Available online at www.sciencedirect.com

ScienceDirect

Int. J. Human-Computer Studies 65 (2007) 583–594

International Journal of Human-Computer Studies

www.elsevier.com/locate/ijhcs

Using OWL to model biological knowledge

Robert Stevens*,[1], Mikel Egaña Aranguren[1], Katy Wolstencroft[2], Ulrike Sattler, Nick Drummond[3], Matthew Horridge[3], Alan Rector

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

Available online 4 April 2007

Abstract

Much has been written of the facilities for ontology building and reasoning offered for ontologies expressed in the Web Ontology Language (OWL). Less has been written about how the modelling requirements of different areas of interest are met by OWL-DL's underlying model of the world. In this paper we use the disciplines of biology and bioinformatics to reveal the requirements of a community that both needs and uses ontologies. We use a case study of building an ontology of protein phosphatases to show how OWL-DL's model can capture a large proportion of the community's needs. We demonstrate how Ontology Design Patterns (ODPs) can extend inherent limitations of this model. We give examples of relationships between more than two instances; lists and exceptions, and conclude by illustrating what OWL-DL and its underlying description logic either cannot handle in theory or because of lack of implementation. Finally, we present a research agenda that, if fulfilled, would help ensure OWL's wider take up in the life science community.

Keywords: OWL-DL ontology; Universals; Ontology Design Patterns; Biology; Bioinformatics; Phosphatase

1. Introduction

In this paper we investigate the ontological needs of biology and the associated discipline of bioinformatics. Much has been written about what knowledge representation languages such as the description logic (DL) variant of the Web Ontology Language (OWL) can offer domain experts in terms of modelling facilities (Dean et al., 2002). Much less has been written about what particular domains need to capture in such modelling languages. In this paper, we will put forth the knowledge modelling requirements of biology and bioinformatics. This will highlight the limits of modern description logics (DL) as knowledge representation languages. The expressive restrictions of DLs are well known (Baader et al., 2003, Chapter 1), in this article, we take the perspective of the needs of a domain, rather than a computer science research agenda.

OWL-DL is underpinned by a DL (Baader et al., 2003), a fragment of first order logic. This means that an OWL-DL ontology is expressed in a formalism with well-defined semantics and over which automated reasoning can take place. We will describe OWL-DL's use in this context and how it captures biology and bioinformatics domain knowledge in ontologies. One major question to be asked is whether the logical approach followed by OWL-DL suits the description of the natural world, with all its complexities and inconsistencies.

Bioinformatics is the use of computational and mathematical techniques to store, manage and analyse biological data to answer biological problems (Kaminski, 2000). At the centre of bioinformatics is the analysis of DNA and protein sequences. Its goal is to characterise nucleic acid sequences (genes) and their products, primarily proteins. Biology, however, is unlike physics and much of chemistry in that—although it contains many laws and models—few of these are reduced to a mathematical form. It is not possible to take a protein's sequence of amino acids, apply

*Corresponding author. Fax: +44 161 2756204.
E-mail address: robert.stevens@manchester.ac.uk (R. Stevens).

[1] Mikel Egaña Aranguren is funded by an EPSRC studentship.

[2] Katy Wolstencroft's work on the phosphatase ontology was funded by an MRC studentship.

[3] Matthew Horridge and Nick Drummond are funded by JISC - Semantic Grid and Autonomic Computing Programme grant.

1071-5819/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ijhcs.2007.03.006

some formula, and derive a set of characteristics such as location, functionality, forms of modification, regulation, etc.

Instead of mathematical laws, bioinformaticians use similarity. The central dogma of bioinformatics is that if an uncharacterised sequence is sufficiently similar to one that has been characterised, then the understanding can be transferred from the characterised to the uncharacterised. Many tools are provided for comparing sequences against databases of other sequences (Attwood and Miller, 2001). This search for similarity is, however, not simply done on the basis of some statistical measures. A good bioinformatician will use all the facts recorded about the entity and the nature of the matches between the sequences in order to infer any biological relationship. This is why both biology and bioinformatics have been characterised as a "knowledge based discipline" (Baker et al., 1999).

As a consequence of needing to record this knowledge in a consistent and computationally amenable form, ontologies of various kinds have become very important in bioinformatics (The Gene Ontology Consortium, 2000; Stevens et al., 2003).[4] Molecular biologists wish to describe and record a wide range of knowledge items. These include, but are not limited to:

- names of things;
- classifications (such as species);
- the size (absolute and ranges, both real and integers), shape and numbers of things;
- functions, processes and behaviours of things;
- structure and substance (atoms, molecules, tissues, etc.);
- evidence (both experimental and literature) for facts about the world;
- patterns (regular expressions in sequences indicative of some feature, etc.);
- parts of things to describe anatomy, composition of molecules and assemblies of molecules, etc.;
- the order of things and their transformation, such as life cycle stages, metabolic pathway reactions, exons in genes;
- degree of match and similarity of things.

The biology community has realised a need for ontology. OWL is a recommendation for the representation of ontology. It is pertinent, therefore, to examine OWL's ability to fulfil the ontological needs of the biological domain. As we will see, OWL-DL has its limitations in meeting these goals. the motivation for its use, however, in attempting to form ontologies of molecular biology are strong. OWL's ability to model incomplete, irregular knowledge fits well our incomplete, irregular knowledge of biology. OWL-DL's computational qualities of consistency checking and classification are also invaluable in creating coherent and useful ontological models of a very complex domain (Rector et al., 2001; Wroe et al., 2003).

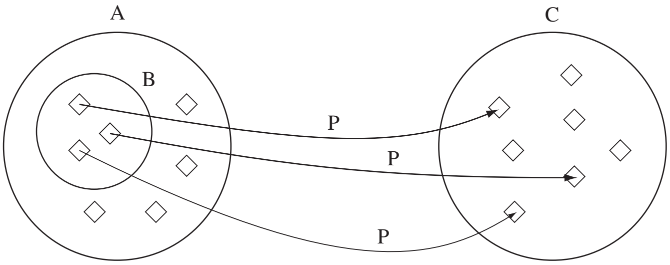

Fig. 1. An illustration of the set based semantics of description logics such as OWL-DL. Sets A, B and C contain instances. C is a subset of A, and all instances in C are also members of the superset A. If a property is held by one instance in C, then it must be held by all instances in C.

In Section 2 we describe the approach followed by OWL-DL and its modelling constructs and the application of automated reasoning to OWL-DL ontologies. This section can be skipped by those familiar with OWL-DL. We then present a protein family as a case study for ontological modelling in Section 3 and an ontology of that family in Section 4 to describe what can be straightforwardly captured in OWL-DL. Then, in Section 5 we show how some of the limitations of OWL-DL can be circumvented with the use of Ontology Design Patterns. Finally, in Section 6 we discuss what cannot be captured in OWL-DL and use Section 7 to provide a general discussion of the limitations of OWL-DL to represent knowledge in the life sciences.

