Detecting Influences of Ontology Design Patterns in Biomedical Ontologies

Christian Kindermann, Bijan Parsia, and Uli Sattler

University of Manchester, UK

Abstract. Ontology Design Patterns (ODP) have been proposed to facilitate ontology engineering. Despite numerous conceptual contributions for over more than a decade, there is little empirical work to support the often claimed benefits provided by ODPs. Determining ODP use from ontologies alone (without interviews or other supporting documentation) is challenging as there is no standard (or required) mechanism for stipulating the intended use of an ODP. Instead, we must rely on modelling features which are suggestive of a given ODP's influence. For the purpose of determining the prevalence of ODPs in ontologies, we developed a variety of techniques to detect these features with varying degrees of liberality. Using these techniques, we survey BioPortal with respect to well-known and publicly available repositories for ODPs. Our findings are predominantly negative. For the vast majority of ODPs we cannot find empirical evidence for their use in biomedical ontologies.

CEUR-WS.org/Vol-2373/paper-18.pdf

1 Introduction

The idea of Ontology Design Patterns (ODP) has been introduced as a means to facilitate ontology engineering [8, 11]. Generally thought of as best practises and well-proven modelling solutions, a variety of different kinds of ODPs exist [7, 8, 12, 25]. Despite conceptual contributions for more than a decade, there is very little empirical work to provide support claims of benefits provided by ODPs. Ways of determining the prevalence of ODPs in practise is a first step for evaluating ODP's impact in practise. However, recognising ODP use from ontologies alone (without interviews or other supporting documentation) is challenging as there is no (de facto) standard mechanism for stipulating the intended use of an ODP. In this paper, we take on this challenge and develop algorithmic techniques to automate the identification of a given ODP's influence.

The contributions are as follows: (i) we develop a variety of techniques to detect modelling features that are suggestive for a given ODP's influence, (ii) we design and implement an algorithm using these techniques, (iii) and we perform an empirical study on the prevalence of ODPs in biomedical ontologies.

2 Background on Ontology Design Patterns

Different frameworks for working with patterns in Ontology Engineering have been proposed [8, 9, 11, 16, 20, 22, 24, 27, 28]. Each framework is based on a different approach for capturing assumed benefits of patterns and introduces its own terminology as well as its own notation. While these different approaches bear similarities to each other in some respects, there have been no efforts towards a standardisation process. There is also no generally accepted de facto standard for working with patterns in practise.

A unifying concept for a majority of these proposals is a practical notion *pattern reuse*. Such notions often involve prefabricated components expressed in some representation formalism on the one hand, and operations to manipulate these components on the other.

Consider the following examples in which a pattern has been proposed to be reused as

- "[...] a first-order theory whose axioms are not part of the target knowledge base, but can be incorporated via renaming of their non-logical symbols [9]."
- "[*a*] distinguished ontolog[y]." The basic mechanism for its application is OWL ontology import in which pattern elements cannot be modified. Otherwise, common operations for patterns are "clone, specialisation, generalisation, composition, expansion." [20].
- "[...] an ontology fragment, including directly reusable elements (classes, properties, etc.) as well as demo-elements that would be replaced by the user's own. The directly reusable elements should typically be borrowed from upper level ontologies [27]."

Clearly, these ideas of pattern reuse are based on a set of predefined axioms that may or may not be modified. In the scope of this work, we will restrict our attention to ODPs of this kind, i.e. ODPs that are captured by a set of axioms or an OWL ontology. In the following, a set of axioms (with or without variables) given as part of an ODP that is meant for some notion of reuse, will be referred to as a *reusable component* of the ODP. ODPs with reusable components have been the focus of the academic literature for over a decade and are commonly classified into two types: *Content Ontology Design Patterns* (CODP) and *Logical Ontology Design Patterns* (LODP).

