes on demand—compositional classes. As the complexity of the ontology increases there is often a need to offer all potential combinations possible along the multiple axes described in Step 4. For example, in ICD 9 all combinations of accident type, occupation, location, etc. are enumerated for use in recording statistics. Not only is the collection of over 900 kinds of accident incoherent to a user or ontology developer, but many of the combinations are bizarre. For instance, there is a concept for `accident while riding an exploding bicycle`.

Instead of attempting to exhaustively provide all possible combinations, it is possible to offer an ontology that is dynamic rather than a static artefact. In a dynamic ontology, the building blocks of the potential classes in a system are described, as are the relationships between them. When a new kind of term is needed, the building blocks are *composed* to dynamically build a new term or class. This is the approach taken by the DL systems (see below) [12,13,6]. In such systems, a class is described in terms of the properties it holds. These logical descriptions can be used to classify a new class *on the fly*. Such a system has obvious benefits, but is at the top of the feature escalator.

In Steps 3–5, strict `is-a` is the only ontological structure taken into account. The `is-a` or taxonomic relationship is the one that gives an ontology its structural form. Other relationships are, of course, important, but the presence of these for a particular class is largely governed by inheritance along the structure provided by the taxonomy. Whilst the other relationships are of great significance in an ontology, the feature escalator concentrates on these fundamental, structural relationships upon which other properties may be placed (as is indeed the purpose of the GONG process).

[2] http://en.wikipedia.org/wiki/Bills_of_Mortality.
[3] http://www.who.int/classifications/apps/icd/icd10online/.

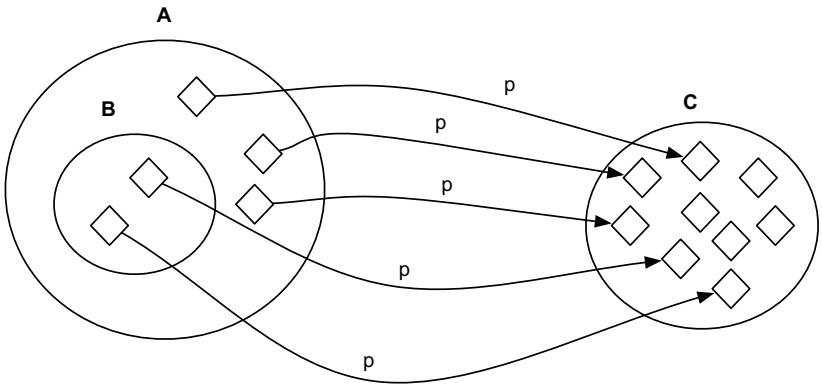

Fig. 2. An illustration of the set based semantics of DL such as OWL-DL. Sets (classes) A, B and C contain instances. All instances in set B are also members of the superset A, therefore, B is a subclass of A. A property relates one instance to another.

Moving up the escalator from a lean ontology to one that is richer requires more expressive power. It is this expressive power that enables features and also allows more to be stated more precisely within the ontology.

2.1. OWL and the feature escalator

This feature escalator can be instantiated in anything from plain text (for the simplest list of words) up to using the variety of constructs available in a sophisticated KR language such as a DL based language like OWL-DL. The web ontology language (OWL) [1] is a W3C[4] recommendation for a language to build ontologies for the Semantic Web [14]. OWL is available in three forms, depending on expressivity and computational tractability:

OWL-Lite. OWL-Lite is the less expressive form.
OWL-DL. OWL-DL maps to the DL $\mathcal{SHOIN}(\mathcal{D})$.[5] It is more expressive than OWL-Lite specially with regards to class constructors.
OWL-Full. OWL-Full is the most expressive OWL type and the computational tractability is not guaranteed.

DLs are a decidable fragment of first-order logic and thus have a well-defined, two-valued semantics, i.e., they allow to be express that which is universally true [6]. In OWL-DL, the basic unit of an ontology is a "class" that represents a set of individuals, its "instances". Moreover, we consider "properties" that represent (binary) relations between individuals (see Fig. 2).

OWL-DL classes can be linked by the subclass axiom—that is, all instances of the subclass are also instances of the superclass. Obviously OWL-DL can represent any simple, tree-like subsumption hierarchy as described in Step 3. In fact, by making all classes siblings of a root class, then, in essence, an *unstructured* collection of terms as described in Step 1 is made. The simple knowledge organisation system[6] (SKOS) uses OWL objects and `hasBroaderThan`, `hasNarrowerThan` and `relatedTo` properties between them to represent the more informal hierarchies of Step 2.

Fig. 3 shows the definition of a complete class in OWL-DL. It can be used to describe the conditions for class membership that an individual must fulfil. These conditions come in the form of "restrictions" upon the "properties" that form the binary relationships between individuals of two classes. A restriction is so-called as it *restricts* those objects that may be members of a class. Without restrictions, any object may be a member of a class. In OWL, restrictions are *universals*: that is, they apply to *each* and *every* member of that class. The filler at the other end of the restriction can be a class expression, a class or concrete data type. It is these restrictions that can be used to form other structures than the subsumption hierarchies that form Step 3 on the feature escalator.

Class expressions can be formed using the standard Boolean operators. Restrictions can be either existentially or universally quantified. OWL-DL also offers a range of supplementary axioms such as disjunction, union and negation. Properties can be defined in terms of domain/range, subproperties, and property features (functional, inverse functional, transitive, symmetric).

The restrictions used to describe the members of a class fall into two categories:

(1) *Necessary* conditions are those restrictions that must apply to any individual of the class, but are not enough on their own to define the individual as a member of the class. For example, having a brain is a necessary condition for being

[4] http://www.w3.org.
[5] That is, it is an attribute language with complex concept negation, transitive properties, subproperties, nominals, inverse properties, cardinality restrictions and use of data types.
[6] http://www.w3.org/2004/02/skos/.

$$\text{MalateDehydrogenase} \sqsubseteq \exists \text{ catalyses } (\text{Reducing} \sqcap (\exists \text{ acts\_on Oxaloac-}$$

$$\text{etate}) \sqcap \text{Oxidising} \sqcap \ (\exists \text{ acts\_on NADPH}))$$

$$\text{MalateDehydrogenase} \sqsubseteq \exists \text{ has\_reagent\_on\_side \_B Oxaloacetate}$$

$$\text{MalateDehydrogenase} \sqsubseteq \text{enzymatic\_function}$$

$$\text{MalateDehydrogenase} \sqsubseteq \exists \text{ has\_reagent\_on\_side \_A NADPAnion}$$

$$\text{MalateDehydrogenase} \sqsubseteq \exists \text{ has\_reagent\_on\_side \_A Malate}$$

$$\text{MalateDehydrogenase} \sqsubseteq \exists \text{ catalyses } (\text{Reducing} \sqcap (\exists \text{ acts\_on NADP}) \sqcap \text{Oxi-}$$

$$\text{dising} \sqcap (\exists \text{ acts\_on Malate}) \sqcap (\exists \text{ acts\_on\_donar}$$

$$\text{\_group CH-OHGroup}))$$

$$\text{MalateDehydrogenase} \sqsubseteq \forall \text{ catalyses } ((\text{Reducing} \sqcap (\exists \text{ acts\_on NADP}) \sqcap \text{Oxi-}$$

$$\text{dising} \sqcap (\exists \text{ acts\_on Malate}) \sqcap (\exists \text{ acts\_on\_donar}$$

$$\text{\_group CH-OHGroup})) \sqcup (\text{Reducing} \sqcap (\exists \text{ acts}$$

$$\text{\_on Oxaloacetate}) \sqcap \text{Oxidising} \sqcap (\exists \text{ acts \_on NADPH})))$$

$$\text{MalateDehydrogenase} \sqsubseteq \exists \text{ has\_reagent\_on\_side \_B NADPH}$$

Fig. 3. A complex class description in OWL-DL that exemplifies a part of that which OWL-DL can express.

human, but having a brain is not enough to define an organism as human (there is no human without a brain, but there are other organisms with brains).