2. The OWL-DL model of the World

DLs are a decidable fragment of first order logic and thus have a well-defined, two-valued semantics, i.e., they allow us to express what is universally true (Baader et al., 2003). In OWL-DL, the basic unit of an ontology is a *class*, which represents a set of individuals, its *instances*. Moreover, we consider *properties*, which represent (binary) relations between individuals. Individuals, together with the information about which individual is an instance of which class, and how the individuals are related *via* properties. Constraints[5] on such interpretations, and an interpretation that satisfies all constraints expressed in an ontology is called a *model* of this ontology; note that one ontology can have numerous such models. In Fig. 1, we show such an interpretation with three classes, A, B, and C, and one property p. In this ontology, all instances of B are also instances of A, and all instances of B have a p-successor which is an instance of C. If, in all models of an ontology, the instances of B are also instances of A, then B is called a *sub-class* of A. Recall the above remark on OWL-DL's ability to express what is universally true, and note how the word *all* occurs frequently in our intuitive description of OWL-DL's semantics.

OWL-DL allows, for example, to express that each instance of class B is related to (at least) one instance of C

[4] See http://obo.sf.net

[5] We use the word "constraints" here in a completely informal way.

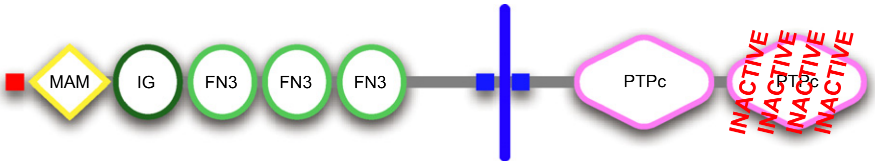

Fig. 2. The different functional domains of a receptor tyrosine phosphatase type R2A. From right to left, the p-domains are: MAM—Meplin/A5 domain, IG—immunoglobulin-like domain, FN3—fibronectin III type repeat, transmembrane domain, and PTPc protein tyrosine phosphatase catalytic domain. The second catalytic domain is inactive.

via the property p. The interpretation in Fig. 1 satisfies this restriction, and thus could be a model of an ontology containing such a restriction. Note that the *vice-versa* is not automatically the case, i.e., some instances of C do not participate in a p relationship with an instance of B (or any other individual). In addition to such an "existential" restriction—we require that, for each instance of B, there "exists" at least one p-successors in C—OWL-DL also allows for *universal* and *cardinality* restrictions: the former allows to state that *all* p-successors of a class have to be instances of a certain class, and the latter allows us to restrict the *number* of p-successors of a certain class. Finally, OWL-DL allows us to combine these and more restrictions using the usual Boolean constructors like AND, OR, and NOT. Using restrictions OWL-DL allows two different ways to describe a class: a class can be described by expressing *necessary* or *necessary and sufficient* conditions for an individual to be an instance of this class.

Finally, OWL-DL assumes an open world rather than the closed world approach used in systems such as databases. In closed world models, if something is not (explicitly or implicitly) stated, then it is assumed not to be the case. In contrast, following an open world approach, such as in OWL-DL, if something is not (explicitly or implicitly) stated, then it may or may not be the case. For example, when defining a class by means of the parts its instances have, if one wishes to state that the parts mentioned are all and only the parts of the instances of this class, then a so-called closure axiom must be used (Rector et al., 2004). Having stated that certain `Parts` exist for the class using an existential restriction, a universal restriction can be used to restrict which `Parts` are permitted. This openness is a feature of OWL-DL whose applicability to modelling biology will be discussed in Section 4.

An OWL-DL ontology corresponds to a set of logical formulae and, for a given ontology, this set can be generated automatically and then be submitted to a description logic reasoner, such as FaCT++ (http://owl.man.ac.uk/factplusplus/; Tsarkov and Horrocks, 2004), Pellet (Evren Sirin, 2004) or RACER (Haarslev and Möller, 2001) etc. The reasoner will then (a) check each class as to whether it is consistent with the ontology (if not, this indicates a modelling error which should be repaired) and (b) compute the subsumption hierarchy, taking into account all constraints given in the ontology. The latter reasoning service can reveal new, useful subsumptions between classes as well as un-intended ones, which might indicate, again, a modelling error which needs to be repaired. Finally, given a description of an individual, Pellet, Racer, and the Instance Store (Bechhofer et al., 2005) can compute the classes of which this individual is an instance.

3. A knowledge case study

Proteins are divided into broad functional classifications called families. Protein phosphatases and protein kinases are two families that control the phosphorylation events in a cell (Alberts et al., 1989).[6] Biologists classify phosphatases according to their functionality and evolutionary relationships to each other. Tertiary structure units within a protein can form functional areas within a protein called domains (see Fig. 2). The presence of a particular collection of what we will refer to as p-domains can, in some cases, be diagnostic for a particular function for a given protein. Tyrosine phosphatase specificity is, for example, determined by the composition of p-domains found within each class of single sequence proteins (see Fig. 3 for an example).

In contrast to the single sequence tyrosine phosphatases, serine/threonine protein phosphatases are multi subunit complexes, combining a catalytic subunit with regulatory and targeting subunits. The final combination of subunits produces the resulting number of each serine/threonine phosphatase in a given organism.

From this description of how the functionality of a protein family is determined, we immediately see some of the many kinds of biological knowledge we need to capture in order to have a computationally useful form of this understanding:

(1) a classification in terms of classes and sub-classes, with sub-classes retaining features of its super-class;
(2) the functions and processes in which the enzyme partakes;
(3) the chemical entities with which a phosphatase interacts, binds, transforms, etc.;
(4) the composition or parts of proteins, in the case of phosphatases this is mainly p-domains;

[6] A phosphatase is an enzyme that removes a phosphate group from another chemical and a kinase is one that adds a phosphate group to another chemical.

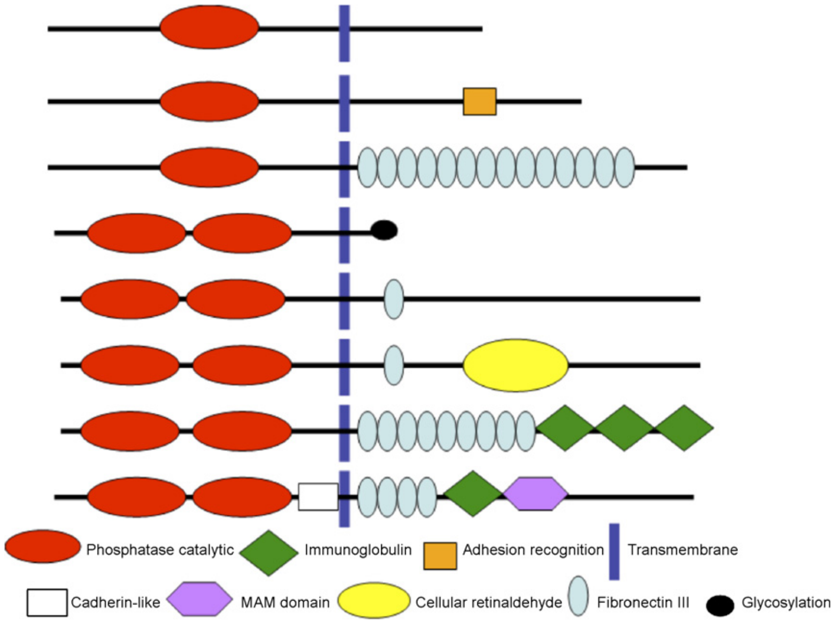

Fig. 3. The protein domains present in the tyrosine phosphatases. Each type has the defining catalytic subunit, but each type has a unique set of protein domains. (after Andersen et al., 2001). The presence of a particular set of domains is enough to determine class membership. In this particular case, order of domains does not have any influence on type membership, though it would on protein function.