CODPs are motivated as conceptual modelling solutions featuring a domain dependant signature, possibly extracted from Upper Level Ontologies to be applicable across different domains [20]. For example, the following axioms have been proposed as part of the `AgentRole` pattern which aims to provide a generic modelling solution for capturing role relationships [13]:

$$\begin{aligned}
\mathsf{Role} &\sqsubseteq \exists \mathsf{hasTemporalExtent}.\mathsf{TemporalExtent} \\
\mathsf{AgentRole} &\sqsubseteq \mathsf{Role} \\
\exists \mathsf{rolePerformedBy}.\mathsf{Agent} &\sqsubseteq \mathsf{AgentRole} \\
\mathsf{rolePerformedBy} &\equiv \mathsf{performsRole}^{-}
\end{aligned}$$

LODPs on the other hand are motivated as structural modelling solutions that are domain-independent [11, 21]. They are characterised by a set of axioms containing variables that are to be replaced as needed in the context of some use case. For example, the LODP `Partition` describes how a concept can be divided into distinct, non-overlapping, but covering subconcepts:

$$S \equiv P_1 \sqcup \ldots \sqcup P_n \tag{1}$$

$$P_i \sqcap P_j \sqsubseteq \bot \quad (\text{for all } i, j \leq n \wedge i \neq j) \tag{2}$$

Here, $S, P_1, \ldots, P_n$ are understood as variables for concepts that need to be replaced. The variable concept S is the divided into covering parts $P_1, \ldots, P_n$ (see Axiom 1). The parts $P_1, \ldots, P_n$ are made non-overlapping by pairwise disjointness constraints (see Axioms 2).

3 Pattern Detection

The lack of a generally agreed upon notion for ODP reuse poses a challenge for determining whether an ODP has in fact informed the design of a given ontology. Different approaches for ODP reuse result in different modelling features suggestive for a given ODP's influence. Therefore, we must design a detection mechanism that accounts for this uncertainty.

In the scope of this work, we limit our investigation to approaches that are based on ODPs documented with reusable components (cf. Section 2). Furthermore, we assume these components to be given in the form of ontologies or more generally sets of axioms. Given such a component $\mathcal{P}$, the problem of detecting modelling features which are suggestive of the ODP's influence in a given ontology $\mathcal{O}$ can be reduced to detecting features of $\mathcal{P}$ shared with $\mathcal{O}$. In the following, we formulate a list of non-exhaustive criteria that may be used to determine shared features between $\mathcal{P}$ and $\mathcal{O}$.

3.1 Detection Techniques

One of the earliest approaches for reusing an ODP's $\mathcal{P}$ proposes to use ontology imports as the basic mechanism for reuse [20]. This approach has been adopted by the NeOn project [21], in the context of which a large amount of work has been carried out that promulgated into the academic literature.

Import containment Detecting whether a given $\mathcal{P}$ of some ODP has been imported in an ontology $\mathcal{O}$ comes down to a straightforward analysis of $\mathcal{O}$'s import declaration. Given our primary concern of detecting an ODP's influence without any further qualification, we will generally equate an ontology with its import closure unless stated otherwise.

The analysis of $\mathcal{O}$'s import declarations is based on the two ways an ontology may be imported. Namely, *import by name* and *import by location*. Import by name is performed by interpreting the object of an import declaration as the name of an ontology in a predefined list of ontology repositories. If the object of an import declaration can be matched with the name of an ontology in said repositories, then the ontology is imported. Contrary, import by location is performed by interpreting the object of an import declaration as a physical location of an ontology. This location may be a location in the local file system.

Import by name allows for an unambiguous way to determine whether a given $\mathcal{P}$ has been imported, if its name in some ontology repository is known. Import by location on the other hand, poses a challenge due to the possibility of arbitrary renaming of local files. Nevertheless, it is reasonable to assume that the name of a local file is suggestive of its contents. Hence, it is worthwhile to consider import declarations as candidates for $\mathcal{P}$ reuse if the object of the declaration is lexically close to the respective ODP's name.

These considerations motivate an `ImportCheck` for an ODP captured as follows. First, check whether $\mathcal{P}$ is imported by name in $\mathcal{O}$ (including the import closure). If $\mathcal{P}$ is not found, we test whether the object of any import declaration in $\mathcal{O}$ is lexically similar to the ODP's name captured by $\mathcal{P}$.

Signature overlap It has been proposed to reuse a given $\mathcal{P}$ by coping its contents into a target $\mathcal{O}$ [23]. Copying any logical entities in $\mathcal{P}$ verbatim will result in syntactic traces, i.e, $\widetilde{\mathcal{P}} \cap \widetilde{\mathcal{O}} \neq \emptyset$, where $\widetilde{\mathcal{O}}$ denotes the signature of an ontology, i.e., its concept, role, and individual names. Hence, we specify an `IRICheck` that tests for all logical axioms $\alpha \in \mathcal{P}$ whether the IRI of any $e \in \widetilde{\alpha}$ occurs in $\mathcal{O}$. This occurrence test in $\mathcal{O}$ includes checking non-logical parts such as annotations and entity declarations.