(2) *Necessary and sufficient* conditions are those restrictions on an individual that suffice in order to consider it a member of a class: having a very developed neocortex in the brain is enough to consider an organism as human (humans are the only organisms with a highly developed neocortex).

These sufficiency conditions, expressed in OWL-DL's strict semantics, are enough to enable automatic reasoning. DL reasoners are able to check the collection of axioms that forms an OWL-DL ontology for satisfiability. That is, the reasoner will indicate any inconsistency and infer any additional subsumption relationships implied by the axioms in the ontology. OWL-DL can, therefore, automatically infer the multiple hierarchies described in Step 4 and provide the composition and classification on demand described in Step 5.

OWL-DL is thus capable of representing anything from the simplest handcrafted tree of terms through to rich descriptions of classes in terms of restrictions upon their members. Multiple hierarchies in these ontologies can be either handcrafted or inferred by reasoning over defined classes. Consequently, OWL-DL can represent each stage in the ontology feature escalator described earlier.

Providing such formal definitions in a DL representation, as shown in Fig. 3, is a potentially costly undertaking. It is therefore not realistic to expect a community to do this style of ontology building *ab initio*, nor is it possible to make a sudden switch to this formal approach. Experience from SNOMED-RT and SNOMED-CT[7] show that it takes $26 m worth resources and over four years to make such a move. The only viable option for non-commercial community efforts is, therefore, to start with a simple representation, then slowly migrate to such an approach when and if it becomes necessary.

3. The Gene Ontology

The gene ontology[8] (GO) was developed as a resource to promote comparisons between genomic databases of different model organism species [7,15]. Consequently, pooling of results across databases is essential (see Step 1 in Section 2).

GO provides a controlled vocabulary of some 21,974 terms[9] for describing the molecular function, biological process, and cellular component (cellular location) of gene products. Therefore, descriptions of gene products (proteins and RNA) sharing common function, processes, and locations can be shared across databases enabling a *de facto* integration [7].

[7] http://www.snomed.org/.
[8] http://www.geneontology.org.
[9] January 9, 2007 at 2:00 Pacific time.

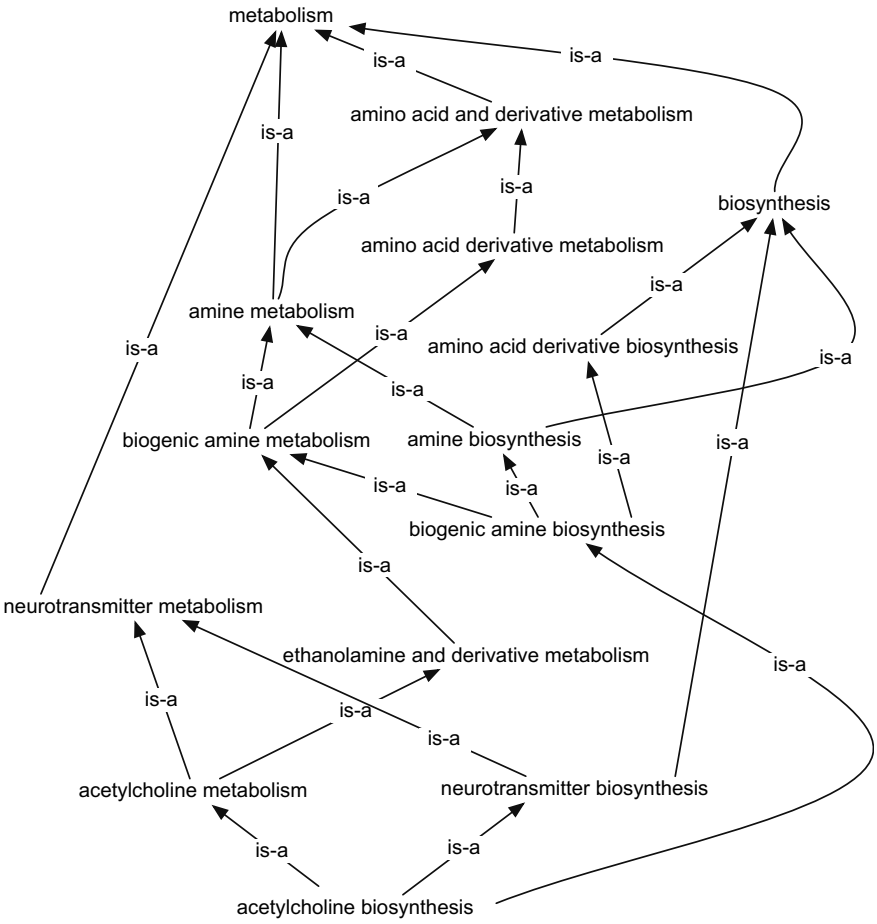

Fig. 4. A simplified fragment from the gene ontology.

Terms of GO are placed within a multiple subsumption hierarchy (see Fig. 4), represented by the GO Consortium as a Directed Acyclic Graph (DAG) in the OBO file format [16]. Terms are also placed within a multiple partonomy hierarchy, using the `part-of` relationship. GO, therefore, can be classified in Step 4 of the feature escalator, even though two types of relationships are used, because new terms can not be defined in a combinatorial manner. Terms within the ontology label nodes in the DAG and represent classes of the instances to which they refer. For example `acetylcholine biosynthesis` represents the class of instances of that process. The `is-a` and `part-of` relationships (arcs or edges in the DAG) represent semantic connections between those instances. As well as the term, each node in the DAG has a natural language definition; for example, `acetylcholine biosynthesis` is defined as "The formation from simpler components of acetylcholine, the acetic acid ester of the organic base choline". Synonyms, links to other databases and other useful information is also provided as annotations upon a term node.

GO is also used for aggregated statistics (Step 3 on the escalator). Users can ask the question "How many gene products across all model organisms have `kinase activity`?". For example, the multiple hierarchy enables proteins with protein kinase C activity to be aggregated according to either its `receptor activity` or `kinase activity`.