(5) the numbers of identified parts present;
(6) the presence of specified numbers of protein subunits to form complexes.

4. A phosphatase family ontology

We developed a phosphatase ontology to help semi-automatically support a phosphatase protein family database (Wolstencroft et al., 2005a,c) and to automatically classify proteins found in a genome (Wolstencroft et al., 2005b, 2006).

In this ontology, the classes of phosphatase were defined in terms of their p-domain composition. Fig. 3 shows how the p-domain composition of each protein can be *sufficient* to recognise to which phosphatase sub-family it belongs. We know, for instance, (from the literature) that *all* instances of protein tyrosine phosphatase are also instances of protein phosphatases, and thus are all phosphatases. Then, for example, *all* members of the receptor tyrosine R2A phosphatases are tyrosine phosphatases and in turn protein phosphatases. In OWL-DL, this corresponds to declaring that the class `Protein tyrosine phosphatase` is a subclass of `Receptor phosphatase` which, in turn, is a subclass of `Phosphatase`. This pattern of strict sub-class relationships is repeated throughout the functional classification of phosphatases, i.e., we do not have to cope with “exceptions” or similar phenomena.

Moreover, we know that having at least one of the possible phosphatase catalytic domains is sufficient to be recognised as any kind of phosphatase, and that all phosphatases have at least one of these p-domains. Similarly, a phosphatase having a transmembrane p-domain is a receptor tyrosine phosphatase. In OWL-DL, this translates to necessary and sufficient class definitions, and OWL-DL reasoners can use these to infer, for example, that a given phosphatase which happens to have a transmembrane p-domain is recognised as a receptor tyrosine phosphatase.

Thus, the phosphatase family of proteins can be easily modelled in an OWL-DL ontology: *all* instances (without exceptions) of each subclass of `Phosphatase` satisfy *all* restrictions specified for this class and, if an instance does (or does not) satisfy these restrictions, then it is (or it is not) a member of that class of protein phosphatases.

For example, a member of the R2A phosphatase subfamily (see Fig. 2) contains:

- 2 protein tyrosine phosphatase p-domains (from the `Protein tyrosine phosphatase` class);
- 1 transmembrane p-domain (from the `Protein tyrosine phosphatase` class);
- 4 fibronectin p-domains;
- 1 immunoglobulin p-domain;
- 1 MAM p-domain;
- 1 cadherin-like p-domain.

All R2A phosphatases must have these p-domains and any protein with all these p-domains is an R2A phosphatase. More generally, for each class of phosphatase, our ontology contains a (necessary and sufficient) definition. For this family of proteins, this definition is a conjunction

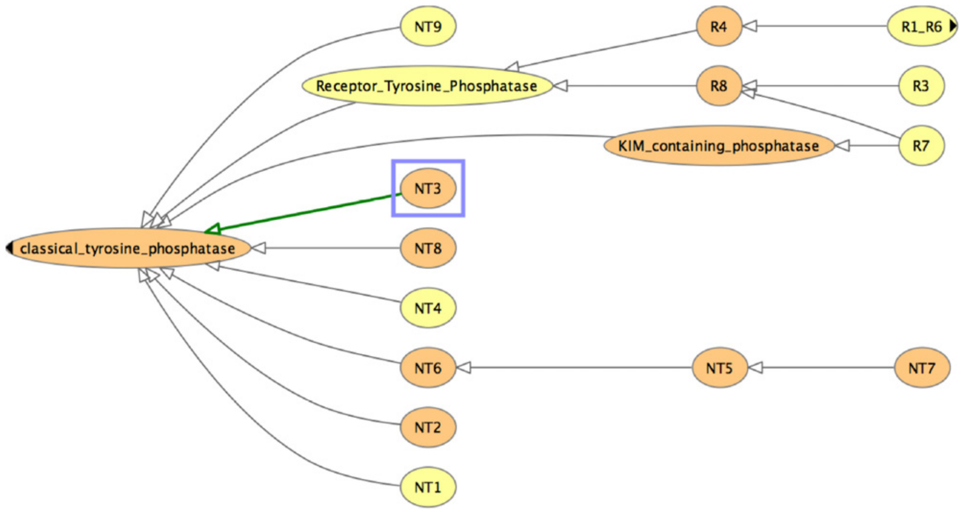
NT9
R4
R1_R6
Receptor_Tyrosine_Phosphatase
R8
R3
KIM_containing_phosphatase
R7
NT3
classical_tyrosine_phosphatase
NT8
NT4
NT6
NT5
NT7
NT2
NT1

Fig. 4. An OWLViz display of the classification of classical tyrosine phosphatases inferred from the OWL-DL definitions.

of p-domain compositions. A typical case of a phosphatase class definition is as follows, where X_i are p-domains:

> *If a Y protein contains exactly n_1 p-domains of type X_1 and … exactly*
> *n_m p-domains of type X_m, then this protein also belongs to class Z.*

A snapshot of the resulting OWL-DL ontology for the classical tyrosine phosphatases using the OWLViz[7] tool in the Protégé OWL plugin (Holger Knublauch and Musen, 2004) can be seen in Fig. 4.

For the phosphatase ontology, OWL-DL's strict, two-valued view of the world suits modelling of phosphatases based on p-domain composition, almost perfectly:

- Description of classes and subclasses (interpreted as sets and subsets of proteins) reflect the biologist's classification of proteins in a taxonomy: the different functional families and sub-families of phosphatase are interpreted as sets, all of whose instances share the same properties. In addition, sub-families simply extend these properties and are therefore true sub-classes.
- Our knowledge about phosphatases allows us to make use of OWL-DL's necessary and sufficient class definitions. Such definitions are required for the automatic recognition of individuals to be instances of a class. This is possibly more specific than the one mentioned explicitly for this individual—due to the properties of this individual which "match" the definition of this more specific class.
- OWL-DL's restriction to unary predicates (i.e., classes) and binary predicates (i.e., properties) is fine for phosphatases since the only important predicate, in this restricted case, is `contains`, which holds between (instances of the class) phosphatases and (instances of the class) p-domains.
- OWL-DL's ability to distinguish between "all instances of class X have a p-successor which is an instance of class Y" and "all instances of class Y are p-successor of some instance of class X" reflects our biological knowledge well: for example, every tyrosine phosphatase has a fibronectin p-domain, but a fibronectin p-domain does not need to occur in a phosphatase since they can also occur in other receptor proteins.
- OWL-DL provides full Boolean operators. In our ontology, we use negation, for example, to express disjointness of certain p-domains: being an instance of a p-domain class X implies being *not* an instance of a p-domain Y.