In addition, we specify a `NamespaceCheck` that tests whether the longest common prefix of IRIs of many entities in $\mathcal{P}$, occurs in $\mathcal{O}$.

Lexical variation Apart from approaches to ODP reuse that preserve the IRIs of elements in $\mathcal{P}$, there are proposals allowing for the possibility of a renaming of copied elements [14]. In this case, the reuse of axioms $\alpha \in \mathcal{P}$ can be identified by some substitution $\sigma : \widetilde{\mathcal{P}} \to \widetilde{\mathcal{O}}$ such that $\sigma(\alpha) \in \mathcal{O}$. Here, we require substitutions to respect types, i.e. concepts, roles, and individuals are only mapped to concepts, roles, and individuals respectively. However, with no information expressly declaring that $\mathcal{P}$ has been reused via some σ in $\mathcal{O}$, determining whether $\mathcal{P}$ has been reused under some elusive substitution is a challenging task.

Based on the assumption that entities $p \in \widetilde{\mathcal{P}}$ exhibit lexical similarities to their mappings $\sigma(p) \in \widetilde{\mathcal{O}}$, we can generate candidate substitutions. For $p \in \widetilde{P}$, find a $p' \in \widetilde{\mathcal{O}}$ that is lexically similar. If such a p' exists for all $p \in \widetilde{\mathcal{P}}$, then we can define a substitution σ simply by $p \mapsto p'$. Given a candidate σ, we specify a `SubstitutionContainmentCheck` that tests whether $\sigma(\alpha) \in \mathcal{O}$ holds for all axioms $\alpha \in \mathcal{P}$, where $\sigma(\alpha)$ denotes an axiom in which all substitutions specified in σ have been performed.

Logical variation Besides changing the signature of an ODP's $\mathcal{P}$, there have been proposals for ODP reuse based on reimplementing aspects of $\mathcal{P}$ by analogy [10]. In this case, both the logical structure as well as the signature of axioms $\alpha \in \mathcal{P}$ may be subject to change. Based on motivations for logical rewritings of $\mathcal{P}$ [15], we specify a `SubstitutionEntailmentCheck` that tests whether there exists some substitution σ (generated as above) such that for all $\alpha \in \mathcal{P}$ it holds that $\mathcal{O} \models \sigma(\alpha)$.

Logical Axiom Agreement In addition to detection techniques searching for positive evidence of a given ODP's influence, we may also test for necessary requirements imposed by some notion of ODP reuse. If these requirements are not met by some ontology $\mathcal{O}$, then we can exclude an ODP's reuse with respect to the notion in question. For example, positive evidence for $\mathcal{P}$ under `SubstitutionContainmentCheck` requires an ontology $\mathcal{O}$ to contain structurally identical axioms to $\mathcal{P}$ since a simple renaming of entities in axioms of $\mathcal{P}$ does not affect their logical structure. Thus, if an ontology $\mathcal{O}$ does not exhibit at least as many structurally identical axioms as a given $\mathcal{P}$, then we can conclude that $\mathcal{P}$ can not have been reused by a simple renaming of its signature.

The above argument motivates an `AxiomTypeCheck` that tests whether a given ontology $\mathcal{O}$ contains at least as many axioms of a given type as $\mathcal{P}$. The OWL 2 language specification distinguishes between three categories of axiom types: class expression axioms, object property axioms, and data property axioms. Each category defines a number of axiom types, e.g., subclass axioms, inverse object properties, or disjoint data properties [6].

Logical Expression Agreement Orthogonal to a structural agreement in terms of axioms, we can also specify a structural `ExpressionCheck` that tests whether certain logical constructs or combination of logical constructs occurring in a given $\mathcal{P}$ are present in an ontology. For example, if a logical constructor, e.g concept disjunction $\sqcup$, occurs in some expression used in $\mathcal{P}$, but there is no such expression in $\mathcal{O}$ (as is often the case for biomedical ontologies conforming to the EL profile), then certain notions of reusing $\mathcal{P}$ can be ruled out.