GO is now used by some 20 species genome databases and several community wide databases of gene products. It is a handcrafted ontology; each term is placed within the `is-a` and `part-of` hierarchy according to the curators informal criteria [15]. The GO has been widely adopted and accepted by the research community [17]. Like any ontology, however, it will contain errors that fall into two broad categories:

(1) Biological validity—does the ontology, using the expressivity of its KR language, capture the biological nature of the world with high-fidelity?
(2) Structural validity—are all the terms in the most appropriate location with all the subsumption relationships implied by their definitions?

These two categories obviously overlap. It is the second category, however, that can be addressed by Steps 3 and 4 in the feature escalator. The Gene Ontology Next Generation (GONG) methodology principally addresses the structural validity of

an ontology and uses the migration towards a rich OWL-DL ontology in order to fix these drawbacks. Structural defects will generate biological defects, but the biological defects in their own right are a deeper conceptual malaise than a misplacement of a term—the biology itself is incorrect.

4. Methodology

4.1. Untangling

With complex conceptualisations, it is very easy to get into a "tangle". A complex concept has different aspects all of which can be used to create a subsumption relationship. Fig. 4 shows such a tangle in GO originating from the term `acetylcholine biosynthesis`. Such a tangle has resulted from a concept which has only two key aspects: (i) the nature of the biological process—`biosynthesis` (ii) the nature of the chemical substance on which the process acts—`acetylcholine`.

Rector et al. [18] have proposed a methodology for untangling such taxonomies, and so easing their maintenance. The stages in this "normalisation" are:

(1) Choose only one aspect with which to manually create subsumption relationships. In the tangle above this could be the nature of the biological process.
(2) Re-express most of the other aspects of the class, not as subsumption relationships, but as restrictions on each concept. For `acetylcholine biosynthesis`, this would involve expressing the restriction `acts_on acetylcholine` and adding `biosynthesis` as a superclass. Likewise for its original parent `neurotransmitter biosynthesis`, add `acts_on neurotransmitter` as an OWL restriction and add `biosynthesis` as a superclass. So from expressing all the possible relationships in a hard-coding fashion in the GO style, we only express the conditions for class membership: from saying that "`acetylcholine biosynthesis` is a subclass of `neurotransmitter biosynthesis`, `acetylcholine biosynthesis` and `biogenic amine biosynthesis`"[10] to simply saying, in OWL, that "`acetylcholine biosynthesis` is a kind of `biosynthesis` that acts on `acetylcholine`".
(3) The previous step does not provide sufficient information to recreate the original subclass link to `neurotransmitter biosynthesis`. A separate chemical taxonomy must be created to explicitly state that `acetylcholine` is a subclass of `neurotransmitter`.
(4) Submit both ontologies to a DL reasoner such as FaCT++[11] to infer subsumption relationships and check that all property-based definitions are logically consistent.
(5) Report back on satisfiability, including any inferred subsumption relationships found.

Fig. 5 shows how these various steps can untangle such an example.

4.2. One-off or incremental change?

One approach taken by the GALEN project [19] to this untangling is to use the existing terminologies as source information from which to formulate wholly new taxonomies and property-based definitions [20]. Reasoning is then used to construct an overall taxonomy that is based on these new logic definitions.

There are limitations to this approach:

(1) Building taxonomies and property-based definitions for all of an ontology is a large knowledge elicitation task.
(2) It may not be possible to recreate an ontology that resembles the original taxonomy with which users are familiar.

We cannot provide complete definitions for all concepts within the GO in a one-off effort. In fact, many believe that the task of writing definitions is an ongoing one in which definitions are amended constantly to reflect response to feedback from users. This is the approach that the GO Consortium itself takes [17]. They released the ontology early in its development and it subsequently underwent a rapid evolution led by user feedback. Indeed, the GO is a constantly changing artefact; as new organism databases join the consortium, new requirements expand the ontology. Additionally, as our understanding of the world of biology changes so will the GO. The GONG project has developed a similar evolutionary methodology, but within the DL environment [8].

In the GONG methodology, small groups of concepts are defined in terms of OWL-DL and then the ontology is submitted for DL reasoning. The reasoner infers new subsumption relationships and flags any logical inconsistencies that can be fed back to the GO editorial team. The benefits of providing definitions can be shown in small increments and the whole approach can be validated at an early stage in the project. The GONG project, therefore, took the following approach:

[10] And possibly many more relationships, as we are only using a simplified sub-set for the sake of clarity of the example.
[11] http://owl.man.ac.uk/factplusplus/.