  Disjunction is used to describe that one instance of a choice of p-domains must be present in a protein. For example, a classical tyrosine phosphatase has at least one low molecular weight phosphotyrosine *or* one tyrosine specific with dual specificity p-domain.
- In contrast to other formalisms, OWL-DL allows use of *complex* class descriptions in restrictions. That is, we can describe a certain class of phosphatases by requiring that they contain a p-domain from a disjunction of p-domains, without our ontology having a class defined for this disjunction. This means that we do not have to clutter our ontology with supplementary class definitions, and this helps to construct a clean, concise ontology.
- The first point where OWL-DL ceases to be of adequate expressive power are "number restrictions". As our example indicates, it is not enough to say that there is at least one (existential quantification) or no (universal quantification) p-domains of a certain kind: an R2A phosphatase contains *exactly* 4 fibronectin p-domains. In OWL-DL, we can say that an R2A phosphatase contains *exactly* 4 "things", but we cannot *qualify* of which kind these 4 things should be, which is important since R2A phosphatases contain many other p-domains.

  The Protégé OWL plugin (Holger Knublauch and Musen, 2004) and the reasoner Racer which we used

[7] http//www.co-ode.org/downloads/owlviz

(Volker Haarslev, 2001), however, support OWL-DL extended with this kind of *qualified* number restrictions, and thus (after some initial problems Wolstencroft et al., 2005b) we did not encounter any problems. This feature will also be supported in OWL 1.1, a W3C member submission in preparation.[8]

OWL's open world assumption (see Section 2) is also appropriate in this case and also much of biology. At the level of an ontology, this assumption fits neatly with knowledge about biology, which is certainly not complete. For example, we have said that a certain class of phosphatases contains a particular conjunction of p-domains. Unless we place a closure axiom on that description, we are not stating that an instance of that class can contain no other kind of protein domains. In addition, even though we have said nothing about other protein features, chemicals that bind to the protein, post-translational modifications, etc., this does not imply they do not exist. In the closed world model of a database, the implication of these things not being explicitly stated would be that they do not exist. Consequently, the open world assumption of OWL-DL suits our biological domain well. The consequences of the open world assumption for the instance level is not so clear cut, but is beyond the scope of this paper.

5. Using OWL with Ontology Design Patterns

In this section, we will concentrate on those limitations of OWL that can be worked around by using Ontology Design Patterns (ODPs).[9] We do not exhaustively explore ODPs for OWL, but illustrate how they can use OWL-DL's current expressivity to work around some of the inherent limitations of OWL-DL.

ODPs are based on the same principle as Design Patterns in Object Oriented Programming (Gamma et al., 1995), which are abstractions of workable solutions for modelling common problems in software design. In the case of ODPs, the same principle is applied to common problems in modelling in a knowledge domain. We can split ODPs into three categories: *modelling* patterns, *limitation* patterns, and *domain* patterns. Limitation patterns are used to circumvent the limitations of OWL-DL's expressive power. We use the examples of *n*-ary relations, exceptions, and lists to describe how aspects of our knowledge about phosphatases can be expressed in an OWL-DL ontology, despite the fact that they go beyond OWL-DL's immediate modelling capabilities. Modelling patterns capture *best practice* in ontology development, such as the value partitions used in ontology normalisation (Rector et al., 2001). Domain patterns are those designed to capture the particular peculiarities found in domain knowledge. A brief example of modelling patterns would be that in describing physical objects, there is a common pattern of needing classes for the object itself; parts of that object; assemblies of parts of that object and, finally, assemblies of the object itself. A good example of this is the description of the anatomy of a flower. We will not discuss the usage of either modelling patterns or domain patterns any further in this article.

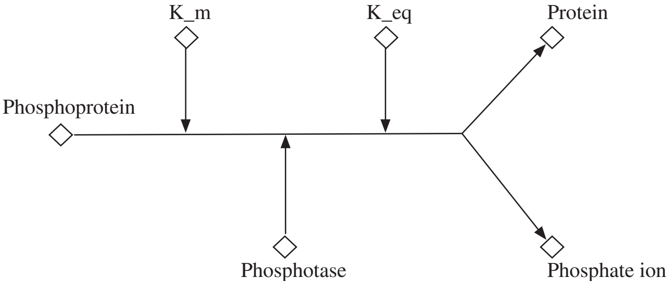

Fig. 5. In the reaction catalysed by a phosphatase, a phosphoprotein is the typical substrate and the hydrolysed products are a phosphate ion and the protein. The two constants are individual concrete values for this particular reaction.

5.1. Relationships of higher arity

As described in Section 2, OWL-DL only provides properties, which correspond to *binary* relationships, i.e., are interpreted as *pairs* of individuals. It is often desirable, however, to use relations that link more than two individuals at the same time. For example, in our phosphatase ontology, we wish to describe the fact that phosphatases catalyse reactions that link many individuals and thus require relations of arity higher than two. A phosphatase catalyses a reaction from a substrate to a product with an equilibrium constant (K_{eq}) and Michaelis–Menten constant (K_{M}). Such a *n*-ary `Catalyses` relationship is shown in Fig. 5, where a phosphoprotein has derivatives into phosphate ions and protein, and where the `Catalyses` relationship involves six individuals.

To represent the `Catalyses` relationship in OWL-DL, the `Catalyses` relation is turned into the class `PhosphataseCatalysis`. Instances of the `Phosphatase-Catalysis` class represent the 5-ary relation, and bind together the individuals in this 5-ary relation using the binary relationships `Has_substrate`, `Has_product` and `Has_constant`. The resulting structure is depicted in Fig. 6. The `Catalyses` relationship is a typical *derivation* relationship between `Physical Continuants`. using a class to represent this relationship produces a class of `Occurant`.

Fig. 7 shows the OWL statements for constructing the `Phosphatase-Catalysis` class. This ODP, however, only approximates *n*-ary relations—for a more precise reification, see Calvanese et al. (2001, Section 6.5).

[8] http://www-db.research.bell-labs.com/user/pfps/owl/overview.html

[9] Some patterns, including the n-ary relationship below, can be found in the Semantic Web Best Practices and Deployment Working Group http://www.w3.org/2001/sw/BestPractices/ and the Wiki http://esw.w3.org/topic/FrontPage

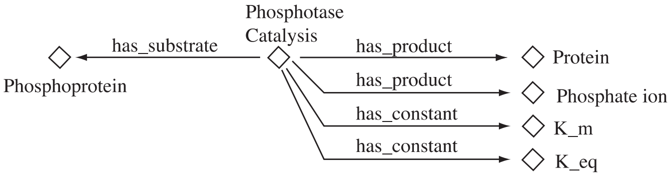

Fig. 6. A model of the catalysis of phosphoprotein. Each diamond represents an individual. The labels next to individuals indicate the class that the individual is a member of.

```
Class: PhosphataseCatalysis
Necessary:  has_substrate some Phosphoprotein
            has_product some Phosphate_ion
            has_product some Protein
            has_constant some K_m
            has_constant some K_eq
```

Fig. 7. Description of the OWL-DL expression of the *n*-ary relationship ODP.