In the context of this work, we specify expression checks for two logical structures that seem to be crucial for a fair number of ODPs and LODPs in particular. These structures are disjoint unions on the one hand, and n-ary relationships on the other hand. We test for the presence of disjoint unions as specified in [6]. Since OWL 2 only allows for binary relationships, ODPs have been proposed to model n-ary relationships by using multiple binary relationships in combination [2]. Hence, we test for the necessary condition of a concept that is subsumed by at least two role restrictions. We refer to the checks of both of these structures as `DisjointUnionCheck` and `NAryRelationCheck` respectively.

3.2 Algorithm

Most techniques introduced in the previous section involve some form of lexical comparison between entities of $\mathcal{O}$ and $\mathcal{P}$. In order to maximise the recall of our detection mechanism, we employ a threefold procedure for establishing a lexical similarity between two strings s_1 and s_2 with an increasing degree of liberality.

The first part is a strict equality that requires all symbols occurring in s_1 to coincide with symbols in s_2 at their respective positions. The second part is a loose string match between s_1 and s_2 that removes all symbols not in the Latin alphabet, converts all characters to lower case, and tests for string containment of s_1 in s_2. The third part consists of calculating a string similarity score greater that 0.8 between two strings s_1, s_2. The similarity score is calculated by $\frac{M - LevenstheinDistance(s_1,s_2)}{M}$, where $M = \max(s_1.length, s_2.length)$.

A lexical association between two elements $e_1 \in \widetilde{\mathcal{P}}$ and $e_2 \in \widetilde{\mathcal{O}}$ is established by applying the above string comparison procedure to (1) both IRIs of e_1 and e_2, (2) both ShortFormIRIs of e_1 and e_2, (3) e_1's IRI and e_2's annotations, (4) e_1's ShortFormIRI and e_2's annotations.[1]

Using this string comparison procedure for lexical comparisons between entities of $\mathcal{O}$ and $\mathcal{P}$ in techniques presented in the previous section, we specify the following algorithm to detect influences of a given ODP exhibiting lexical modelling features.

Algorithm 1: Pattern Detection

Input : Ontology $\mathcal{O}$, Pattern $\mathcal{P}$
Output : Suggestive evidence for influence of $\mathcal{P}$ in $\mathcal{O}$

if *ImportCheck*$(\mathcal{O}, \mathcal{P})$ **then**
  return *Import declarations in $\mathcal{O}$ containing $\mathcal{P}$*
if *IRICheck*$(\mathcal{O}, \mathcal{P})$ **then**
  return *All $e \in \mathcal{O}$ that account for evidence of the check*
if *NamespaceCheck*$(\mathcal{O}, \mathcal{P})$ **then**
  return *All $e \in \mathcal{O}$ that account for evidence of the check*
if *AxiomTypeCheck*$(\mathcal{O}, \mathcal{P})$ **then**
  if *SubstitutionContainmentCheck*$(\mathcal{O}, \mathcal{P})$ **then**
    return *All σ such that $\sigma(\mathcal{P}) \in \mathcal{O}$*
  end
if *SubstitutionEntailmentCheck*$(\mathcal{O}, \mathcal{P})$ **then**
  return *All σ such that $\mathcal{O} \models \sigma(\mathcal{P})$*
end

For ODPs that only propose a set of axioms with variables to be instantiated we cannot sensibly apply Algorithm 1. Instead, the only applicaple detection techniques are the structural `AxiomTypeCheck`, `DisjointUnionCheck`, and `NAryRelationCheck`.

4 Methods

In Section 2, we have characterised the status quo of academic research around ODPs by a diversity of ideas regarding both the notion of ODPs itself and ODP reuse. This motivates an investigation of the research question as to how prevalent ODPs influences in biomedical ontologies are. In the following, we describe our procedure for answering this question.