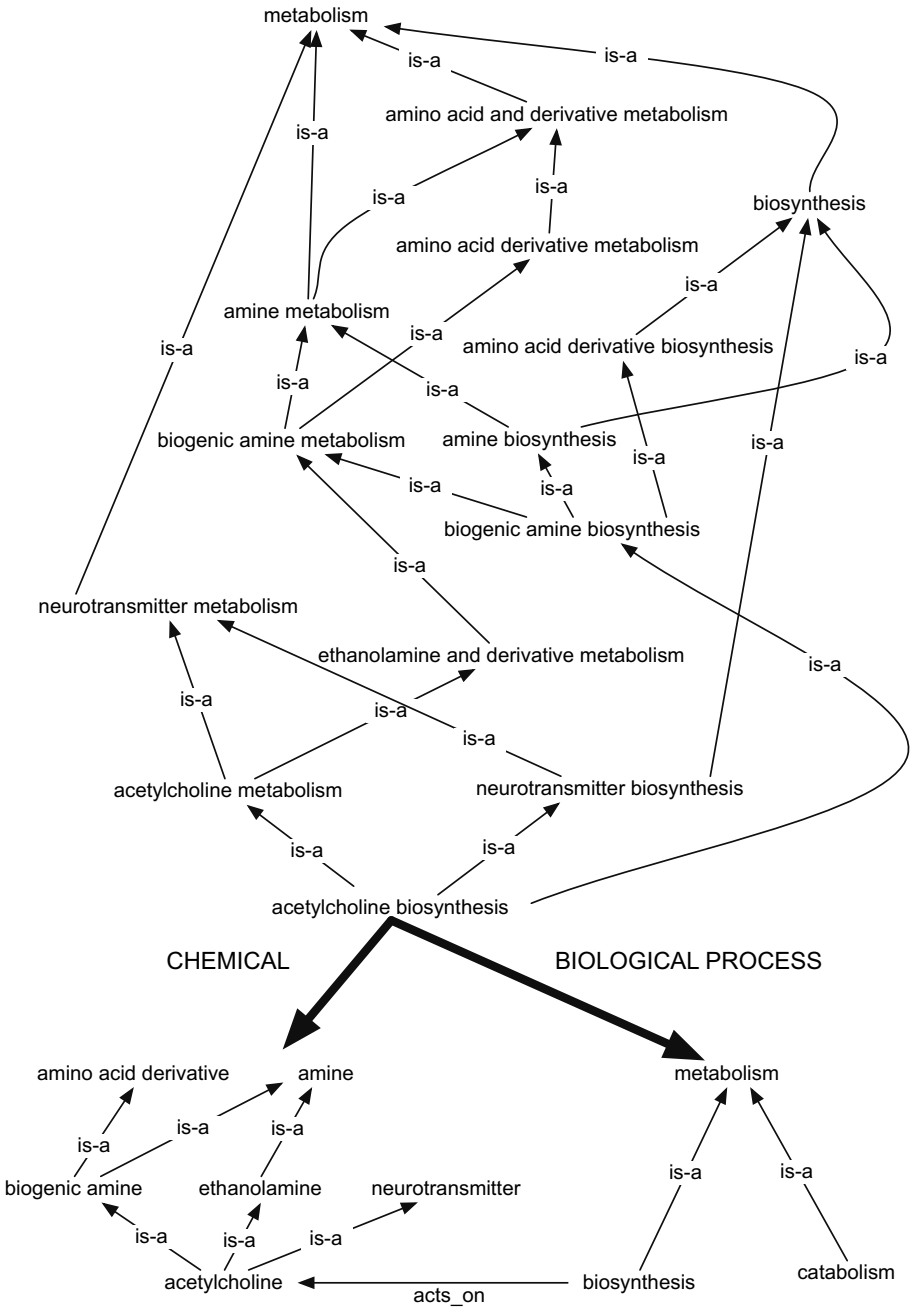

Fig. 5. Untangling of the term `acetylcholine biosynthesis`.

(1) Conversion from source encoding to OWL-DL.
(2) Small areas untangled in situ.
(3) Descriptions of classes derived, either complete or partial:
    (a) Automatically, from dissection of GO term names;
    (b) By hand.

(4) Definitions may not be complete; that is, with necessary and sufficient conditions.
(5) New information added to the ontology through reasoning over descriptions. No existing information is removed a priori. Only additional semantic information and subsumption relationships added via reasoning over descriptions. Inconsistencies created during this migration are flagged by the reasoning process.
(6) Changes reported back to the source ontology that remains in its native form.

To mitigate Step 3 it is important to harness as many internal and external sources of pre-existing knowledge in order to form the richer concept definitions and associated taxonomies (the building blocks for the normalisation approach discussed earlier). These may include the class or node label (the term), text definition and external database information linked to the concepts.

Moving a GO with around 22,000 terms in a one-off fashion, with a complete change to a formal representation with complete formal definitions is a significant knowledge elicitation task that is unsustainable by the community. Therefore, small areas of GO must be addressed, in an incremental fashion, at any one time.

Deciding when it is necessary to migrate to a more formal representation is a question that has not received much attention. From experience in the GONG project and others, several triggers can be identified that indicate that such a move is advisable. Size in terms of number of concepts is itself not a justification to move. If the large number of concepts are present in a simple tree structure, then moving to a formal environment will not provide great benefits in maintaining that structure. If the structure is, however, complex, in which concepts have more than one parent then experience shows that the taxonomic structure becomes difficult to maintain by hand.

The difficulty involved in maintaining the structure appears to be a function of both the number of concepts in the ontology and the average number of subsumption relationships. A small ontology with only hundreds of concepts can become impossible to maintain if it has an average number of superclasses of six [21].

Although the initial ontology may be simple, it must be sufficiently well formed to enable a migration to a formal representation. That is, it must be able to be transformed into OWL-DL. Heuristics to follow are: (i) distinguish between classes and terms; (ii) provide a unique identifier for each concept that is independent of its human readable name; (iii) make the nature of all relationships explicit and try to limit parent–child relationships to only be subsumption relationships.

Keeping in step with mainstream development of the source ontology is essential to this approach. This requires that the migration only adds to the existing structure and does not remove anything. Change to the original structure can be a recommendation from the inferences made over the migrated ontology. This is an amendment to the normalisation methodology in which the existing ontology is completely deconstructed and re-built using terminological reasoning. Any changes made by the mainstream development team must be fed into the more formal representation. Eventually, the OWL-DL version may become the default version, along with a suite of migrations for all, some or most areas of the original ontology.

5. Using the Gene Ontology Next Generation (GONG) methodology

5.1. Converting DAG to OWL-DL

The first stage in migrating towards an OWL-DL, property-based form for GO is its conversion from DAG into an OWL-DL representation. This is simply a change of representation. What has been stated in the original GO is also stated in the OWL-DL version of GO, including any biological mistakes.

There is no generic solution to the step of converting non-OWL-DL representations. Conversions into OWL-DL have to be developed for each representation. If the meaning of statements in the source encoding cannot be represented in OWL-DL, then the GONG process cannot commence. One advantage, however, of OWL-DL is its ability to represent a wide range of conceptualisations from simple taxonomies, to complex graphs with many kinds of relationships.

5.2. Sub-setting the ontology: attacking small areas

There are several motivations for a piecemeal, incremental migration of small areas of a large ontology such as the GO. It is simpler to keep track of progress if defined topic areas are identified and addressed in turn. Working with a sub-set reduces the time it takes to iterate through the process and feed enhancements back to the mainstream community.

The reduction in time is as a result of both the reduced scope and amount of knowledge acquisition and also the reduced computational time taken by reasoning software. At the moment reasoning time rises exponentially as a function of both ontology size and the complexity of formal definitions. Keeping to small sub-sets ensures reasoning time does not become a limiting factor in the process.

Most ontologies vary widely in the style of their conceptualisation. Some may be straight forward to migrate to a formal representation with clear formal definitions, others may require detailed discussion about possible definitions which themselves become complex. Sub-setting the ontology allows the early stages of the migration to focus on those areas that provide the best return on effort. All these considerations are in line with the avoidance of the one-off re-formulation of the normalisation of the ontology.

5.2.1. Implementing a sub-setting step

The simplest manner to describe a sub-set in a hierarchical structure is to identify a concept in the hierarchy and then include in the sub-set all descendants of that concept. This is, however, not sufficient and forms only the first step of the sub-setting procedure. To ensure that all information about each sub-set concept is included, it is also essential to include all the ancestors of the descendant concepts (they may have multiple parents that fall outside the sub-set). It is also essential

to include all concepts linked to concepts in the sub-set by non-taxonomic relationships (e.g. `part_of`), and in turn their ancestors.

Whilst it may be thought that this would result in including most of the original ontology, in fact experience with GO has shown that links outside the sub-set hierarchy are sparse. This simple method works for the lean ontologies from which we are migrating. More sophisticated techniques do exist for richer ontologies [22,23].

5.3. Automated dissection where possible

Producing formal definitions for classes by hand has several disadvantages. Firstly, it requires significant time from an individual trained in writing formal concept definitions to commit to a consistent pattern. Secondly, any such process is prone to human error. Therefore, the methodology aims to automate mundane time consuming tasks and moves the human effort from definition production to definition checking and augmentation.