The *n*-ary relation pattern is a very important ODP for the biological world. Whilst much can be modelled with binary relationships, there is a wide need for relationships of higher arity. There is much in biology that is not absolute, as we will see in the next section, and it is often the case that we need to say more about relationships between things other than that it is simply in existence. Strengths of observations; probabilities; severity; authorities; sources; evidence; etc. are just a few of the cases in which *n*-ary relationships will be desirable.

5.2. Exceptions

OWL-DL talks in "universals", i.e., class definitions state what is always true for all its instances. Yet the idea of exceptions (Ringland and Duce, 1988) is strong in biology. From the classic case of birds normally flying (but not penguins, ostrichs, emus, ...) to molecules, there are exceptions. We will present an ODP for a well-behaved case of exceptions that relies on making the exceptions explicit and on the use of a reasoner to maintain a coherent class hierarchy.

For example, eukaryotic cells are canonically defined as cells with a proper nucleus (Alberts et al., 1989). In principle, having a proper nucleus is a necessary and sufficient condition for a cell to be considered eukaryotic—there are, however, cells that are considered eukaryotic and lack a proper nucleus, like mammalian red blood cells. In OWL-DL, the class MammalianRedBloodCell would inherit the condition of having a proper nucleus (hasNucleus = 1) if it is a subclass of EukaryoticCell, which is incorrect. Trying to describe a world in which universals are not always easily found is always a non-trivial task (see Fig. 8).

An exception pattern for a class *X* consists of the following steps, which are illustrated below using the example of eukaryotic cells:

- Two new, disjoint subclasses of *X* are introduced, one that accounts for the typical and one that accounts for the atypical case.[10]
- We state that *all* instances of *X* must be an instance of one or the other of these two subclasses by use of a covering axiom.
- The conditions to which exceptions are known are not used in *X*'s class definition, but are pushed down into the Typical and Atypical subclasses.
- All other subclasses of *X* remain unchanged, thus possibly as "siblings" of the new subclasses.

Thus this ODP is based on disjoints and covering axioms, negation and restrictions. For a more detailed description, see Rector (2004).

A EukaryoticCell is defined using only those restrictions that are genuinely universal, i.e., held by *all* EukaryoticCells. Examples would be the possession of a cell wall and the use of mitochondria to generate energy. Since only members of typical EukaryoticCell contain nuclei, we state this restriction (HasNucleus = 1) only in the subclass TypicalEukaryoticCell and not in AtypicalEukaryoticCell.

Next, if we made RedBloodCell a sub-class of TypicalEukaryoticCell, this would imply that each red blood cell has a nucleus. MammalianRedBloodCell is clearly a subclass of RedBloodCell, and mammalian red blood cells do *not* have a nucleus. Therefore, as discussed above, we should make RedBloodCell a subclass of EukaryoticCell, use the same pattern of either having or not having a nucleus on the level of RedBloodCell (creating the subclasses TypicalRedBloodCell—with the restriction hasNucleus = 0—and AtypicalRedBloodCell), and then let the reasoner infer the proper sub-class relationship. The classified model of various cells is shown in Fig. 9. In the case of red blood cells most of the vertebrate groups have a nucleus, so in this example MammalianRedBloodCell is considered by the reasoner a subclass of AtypicalRedBloodCell, as mammalian red blood cells lack a nucleus, whereas AvianRedBlood Cell (all avian red blood cells have a nucleus), is considered by the reasoner a subclass of TypicalRedBloodCell. In the level at the top of the classification, the class MammalianRedBloodCell (no nucleus) is a subclass of AtypicalEukaryoticCell and AvianRedBloodCell (with a nucleus) is a subclass of TypicalEukaryoticCell. Both are subclasses of RedBloodCell, which itself is still only classified as far as being a subclass of EukaryoticCell, as it lacks

[10] Here the prefixes 'typical' and 'atypical' are used for illustrative purposes only. In this biological example, terms like 'nucleate' and 'non-nucleate' might be used.

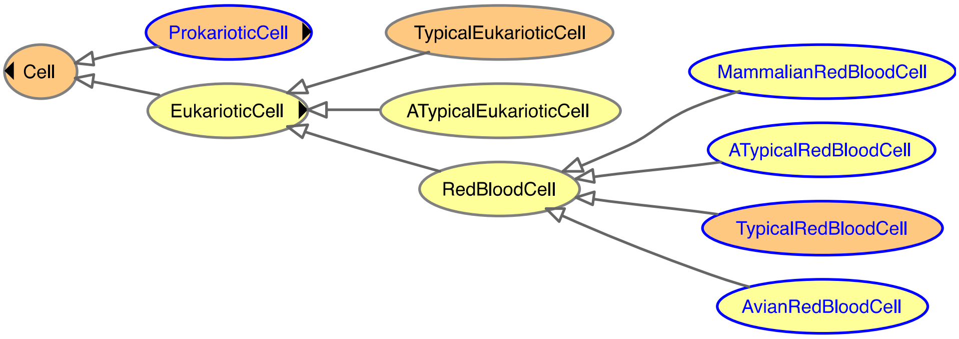

Fig. 8. The exception ODP used twice to describe cells: first those ones that are exceptions to the definition of `EukaryoticCells` and second those ones that are exceptions to definition of `RedBloodCells`.

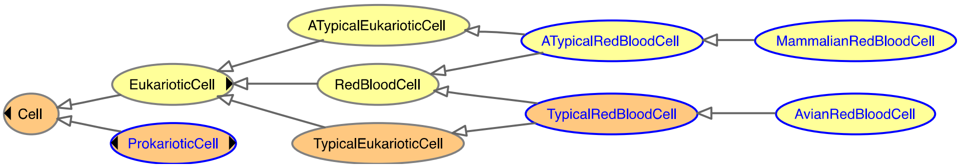

Fig. 9. The `EukaryoticCell` exception ODP after classification. Notice the multi-parent classification that has been computed by the reasoner.

sufficient restriction on presence of nuclei to place it further down the classification.

This ODP suffices for simple exceptions. However, exceptions can be piled upon exceptions, eventually leading to a combinatorial explosion. Worse still, some cells, such as muscle cells, have many nuclei. This means we would have to model a three way split with zero, one or many nuclei in a cell. Not only will this become unmanageable, even with reasoning, but it is also likely to contradict what biologists expect to see since all these additional classes clutter the ontology. In contrast, they might wonder why it is not possible to simply read the subclass relationship with a slightly less strict semantics, i.e., as "*if not stated otherwise*, the instances of a subclass inherit all properties specified for its superclasses". Nevertheless, this ODP provides a small increment in what OWL-DL can represent at the class level.

5.3. Lists

Being based on first order logic, OWL-DL does not provide means to construct or talk about lists of individuals, yet this feature is a regular need in modelling the biological world. Hence we developed a list ODP, which is similar to LISP (Steele, 1990) lists.[11] Following this pattern, a list has a head, which points both to the `Next` list element and its `Contents`, and it is terminated by a *null* element. The OWL-DL statements that are used to represent a list in this manner are very complex—so much so that the Protégé OWL plugin has a list wizard in order to create OWL-DL lists.[12]

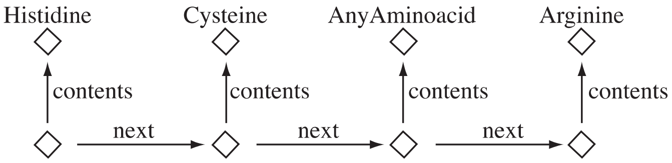

Fig. 10. The phosphatase motif. Each diamond represents an individual. The labels next to individuals indicate to which class of `AminoAcid` the individual belongs.