5.3.1. Mining class labels

Many class labels of ontologies in the life sciences are actually phrases whose structure and lexical content can provide considerable information about the definition of a concept [8,24,25]. What is more, the structure of the phrase may conform to a pattern which is repeated for many concepts in a given area of an ontology. For example, GO includes a large section detailing metabolism concepts. The majority of concept labels, the terms, in this section follow the pattern "⟨x⟩ metabolism" where x is a class of chemical e.g. `carbohydrate metabolism`. Within the GONG methodology we formally specify how this rubric phrase pattern translates into a formal concept definition pattern. Obviously these patterns are often local to an area of a specific ontology; increasing the need to migrate sub-sets of the ontology. A domain expert must be involved in developing the formal definition pattern and its mapping, because the rubric does not provide a complete definition for the concept.

The resulting formal definition either may or may not capture a complete definition for a concept. In the example above, for the concept `carbohydrate metabolism`, a definition such as "a subclass of metabolism which acts solely on carbohydrate" can be generated from the rubric and considered complete; it is sufficient to recognise instances of that kind of metabolism.

5.3.2. Implementing rubric mining

The rubric pattern is identified using a regular expression. So for example the pattern described earlier "⟨x⟩ metabolism" would be specified using the regular expression "(.+?) (metabolism$)". This regular expression is divided into two groups specified by the brackets. The first group specifies the "⟨x⟩" matching any words preceding the word "metabolism". The second group matches the word "metabolism" occurring at the end of the rubric.

This approach is not just applicable to GO. SNOMED-CT also contains a large number of rubrics that follow a repetitive pattern [26]; for example, surgical procedures have the structure "excision of ⟨x⟩". We use simple regular expressions; it would also be reasonable to write a lexer and a grammar for many of the ontologies in the life sciences as they are systematic in their naming styles. The approach of using a grammar is taken by Mungall [25].

5.3.3. Mining free-text definitions

The degree to which free-text definitions can be mined depends on the degree to which it really is free-text and the sophistication of natural language processing brought to bare on the task. Many GO definitions intentionally follow a very constrained and stereotyped pattern; for example many concepts have a definition of the form "Catalysis of the reaction: ⟨substrates⟩ = ⟨products⟩". These can be utilised in a similar manner to the concept rubrics to yield formal concept definitions. GO is also migrating towards a form of Aristotelean definition of genera, species and differentiae that are direct natural language counterparts of the computational form we desire. Translation from this form would be relatively easy.

5.4. Hand annotation when necessary

There is a limit to how much of an individual definition or what proportion of an ontology's classes can be automatically constructed from existing electronic resources. Also the automatically constructed definitions must be checked. Therefore, there is a process of hand curation, to check and augment these definitions.

5.4.1. Utilising existing taxonomies and mapping to them

As property-based descriptions are made for classes in the target ontology, supporting ontologies are needed to supply fillers for those properties. Therefore, the OWL-DL definitions of the concepts consist of restrictions, both partial (necessary) and complete (necessary and sufficient), that point to classes of the supporting ontologies.

In the process of normalisation, the untangled ontologies are these supporting ontologies. Sometimes these will be created de novo, often in the form of value partitions.[12] For instance, many GO biological processes involve both `positive` and

[12] http://odps.sourceforge.net/odp/html/Value_Partition.html.

negative regulation. A value partition of negative and positive is formed and linked to the GO class description via a has_regulation property. Often, much larger scale ontologies are needed to drive the untangling. For example, metabolism, transport, binding, enzyme activity, etc., all involve chemicals. An ontology of chemicals was extracted from MeSH for this purpose. The recent release of the chemical entities of biological interest (ChEBI) [27] has replaced this temporary measure. Similarly, all development classes need an anatomy ontology; behaviour processes need phenotype ontologies; cell differentiation classes need a cell types ontology, etc.

The GONG process is somewhat dependent on the existence of such supporting ontologies. This is another reason for the piecemeal or in situ migration of the DAG form to an OWL-DL formalism. The Open Biomedical Ontologies (OBO) project[13] offers a wide and growing range of such ontologies that themselves have to be converted to OWL-DL for the GONG process.

It should be remembered, however, that even a partial migration using the de novo smaller supporting ontologies such as the regulation value partition mentioned above, can offer structural validation. Such partial untangling still automatically classifies along one axis and will structurally validate that axis of the ontology being processed.

5.5. Reporting changes

As each sub-set of the GO is reasoned over and the resulting changes collated, it must be fed back to the owners of the source ontology (the GO curators). The sub-setting of the ontology into "topics" has another effect at this point: by restricting changes to a coherent, topic based sub-set, changes can be fed back to the source ontology in a coherent manner.

The changes are sent to GO curators for reviewing and possible inclusion in GO. This is done using the SourceForge GO Curator Requests Tracker,[14] sending the requests in packs of 25 new relationships or suggestions, giving the curators enough time to decide whether or not to include the changes. As the source ontology changes, the process can be repeated, improving the source ontology as it evolves.

5.6. Implementation

The GONG workflow can be executed using a program called BONG[15] (Biological Ontology Next Generation). BONG is able to execute the GONG workflow using any OBO ontologies. The process is as follows:

(1) The user configures the settings of the workflow using a special OWL ontology, the GONG ontology. The regular expressions for dissecting the terms and the new semantics attached to the regular expressions are added to the ontology by the user; documentation and an OWL template to create a customized GONG ontology are provided with the program. The new semantics attached to each regular expression can be of arbitrary complexity and new classes can be included to complement the process.
(2) If the OBO ontologies are not already in OWL, they can be converted to OWL using another program available in the same web (OBO2OWL).
(3) If all the necessary files (the OBO ontologies in OWL format and the GONG ontology) are in place, the workflow can be executed by calling the BONG program; the program reads the GONG ontology and performs the workflow according to the settings, using the OBO ontologies as input. In the workflow, the terms of the chosen ontology are dissected according to the regular expressions and new semantics added to them in the form of OWL-DL constructs.
(4) Once the dissection has finished, the new dissected ontology with new semantics is sent to a DIG[16] interface compliant reasoner. The reasoner infers new relationships that are introduced in a Hypersonic database[17] for permanent storage and efficient retrieval. The new ontology with the new relationships is stored on disk.

6. Results

Having described our migration process in general and its specific application to the GO, in this section we report upon the results of the GONG process in several topic areas of the GO. We first describe the capture of patterns of lexical forms in GO terms for the generation of OWL-DL descriptions as detailed in Section 4. We then describe the results of reasoning over these descriptions, from metabolism, transporter activity and binding sub-sets within the GO. Finally, we look at the results for the new descriptions of GO terms generated by the workflow.

6.1. Results of regular expressions

Table 1 shows the regular expressions used to exploit the patterns of lexical usage in GO terms. Once a pattern is matched, an OWL-DL description for the class can be generated from a template, to which are added the salient parts of the GO term.

[13] http://obofoundry.org/.
[14] http://sourceforge.net/projects/geneontology/.
[15] http://www.gong.manchester.ac.uk.
[16] http://dig.sourceforge.net/.
[17] http://hsqldb.org.

Table 1
Examples of regular expressions used to match against the terms of the `binding`, `transporter activity` and `metabolism` sub-sets of GO and generate the OWL-DL descriptions for each GO class

Regular expression	Captured example
Binding sub-set	
(.+?) (binding$)	Glutamate binding
(.+?) ([a–z] + er + \s + activity$)	ISG15 carrier activity
Transporter activity sub-set	
(.+?) (channel + \s + activity$)	Calcium channel activity
(.+?)(.+?) (antiporter + \s + activity$)	Acetylcholinehydrogen antiporter activity
Metabolism sub-set	
(.+?) ([a–z] + ism $|[a–z] + ing $|[a–z] + tion $\|[a–z] + sis$|[a–z] + age$)	Neurotransmitter biosynthesis
(.+?) ([a–z] + ism|[a–z] + tion|[a–z] + sis|linkage|[a–z\−] + ing)[\,] + ?via (.+)	L-Alanine biosynthesis via ornithine

Table 2
Result percentages for `binding`, `transporter activity` and `metabolism` sub-sets

	Captured	Changed	Accepted by author	Accepted by GO curators
Binding (%)	98	17	8	5
Transporter activity (%)	94	21	11	8
Metabolism (%)	90	20	8	–

The percentages express the relative number of terms (classes in OWL-DL).

Tables 1 and 2 show that a relatively small number of regular expressions can capture a large proportion of the terms within a particular sub-set.

When capturing the terms with the regular expressions, the semantic content that would be accessible to the reasoner was greatly determined by the design of the regular expressions. In general the approach worked because, as mentioned before, the GO terms are syntactically highly stereotypic, with a high occurrence of strings (subterms) and substrings in a regular manner. There were, however, two kinds of problems:

- Some terms were not captured at all (2% in the case of the `binding` sub-set) that means GO terms are not completely syntactically stereotyped. Examples include: `MHC class II protein binding`, `via lateral surface` and `translation release factor activity`, `codon non-specific`, amongst others.
- Even when captured, in some cases the syntactic subtleties of the term were not translated to new semantics in the final ontology. For example, the term `RNA polymerase II transcription factor activity, enhancer binding` was split in the subterms "RNA polymerase II transcription factor activity, enhancer" (chemical axis) and "binding" (functional axis). The first subterm did not match any class in the MeSH, due to its complexity. On the other hand, there were subterms that could be used taking advantage of OWL-DL's expressivity. Subterms like "enhancer" and "transcription factor" could be captured using alternative regular expressions and mappings to more complex OWL-DL class descriptions.

There is a cost-benefit analysis in writing regular expressions—How many terms or what proportion of terms being captured make it worthwhile to form regular expressions? When only either two or three would be captured, we have resorted to a manual process of mapping. Finally, the fact that some terms have not adhered to the GO house style is a result in itself that is worth reporting to the GO curators. It also means that we do not form a regular expression for these "outliers".

6.2. Example transformations to OWL-DL

The GO term `alanine:sodium symporter activity` is mapped to the class description in Fig. 6. (The class description is provided using Manchester OWL Syntax [28]) The pattern is "⟨chemical⟩ ⟨chemical out⟩ ⟨transport function⟩". There are two existential restrictions via `acts_on` to the "⟨chemical in⟩" and the "⟨chemical out⟩" being transported. These chemicals are classes in the supporting chemical ontology from MeSH.

Finally, this class is made a subclass of `symporter activity` from the `transporter activity` ontology in Fig. 7, that maps from "⟨transport function⟩" in the GO term. The `transporter activity` ontology is created by hand.

6.3. Results of reasoning

The percentages of terms captured in each sub-set and the percentage of classes having new subsumption relationships inferred in each sub-set can be seen in Table 2. For example, in the case of `binding` 771 terms from the original 789 terms were captured applying the regular expressions (98%). From the original terms, 17% were changed (another relationship added, moved in the hierarchy, etc.). All those new changes were reviewed by one of the authors (Egaña Aranguren) to check

```
Class: alanine:sodium symporter activity

EquivalentTo:
    symporter activity
    that transports only (alanine or sodium)
```

Fig. 6. Definition of the term `alanine:sodium symporter activity`.

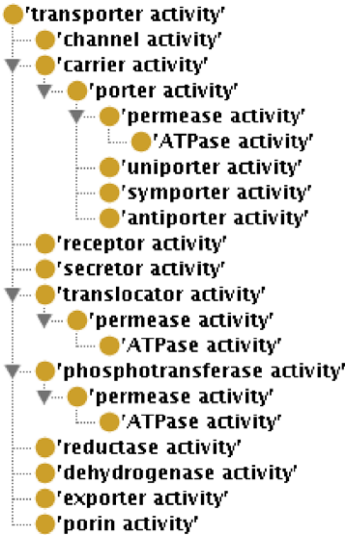

Fig. 7. The `transporter activity` ontology used in mapping the `transporter activity` sub-set in GO.

the biological validity, and 8% of the original terms were accepted. Those were sent to the GO curators and 5% of the original terms had new relationships accepted for inclusion in GO. The metabolism results were sent for reviewing by GO curators as part of the work presented in [29], but they were not reviewed as too many requests were made through the request tracker.

Table 2 shows the tangible results of the GONG process. With very few patterns a great proportion of GO terms were captured in a sub-set. We also see a large number of subsumption relationships inferred from the OWL-DL class descriptions generated via these regular expression matches. We see in Table 2 that many of these new relationships were accepted by the GO curators, giving a strong external validation of the process. The less tangible benefit of the process are in the migration to a strict formalism itself. This strictness, combined with the richer descriptions will facilitate querying (as will the more complete structure) and computational processing. That is, the benefits of moving up the feature escalator. Instead of only asking for `glucose metabolism` or `glucose transporter`, a biologist using an OWL-DL version of a GONGed GO would be able to ask for all biological processes that `acts_on glucose`. In general, the more axioms present in the ontology, especially those forming restrictions, the more questions or inferences may be made over the ontology. Obviously, more sophisticated class expressions than this are possible and consequently, it is possible to dramatically increase the computational possibilities of such knowledge from rich on meaning, but axiomatically lean ontologies.

These details from the results illustrate the range of reasons for the inference of new subsumption relationships. The results of the reasoning over OWL-DL definitions lead to the following categories of results:

- Discovery of missing subsumption relationship;
- Moving a term to a more specific parent;
- Stimulating the addition of a new, more specific, parental class to GO;
- Feedback from domain experts prompts change in supporting auxiliary (and simply wrong) ontology;
- Contradiction—e.g. GO had a `biosynthesis` as a kind of `catabolism`, but disjointness highlights this state.

These categories captured all the kinds of changes inferred by the reasoner. It should be noted that very few of the changes given to the GO curators were wasted. In fact, few of the changes inferred by the reasoner were wasted. Some merely led to changes in the supporting ontologies, but these were useful in reducing errors in another round of reasoning. Even changes ultimately rejected by GO curators that were not *wrong*, but only incorrect in GO's view of the world were useful in prompting discussion.