Biology is full of such list structures, for example sequences of nucleic acids or amino acids, genes, pathways, so-called life-cycles, etc. Within OWL, it is possible to create class expressions that represent fully or partially specified sequences of elements. For example, lists that start, end, contain or exactly match a given sequence of elements. We can then use a reasoner to classify these sequences—e.g., the expression "lists containing a Cysteine followed by a Lysine or aAsparagine, followed by a small amino acid," subsumes "lists that start with Histidine followed by Cysteine, then Lysine and then glycine".[13] Being able to abstract about the elements that make up a sequence allows biologists to describe more general families of proteins or genes, and provides automatic classification of particular "concrete" examples. These lists are class expressions and being classes can be used with standard reasoning just as any other class. Protein motifs can also be

[11] Although RDF(S) has a list specification, the RDF properties that are used to construct the list cause an OWL-DL ontology to fall into OWL-Full.

[12] See www.co-ode.org

[13] These are all amino acid side chains and this kind of list might be used to describe a particular site of biological interest.

expressed in this manner. Such protein motifs use a range of abstractions for amino acids, based on the various restrictions placed upon them—such as *hydrophobicity*, *size*, *polarity*, etc. Fig. 10 shows a motif of amino acids which is characteristic for phosphatases, and which is represented by the following (abbreviated) OWL-DL statement:

```
List AND
    contents SOME Histidine AND
    next SOME (List AND
        contents SOME Cysteine) AND
            (next SOME (List AND
                (contents SOME Aminoacid) AND
                (next SOME ...
```

Since this motif is characteristic for phosphatases, our ontology also contains a statement which ensures that any protein whose sequence of amino acids contains this motif is an instance of the class Phosphatase. However, there are protein motifs which cannot be described in OWL-DL, even with the list ODP. We will discuss such motifs in Section 6.

Just as with software engineering, ODPs deliver common, tried and tested solutions to well-known problems. In OWL, these are a mixture of patterns that model common (biological) knowledge; deliver techniques to avoid messy or tangled ontologies (Rector et al., 2001); and work around some of the limitations imposed by OWL-DL's restricted expressive power. While much of biology can be represented in straight-forward OWL, examples that need the three patterns described here are by no means uncommon. This class of ODP are perhaps *ad hoc* work arounds for the limitations of OWL-DL, but they do allow more to be said in an OWL ontology. As the expressive power of OWL-DL increases, their use will no longer be needed. Next, we will discuss issues that not only go beyond OWL-DL, but also beyond the scope of these patterns.

6. The boundary of the OWL World

In this section, we cross the boundary of the OWL-DL view of the world and explore aspects of biology that OWL-DL cannot represent. Some of these aspects cannot be expressed in OWL-DL or any decidable description logic, because they are known to lead to undecidability, semantic problems, or currently unmanageable computational complexity. Of these other aspects cannot be expressed in OWL-DL, but it is known that an extension of OWL-DL with the corresponding expressive means would be possible. These expressive means are known to be "harmless" for the performance of the reasoning algorithms, and appropriate algorithms already exist—possibly not only on paper, but possibly even in prototypical implementations.

Qualified number restrictions: As already mentioned in Section 4, our phosphatase ontology heavily used *qualified* number restrictions—which are not present in OWL, yet are supported by Protégé OWL and DL reasoners such as Racer, and they are thus of the second group of aspects mentioned above.

Fuzziness, *probabilities*, *and similarity*: As described in Section 1, much of bioinformatics works on degrees of similarity, and often words such as *nearly*, *mostly*, *very*, *probably*, *similar to*, etc. are used in bioinformatics and biological descriptions. The notion of species, in particular, is a very fuzzy notion, especially when descriptions are made higher up the taxonomic tree (Pullan et al., 2000). Other biological examples include: *nearly* all mammals give birth to live young. *Most* phosphatases have at least one active catalytic domain. A *very* small mouse is one whose size is "very small", without wanting to fix a range for "very small". The human heart is *probably* located in his or her left-hand side of the chest. The protein sequences associated with any two instances of tyrosine phosphatases are similar. Another example of similarity and fuzziness comes in the matching of motifs in proteins as described in Section 5. It is usually the case that an individual protein does not have to exactly match a motif for a biologist to infer the feature indicated by that motif is present. As long as the protein *mostly* matches *nearly* all elements of the motif, then sufficient conditions have been met.

Currently, to the best of our knowledge, only very limited reasoning support is available for this kind of knowledge; see e.g., http://faure.iei.pi.cnr.it/~straccia/software/alc-F/alc-F.html. Some theoretical work has been carried out in the DL community, and we refer the reader to Baader et al. (2003) for fuzzy and probabilistic DLs, and to the proceedings of recent DL workshops.[14]

Prototypes, *exceptions*, *and defaults*: As described in earlier sections, exceptions are rife in biology. We have seen with the phosphatases that OWL-DL's strict view on conditions can work much of the time. We would assert, however, that any area of biology will eventually meet with exceptions to general conditions for class membership. Biologists often use prototypes to describe properties of classes which might not be followed strictly in all subclasses but, to the best of our knowledge, the role played by prototypes or their semantics has not yet been clarified in OWL-DL.

Each of the examples for exceptions can also be modelled with the ODP described in Section 5, but the class structure rapidly becomes arcane. In addition, though representing the same view of the world, this structure is cluttered with classes that are unfamiliar to a biologist, who might well struggle to interpret such an encoding. In contrast, saying, for example, that *all* eukaryotic cells have nuclei, and then make appropriate exceptions, would match a biologist's view of the world.

[14] http://dl.kr.org

As for the previous case, only a little support is currently available, and only little is known about how OWL-DL could be extended to suit exceptions better, and what the impact of these extensions would be. In the past, a few extensions of DLs with defaults and other non-monotonic operators have been investigated, and we refer the reader again to Baader et al. (2003) and to Rosati (2005), Eiter et al. (2004) for combinations of DLs with a logic programming approach with negation as failure.

Complex property restrictions: Despite the fact that OWL-DL allows one to state that one property is a subproperty or the inverse of another one, or that a property is transitive, other interesting things one would like to say about properties cannot be expressed.

For example, we would like to say that, whenever a metal ion x is bound to a phosphatase which catalyses a dephosphorylation of a protein y, then x regulates the dephosphorylation of y. This statement would require us to express that a *composition* of properties implies another one, which is not possible in OWL-DL.

Another aspect concerns disjointness or reflexivity of properties: e.g., we would like to say that certain regions of a gene, so-called introns, catalyse their own removal (Cech, 1990). That is, for the class `Intrans`, the property `catalyses` should be reflexive. Otherwise, we would like to say that an individual `Chemical` cannot have both an "atomic weight" and a "molecular weight". To achieve this, we could declare the properties `hasMolecularWeight` and `hasAtomicWeight` as disjoint.

An extension of OWL-DL with all these aspects is, in principle, possible, either directly in a DL (Horrocks et al., 2005) or by combining them with rules, see, e.g., Motik et al. (2004), Rosati (2005). However, the support currently provided by reasoners is limited to the latter approach.[15]

Expressive datatypes: Biology is an observation-based discipline and many observations are based on measurements. These measurements are, of course, represented numerically. OWL-DL's syntax and reasoners can already deal with integer values, but the biological world is not that neat—biologists deal with real numbers as least as much as they do with integers. In addition, it is rare for biological observations to fall at one, precise value. It is often necessary to describe instances with a number range. For example, a biologist might wish to describe a `Small mouse` as one that `has length 2–3` cm.

Analogous to the above case of qualified number restrictions, which can be easily added to OWL-DL and form an important part of a bio-ontologist's requirements, so are the description of classes in terms of *concrete domains* or, more precisely, datatypes such as spatial regions, size, weight, etc. (Baader and Hanschke, 1991; Lutz, 2003). This is a well-understood area of DLs for which, e.g., Racer provides sophisticated reasoning support.

Regular expressions: As mentioned earlier lists or sequences of individuals play a central role in biology, and certain *motifs* often characterise classes. For example, the motif characteristic for tyrosine phosphatase is H-C-X(5)-R which means an occurrence of first histidine, then cysteine, then any five amino acids, and then an arginine (Mulder et al., 2005). Whereas this motif can still be expressed using the list ODP, this ceases to be the case if we replace "then any five amino acids" with "then any number of amino acids": such a motif would require a transitive *closure* operator, which is not provided by OWL-DL.

The impact of this transitive closure operator or, more generally, regular expressions over properties on the computational complexity of DLs is rather well understood and can be said to be similar to the impact of the role operators provided by OWL-DL (Baader et al., 2003), yet (to the best of our knowledge), no reasoning support is currently available for such an extension.

7. Discussion

In this paper, we have explored the ontological requirements posed by biology and bioinformatics and how well OWL-DL's model matches those requirements. There are obviously large areas of the world of biology that can be represented using OWL-DL with great success. It is possible to create OWL-DL descriptions of molecular biology that are both ontologically good and useful in driving applications. Yet, it is relatively easy to find features of biology that do not fit into this strict, universal view. For instance, the ability to represent temporal aspects of biology are missing from the static model in OWL-DL. We have tried to categorise these limitations into those where Ontology Design Patterns can overcome these limitations, and those for which no such solutions exist. The latter group can be, in turn, partitioned into those where a solution is known to exist in principle. Among these, we can distinguish between features:

(1)
- For which even reasoning and tool support exists, such as qualified number restrictions and more expressive datatypes; and
- for which the possibility of such support has only been proven "on paper" such as more complex property restrictions or regular expressions.

(2) It is known that an extension of OWL-DL with a solution for this feature would lead to the undecidability of reasoning problems, and should therefore be provided in "higher" levels of ontology languages. For example, certain forms of more complex property restrictions or rule extensions fall into this category.

(3) Nothing or little is known about how a solution would affect the computational properties of OWL-DL. Examples for this category are defaults and fuzziness.

Note that there are limitations of OWL-DL that we did not mention here. For example, OWL-DL is based on

[15] Some of these features will appear in the OWL 1.1 submission to the W3C for admission to the OWL recommendation.

individuals and classes, and does not provide classes of classes, so-called meta-classes. Even though the standard example for this feature is ''species'', this limitation did not bother us in our particular application. This is not true in general, this kind of meta-modelling, especially when talking about ''species'', is a serious draw-back in OWL that needs to be addressed.

From the above observations, we can compose a list of interesting suggestions and (research) questions:

- It would be helpful if qualified number restrictions would be added to OWL-DL.
- Solving the problems that, so far, prevented OWL-DL from supporting more expressive datatypes, and extending existing reasoners to support a wide range of interesting datatypes. Currently, the uptake of OWL-DL in the life sciences community is hindered until such support is present in the language, reasoners, and the supporting tools.
- As more OWL ontologies are made, more Ontology Design Patterns will be needed to work around the limits of OWL-DL, as well as better designed ontologies and better modelling of domains. To support the domain expert in the usage of these patterns and, in general, to develop a methodology for designing ''good'' ontologies, are still open and interesting problems.
- Clarifying the features that fall under Point 3 above and developing solutions for them is clearly another important issue. From a biology perspective, the representation and reasoning over fuzziness, probability and exceptions is of particular interest.

There are some aspects of how scientists currently understand molecular biology that are difficult or clumsy to model in OWL-DL. It seems likely that our knowledge of biology will not be purely monotonic and attempting to create broad, deep ontologies of biology in such a formalism as OWL-DL will be consequently difficult. There are, however, significant portions of molecular biology that are amenable to OWL-DL's strict and formal view, and there are ways to overcome some of the limitations in OWL-DL and we can expect future extensions of the language to greatly increase what can be modelled.

It is not possible to quantify what proportion of the biological world would fall into modelling that requires current OWL-DL and then each of the proposed extensions. The scope of biological knowledge is open and we do not know what we do not know. We have seen that our particular example works well, but we could take another protein family that does not. The protein phosphatase family could be modelled without use of concrete domains, but others will rely heavily upon such a facility. Perhaps only one generalisation can be made: OWL-DL as it currently stands is more applicable to modelling biological facts as they are observed, rather than how biologists would like the world to be. This is the case in, for example, prototypes and exceptions. It is not possible, however, to avoid fuzziness in biology; it is a statistical science and life is, perhaps, not as neat as we would like. It is clear, however, that it is not possible to say what expressivity is needed in an area of biology *a priori*. Finding islands of consistency and modelling them in OWL-DL will drive forward DL research and make significant contributions to the use of knowledge in a computational form within bioinformatics. OWL's logic representation means community understanding can be captured with high-fidelity and make that understanding computationally amenable. The potential benefits to both communities are consequently great.

References

Alberts, B., Bray, D., Lewis, J., Raff, M., Roberts, K., Watson, J., 1989. Molecular Biology of the Cell. Garland, New York.

Andersen, J.N., Mortensen, O.H., Peters, G.H., Drake, P.G., Iversen, L.F., Olsen, O.H., Jansen, P.G., Andersen, H.S., Tonks, N.K., Moller, N.P., 2001. Structural and evolutionary relationships among protein tyrosine phosphatase domains. Molecular and Cellular Biology 21, 7117–7136.

Attwood, T., Miller, C., 2001. Which craft is best in bioinformatics? Computers and Chemistry 25, 329–339.

Baader, F., Hanschke, P., 1991. A schema for integrating concrete domains into concept languages. In: Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI-91), Sydney, pp. 452–457.

Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.F. (Eds.), 2003. The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, Cambridge.

Baker, P., Goble, C., Bechhofer, S., Paton, N., Stevens, R., Brass, A., 1999. An ontology for bioinformatics applications. Bioinformatics 15 (6), 510–520 URL ⟨http://www.cs.man.ac.uk/s̃tevensr/papers/bioinformatics%-ontology99.doc⟩.