Some of the new subsumption relationships from the `binding` sub-set accepted by the GO staff can be seen in Table 3. There were two main reasons why a new subsumption relationship inferred by the reasoner was not accepted by the GO curators:

Table 3
Some of the new subsumption relationships of the `binding` sub-set accepted by the GO staff

New IsA link
Glycine binding (GO:0016594) IsA neurotransmitter binding (GO:0042165)
Epidermal growth factor binding (GO:0048408) IsA hormone binding (GO:0042562)
FAD binding (GO:0050660) IsA adenyl nucleotide binding (GO:0030554)
New position
Glycosaminoglican binding (GO:0005539) should be under polysaccharide binding (GO:0001871) instead of binding (GO:0005488)
Hemoglobin binding (GO:0030492) should be under protein binding (GO:0005515) instead of binding (GO:0005488)
ISG15 carrier activity (GO:0019793) should be under protein carrier activity (GO:0008320) instead of protein binding (GO:0005515)

"New position" means that one `is-a` link should be deleted and another one added. "New IsA link" means that another `is-a` link should be added in the following manner: class `is-a` superclass.

- The relationship was already there: this in fact demonstrates the accuracy of the GONG workflow, because most of the curators reviewed the new relationships against a newer GO version than the version used for the GONG process. Thus, the GONG workflow added the same new relationships as the human curators did.
- Lack of biological accuracy: this is due to the fact that the MeSH ontology was used for the chemical taxonomy. For example, the suggestion of adding `high affinity ammonium transporter activity` as a subclass of `organic cation porter activity` was rejected because ammonium is not organic.

7. Discussion

Even if the execution of the GONG workflow described herein did not have a 100% performance, the GONG approach can be seen to have worked and demonstrated that the migration to a DL environment can have benefits for the structural validation of an ontology such as GO. New relationships were accepted by the GO team and possible application areas for the axiomatically enriched OWL-DL ontology were suggested. By the application of a simple dissecting procedure, even without full coverage and a minimum of programming and reviewing work, the taxonomic structure of GO improved substantially. This demonstrates the usefulness of the GONG process. Other factors must, however, be taken into consideration: GONG is an automated procedure but not all the semantics will always be fully captured, so human intervention is needed. Human intervention introduces errors and is time consuming and expensive.

The GONG process allows a migration from a simple representation to a more expressive formalism. It moves from an axiomatically *lean* to an axiomatically *rich* form of an ontology. Ontologies such as GO have a wealth of semantics captured in elaborate term names. These terms work well for the purposes of a controlled vocabulary, but the semantics in the term names are not computationally available. The GONG methodology draws the semantics out of the term names and makes them explicit and computationally available in the form of extra axioms in the ontology. GONG is still very much a centralised process with an ontology expert performing the migration. It can, however, be coupled with a community ontology building process. Diligent (DIstributed, Loosely-controlled evolvInG engineering of oNTologies) [30] is a methodology which aims to allow more local adaptation by community groups, followed by a process of centralised harmonisation. Much of the methodology is representation neutral. It is unlikely that biologists, who are not necessarily OWL experts, will directly write OWL. Such experts will, however, readily supply terminology in a systematic manner and therefore, the GONG process could be applied to such contributions as part of a tooling for the Diligent methodology.

The ontology maintainer can stop at any position on the feature escalator that is the optimal stage of balance between expressivity and ease of modelling. It is unrealistic to require large-scale involvement of the existing development team and users during the migration. Therefore, especially in the early stages, the migration must be decoupled from day to day use, whilst improvements must be fed back into the released version. This ensures that the community can see the benefits of the migration, even within the early stages. As the process advances the community will become more confident in working with the more expressive steps of the escalator (as they can always come back to the original stage) and finally, choose the step that suits them in terms of expressivity. Therefore, they will make that step of the escalator the default version of the bio-ontology, thus probably fully exchanging to OWL-DL.

The process we describe is piecemeal. Eventually, we would expect to see a fully normalised GO. The taxonomies pulled out for `metabolism`, `transporter activity`, `binding`, etc. would eventually coalesce to form a taxonomy for molecular function, biological process and cellular component that are not conflated with any other forms of classification. A molecular function taxonomy covering catalysis, transport, binding, regulation of proteins and processes would be supported by:

- Small molecules, atoms and ions;
- Macromolecules;
- Reactions;
- Biological processes;
- Cellular components.

In this eventuality, the currently separate ontologies of the Open Biomedical Ontologies project would interoperate more fully; allow richer queries; allow greater computational usefulness and more re-use. As each ontology is a tree, some calculations become easier. For example, calculating semantic similarity is confounded by multiple inheritance [31]. Instead of most of the biological knowledge being present in the terms it would be available for computational use; especially in making queries. In addition, full normalisation would give a series of smaller, mono-axial ontologies that would be easier to both maintain and extend.

This migration could also be useful outside the life sciences. Such a migratory approach will be important in the development and delivery of the Semantic Web.[18] The Semantic Web relies on the semantic description of Web content and services, possibly via ontologies. Languages such as OWL, even in its OWL-Lite variant, are relatively complex and it is unlikely that domain practitioners will use the full power of these languages. Yet it is these domain users that will have to make ontologies for any Semantic Web. The GONG methodology can be applied to any structured vocabulary whose representation can reasonably be transformed into OWL. Its success is based in dissection of patterns in class labels that can make the semantics of those terms explicit. Where a systematic naming convention is lacking GONG will be less tractable. GONG's usefulness needs not be limited to producing OWL. Semantics hidden inside string literals of a resource description framework (RDF) graph could be made explicit in the form of extra triples. The GONG methodology offers a route by which such practitioners can start with a simple representation and progressively migrate to a richer ontological form as required.

Acknowledgements

We would like to thank Michael Ashburner, Amelia Ireland, Jane Lomax and Chris Mungall, of the GO consortium, without whose help this work would not have been possible. We would also like to thank Olivier Dameron for technical help. Chris Wroe was supported by the GONG project grant (DARPA DAML subcontract PY-1149 from Stanford University) and the myGrid eScience pilot project grant (EPSRC GR/R67743).