Bechhofer, S., Horrocks, I., Turi, D., 2005. The OWL instance store: system description. In: Proceedings of the 20th International Conference on Automated Deduction (CADE-20), Lecture Notes in Artificial Intelligence. Springer, Berlin, pp. 177–181.

Calvanese, D., De Giacomo, G., Lenzerini, M., Nardi, D., 2001. Reasoning in expressive description logics. In: Robinson, A., Voronkov, A. (Eds.), Handbook of Automated Reasoning. Elsevier Science Publishers (North-Holland), Amsterdam.

Cech, T.R., 1990. Self-splicing of group *i* introns. Annual Review of Biochemistry 59 (1), 543–568 URL ⟨http://arjournals.annualreviews.org/doi/abs/10.1146/ann%urev.bi.59.070190.002551⟩.

Dean, M., Connolly, D., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F., Stein, L.A., 2002. OWL web ontology language 1.0 reference. Available at ⟨http://www.w3.org/TR/owl-ref/⟩.

Eiter, T., Lukasiewicz, T., Schindlauer, R., Tompits, H., 2004. Combining answer set programming with description logics for the semantic web. In: Proceedings of the Ninth International Conference on Principles of Knowledge Representation and Reasoning (KR 2004).

Evren Sirin, B.P., 2004. Pellet: an OWL DL reasoner. In: Volker Haaslev, R.M. (Ed.), Proceedings of the International Workshop on Description Logics (DL2004).

Gamma, E., Helm, R., Johnson, R., Vlissides, J., 1995. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional Computing Series. Addison-Wesley Publishing Company, New York, NY.

Haarslev, V., Möller, R., 2001. RACER system description. In: Proceedings of the International Joint Conference on Automated

Reasoning (IJCAR-01), Lecture Notes in Artificial Intelligence, vol. 2083. Springer, Berlin, pp. 701–705.

Holger Knublauch, A.R., Musen, M., 2004. Editing description logic ontologies with the protege-owl plugin. In: International Workshop on Description Logics—DL2004.

Horrocks, I. FacT + +., web site, ⟨http://owl.man.ac.uk/factplusplus/⟩.

Horrocks, I., Kutz, O., Sattler, U., 2005. The irresistible SRIQ. In: OWL: Experiences and Directions (Workshop), Galway, Ireland, November 11–12, 2005.

Kaminski, N., 2000. Bioinformatics. A user's perspective. American Journal of Respiratory Cell Molecular Biology 23, 705–711.

Lutz, C., 2003. Description logics with concrete domains—a survey. In: Advances in Modal Logics, vol. 4. World Scientific Publishing Co. Pte. Ltd., Singapore.

Motik, B., Sattler, U., Studer, R., 2004. Query answering for owl-dl with rules. In: Proceedings of the Third International Semantic Web Conference (ISWC 2004), Lecture Notes in Computer Science, vol. 3298. Springer, Berlin.

Mulder, N.J., Apweiler, R., Attwood, T.K., Bairoch, A., Bateman, A., Binns, D., Bradley, P., Bork, P., Bucher, P., Cerutti, L., Copley, R., Courcelle, E., Das, U., Durbin, R., Fleischmann, W., Gough, J., Haft, D., Harte, N., Hulo, N., Kahn, D., Kanapin, A., Krestyaninova, M., Lonsdale, D., Lopez, R., Letunic, I., Madera, M., Maslen, J., McDowall, J., Mitchell, A., Nikolskaya, A.N., Orchard, S., Pagni, M., Ponting, C.P., Quevillon, E., Selengut, J., Sigrist, C.J., Silventoinen, V., Studholme, D.J., Vaughan, R., Wu, C.H., 2005. Interpro, progress and status in 2005. Nucleic Acids Research 33 (Database issue), D201–D205.

Pullan, M.R., Watson, M.F., Kennedy, J.B., Raguenaud, C., Hyam, R., 2000. The Prometheus taxonomic model: a practical approach to representing multiple taxonomies. Taxon 49, 55–75.

Rector, A., 2004. Defaults, context, and knowledge: alternatives for OWL-indexed knowledge bases. In: Proceedings of the Ninth Pacific Symposium on Biocomputing (PSB), pp. 226–237.

Rector, A., Drummond, N., Horridge, M., Rogers, J., Knublauch, H., Stevens, R., Wang, H., Wroe, C., 2004. OWL pizzas: practical experience of teaching OWL-DL: common errors & common patterns. In: 14th International Conference on Knowledge Engineering and Knowledge Management EKAW, pp. 63–81.

Rector, A.L., Wroe, C., Rogers, J., Roberts, A., 2001. Untangling taxonomies and relationships: personal and practical problems in loosely coupled development of large ontologies. In: K-CAP, pp. 139–146.

Ringland, G., Duce, D., 1988. Approaches to Knowledge Representation: An Introduction, Knowledge-Based and Expert Systems Series. Wiley, Chichester.

Rosati, R., On the decidability and complexity of integrating ontologies and rules. Journal of Web Semantics 3 (1), 61–73.

Steele, G.L., 1990. Common Lisp the Language, second ed. Digital Press URL ⟨http://www.cs.cmu.edu/Groups/AI/html/cltl/cltl2.ht%ml⟩.

Stevens, R., Wroe, C., Lord, P., Goble, C., 2003. Ontologies in bioinformatics. In: Staab, S., Studer, R. (Eds.), Handbook on Ontologies in Information Systems. Springer, Berlin, pp. 635–657.

The Gene Ontology Consortium, 2000. Gene ontology: tool for the unification of biology. Nature Genetics 25, 25–29.

Tsarkov, D., Horrocks, I., 2004. Efficient reasoning with range and domain constraints. In: Proceedings of the 2004 Description Logic Workshop (DL 2004), pp. 41–50.

Volker Haarslev, R.M., 2001. Racer system description. In: International Joint Conference on Automated Reasoning, IJCAR 2001.

Wolstencroft, K., McEntire, R., Stevens, R., Tabernero, L., Brass, A., 2005a. Constructing ontology-driven protein family databases. Bioinformatics 21 (8), 1685–1692.

Wolstencroft, K., Brass, A., Horrocks, I., Lord, P., Sattler, U., Stevens, R., Turi, D., 2005b. A Little Semantic Web Goes a Long Way in Biology, In Fourth International Semantic Web Conference, Vol. 3792, Galway, Ireland, pp. 786–800.

Wolstencroft, K., Lord, P., Tabernero, L., Brass, A., Stevens, R., 2006. Protein classification using ontology classification. Bioinformatics 22 (14), e530–e538 URL ⟨http://bioinformatics.oxfordjournals.org/cgi/content/ab%stract/22/14/e530⟩.

Wolstencroft, K.J., Stevens, R., Tabernero, L., Brass, A., 2005c. PhosphaBase: an ontology driven database resource for protein phosphatases. Proteins 58 (2), 290–294.

Wroe, C., Stevens, R., Goble, C., Ashburner, M., 2003. A methodology to migrate the gene ontology to a description logic environment using DAML + OIL. In: Eighth Pacific Symposium on Biocomputing (PSB), pp. 624–636. URL ⟨http://www.cs.man.ac.uk/~stevensr/papers/Wroe-PSB-03-no%-footer.doc⟩.