References

[1] M. Dean, D. Connolly, F. van Harmelen, J. Hendler, I. Horrocks, D. McGuinness, P. Patel-Schneider, L. Stein, OWL Web Ontology Language 1.0 Reference, February 2004. <http://www.w3.org/TR/owl-ref/>.
[2] M. Uschold, M. Gruninger, Ontologies: principles, methods and applications, Knowledge Engineering Review 1 (2) (1996) 93–155.
[3] T.R. Gruber, A translation approach to portable ontologies, Knowledge Acquisition 5 (2) (1993) 199–220.
[4] G. Ringland, D. Duce, Approaches to Knowledge Representation: An Introduction, Knowledge-Based and Expert Systems Series, John Wiley, Chichester, 1988.
[5] M. Uschold, R. Jasper, A framework for understanding and classifying ontology applications, in: KRR5-99, 1999.
[6] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, P.F. Patel-Schneider (Eds.), The Description Logic Handbook: Theory, Implementation, and Applications, Cambridge University Press, 2003.
[7] The Gene Ontology Consortium, Gene ontology: tool for the unification of biology, Nature Genetics 25 (2000) 25–29.
[8] C. Wroe, R. Stevens, C. Goble, M. Ashburner, A methodology to migrate the gene ontology to a description logic environment using DAML + OIL, in: Eighth Pacific Symposium on Biocomputing (PSB), 2003, pp. 624–636.
[9] C. Welty, M. Gruninger, F. Lehmann, D. McGuinness, M. Uschold, Ontologies: expert systems all over again? in: AAAI-1999 Invited Panel, 1999.
[10] S. Nelson, W. Johnston, B. Humphreys, Relationships in medical subject headings, Relationships in the Organization of Knowledge, Kluwer Academic Publishers, 2001.
[11] J. Rogers, W. Solomon, A. Rector, P. Pole, P. Zanstra, E. van der Haring, Rubrics to dissections to grail to classifications, in: Medical Informatics Europe, vol. 43, 1997, pp. 241–245.
[12] R. Stevens, C. Goble, I. Horrocks, S. Bechhofer, OILing the way to machine understandable bioinformatics resources, IEEE Transactions on Information Technology and Biomedicine 6 (2002) 129–134.
[13] R. Stevens, C. Goble, I. Horrocks, S. Bechhofer, Building a bioinformatics ontology using OIL, IEEE Transactions on Information Technology and Biomedicine 6 (2) (2002) 135–141.
[14] T. Berners-Lee, M. Fischetti, Weaving the Web: the original design and ultimate destiny of the World Wide Web by its inventor, Harper San Francisco, 1999.
[15] Gene Ontology Consortium, The Gene Ontology (GO) database and informatics resource, Nucleic Acids Research 32 (2004) D258–D261.
[16] M. Egaña Aranguren, S. Bechhofer, P. Lord, U. Sattler, R. Stevens, Understanding and using the meaning of statements in a bio-ontology: recasting the gene ontology in OWL, BMC Bioinformatics 8 (2007) 57.
[17] M. Bada, R. Stevens, C. Goble, Y. Gil, M. Ashburner, J. Blake, J. Cherry, M. Harris, S. Lewis, A short study on the success of the gene ontology, Web semantics: science, Services and Agents on the World Wide Web 1 (2) (2004) 235–240.
[18] A. Rector, C. Wroe, J. Rogers, A. Roberts, Untangling taxonomies and relationships: personal and practical problems in loosely coupled development of large ontologies, in: K-CAP 2001, 2001, pp. 139–146.
[19] A. Rector, J. Rogers, Ontological and practical issues in using a description logic to represent medical concept systems: experience from GALEN, in: Reasoning Web, 2006, pp. 197–231.
[20] A. Rector, P. Zanstra, W. Solomon, J. Rogers, R. Baud, W. Ceusters, Reconciling users' needs and formal requirements: issues in developing a re-usable ontology for medicine, IEEE Transactions on Information Technology and Biomedicine 2 (4) (1999) 229–242.
[21] C. Wroe, J. Cimino, A. Rector, Integrating existing drug formulation terminologies into an HL7 standard classification using OpenGALEN, in: Annual Fall Symposium of American Medical Informatics Association, Washington, DC, 2001.
[22] J. Seidenberg, A. Rector, Web ontology segmentation: analysis, classification and use, in: Proceedings of the 15th International World Wide Web Conference, 2006.
[23] B.C. Grau, Y. Kazakov, I. Horrocks, U. Sattler, A logical framework for modular integration of ontologies, in: Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007), 2007, pp. 298–303.
[24] P. Ogren, K. Cohen, G. Acquaah-Mensah, L.H.J. Eberlein, The compositional structure of gene ontology terms, in: Pacific Symposium on Biocomputing, vol. 9, 2004, pp. 214–225.

[18] http://www.w3.org/2001/sw/.

[25] C. Mungall, OBOL: integrating language and meaning in bio-ontologies, Comparative and Functional Genomics 5 (6) (2004) 509–520.
[26] S. Schulz, S. Hanser, U. Hahn, J. Rogers, The semantics of procedures and diseases in SNOMED CT, Methods of Information in Medicine 45 (4) (2006) 354–358.
[27] K. Degtyarenko, P. de Matos, M. Ennis, J. Hastings, M. Zbinden, A. McNaught, R. Alcántara, M. Darsow, M. Guedj, M. Ashburner, ChEBI: a database and ontology for chemical entities of biological interest, Nucleic Acids Research 36 (2008) D344–D350.
[28] M. Horridge, N. Drummond, N. Goodwin, A. Rector, R. Stevens, H. Wang, The Manchester OWL syntax, in: OWL: Experiences and Directions 2006 Athens, Georgia, USA, November 10–11, 2006.
[29] M. Egaña Aranguren, Improving the structure of the gene ontology, MSc thesis, 2004. <http://www.gong.manchester.ac.uk/doc/MSc_thesis.pdf>.
[30] S. Pinto, S. Staab, C. Tempich, DILIGENT: towards a fine-grained methodology for Distributed Loosely-controlled and evolvInG Engineering of oNTologies, in: ECAI, 2004.
[31] P.W. Lord, R. Stevens, A. Brass, C.A. Goble, Investigating semantic similarity measures across the gene ontology: the relationship between sequence and annotation, Bioinformatics 19 (10) (2003) 1275–1283.

Mikel Egaña Aranguren is currently completing his PhD in the BioHealth Informatics Group in the University of Manchester's School of Computer Science. He obtained an MSc in bioinformatics in the same institution, and he has a degree in biology by the university of Basque Country. His research focus is on knowledge representation and bioinformatics, especially techniques such as the use of Ontology Design Patterns to rigorously capture biological knowledge in rich, but usable ontologies. He is maintaining an Ontology Design Patterns catalog (http://odps.sourceforge.net) and he is participating in the development of the Cell Cycle Ontology (http://www.cellcycleontology.org).

Dr. Chris Wroe currently works for BT Health on the London Local Service Provider Programme, part of the UK National Health Service National Programme for IT. He is a member of the system design team specifically working on the consistent use of vocabularies such as SNOMED-CT to aid information integration. He started out as a junior doctor, but then spent 7 years in the Medical Informatics and Information Management groups at Manchester University. He worked there as a research associate focussing on the design and development of bio-ontologies in the GALEN, Drug Ontology, Gene Ontology Next Generation and myGrid projects.

Carole Goble is a full Professor in the School of Computer Science in the University of Manchester. Her research interests are on distributed knowledge based information systems, including ontologies, scientific data sharing, scientific workflows and social computing for scientists. She works in many application areas, but primarily in Life Sciences. She currently has a leading role in two major international initiatives: the Semantic Web and the Grid, combined into the Semantic Grid. She is an Editor-in-Chief of the Elsevier Journal of Web Semantics.

Robert Stevens is a senior lecturer in the BioHealth Informatics Group in the University of Manchester's School of Computer Science. His main research interests focus on the role of the supply of knowledge in a computational form to bioinformatics through the use of description logic ontologies. Another research interest is the use of workflow to industrialize bioinformatics analyses and the role of workflows as a form of knowledge within the scientific context. He brings a background in bio-chemistry, biological computation and computer science to these areas.