Pattern Corpus The most well-known catalogues for ODPs are (1) the ODP Semantic Web Portal [4] and (2) the ODPs Public Catalog [3]. Both of these catalogues reflect the focus of the academic literature on CODPs and LODPs

[1] We also considered using annotations of entities $e_1 \in \widetilde{\mathcal{P}}$ to establish a lexical relationship with $e_2 \in \widetilde{\mathcal{O}}$. However, we noticed that annotations of e_1 for (alternative) labels are either equal to its ShortFormIRI or slight variations thereof. Since these variations are already captured by our string comparison procedure, we decided against using e_1's annotations to guard against spurious matches.

and contain mostly submissions for these two types. We select all ODPs from catalogues (1) and (2) that satisfy the following criteria:

(i) The pattern is categorised as either an LODP or CODP in catalogue (1).
(ii) The pattern is published together with an ontology as its reusable component or the pattern is published with an example ontology to demonstrate its reuse.
(iii) The reusable component or example ontology can be loaded and initialised with a reasoner by the OWL API.
(iv) A CODP is documented to belong to some biomedical related domain.

This selection procedure results in the selection of 47 out of 155 CODPs from (1), 4 out of 18 LODPs from (1), and all 16 ODPs from (2). Selected patterns according to criteria (iv) belong to at least one of the following domains: Agriculture, Biology, Cartography, Chemistry, Decision-making, Document Management, Earth Science or Geoscience, Ecology, Event Processing, Explanation, Fishery, General, Geology, Health-care, Management, Manufacturing, Materials Science, Organisation, Participation, Parts and Collections, Physics, Planning, Product Development, Scheduling, Software, Software Engineering, Social Science, Time, Work-flow.

Ontology Corpus We use a publicly available snapshot of BioPortal from 2017 [1]. Choosing the data set that contains all ontologies in their original state, we extract all ontologies from the archive into one folder. Any ontology that could not be loaded or handled by a reasoner in the OWL API was excluded form the study. This procedure results in the exclusion of 78 out of 438 ontologies resulting in a corpus of 360 ontologies.

Experimental Design Our empirical investigation consists of two distinct experiments.

In the first experiment, we run Algorithm 1 over all input combinations of ontologies from the ontology corpus and the 47 CODPs from catalogue (1). This experiment is designed to provide positive indiciations for influences of ODPs that exhibit lexical features, namely CODPs.

In the second experiment, we run the `AxiomTypeCheck` over all input combinations of ontologies from the ontology corpus and LODPs from catalogue (1) as well as ODPs from catalogue (2). We distinguish between two conditions: (a) including and (b) not including the imports closure of a given ODP. Lastly, we perform the `DisjointUnionCheck`, and the `NAryRelationCheck` over all ontologies from the ontology corpus. This experiment is designed to probe ontologies for necessary conditions of several notions of ODP reuse.

5 Results

5.1 Experiment 1: Detection of CODPs

Our detection mechanism generates only scant evidence for suggestive influences of CODPs. The results are summarised in Table 1 (`SubstitutionContainmentCheck` and SubstitutionEntailmentCheck are abbreviated by SContainmentCheck and SEntailmentCheck). Each row reports on the evidence generated by each subcomponent of Algorithm 1. In the following, we will provide further details on these results with respect to the used techniques (1)-(5).

Table 1. Summary of generated evidence for CODPs

Detection Technique	Number of Patterns	Number of Ontologies
(1) ImportCheck	5	6
(2) IRICheck	1	1
(3) NamespaceCheck	3	5
(4) SContainmentCheck	11	46
(5) SEntailmentCheck	0	0

(1) The ImportCheck detects only one pattern that was undisputedly reused by import, namely the AgentRole pattern. Interestingly, this reuse by import was only detected due to AgentRole being both in the corpus of patterns and ontologies. Since each ontology is contained in its own import closure, the detection of AgentRole is as expected. Otherwise, the ImportCheck only generates candidates for ODP reuse via import by location on the basis of lexical association. For example, the pattern Region was generated as candidate in the "Ontology of Geographical Region" since it contained the ontology "http://www.owl-ontologies.com/GeographicalRegion.owl" in its import closure.

(2) The detected reuse of an ODP's IRIs by the IRICheck are unsurprisingly owed to the presence of the AgentRole pattern.

(3) The NamespaceCheck performed with "http://ontologydesignpatterns.org" results in the detection of 5 entities in 3 different ontologies. In all cases, a "seeAlso" annotation reference web pages related to ODPs. For example, the object property "part of" in the "human interaction network ontology" has been annotated with "rdfs:SeeAlso <http://ontologydesignpatterns.org/wiki/Submissions:PartOf>".

(4) The SubstitutionContainmentCheck generated candidate substitutions for 11 patterns in 46 ontologies. Two out of the ODPs account solely for 26 of the 46 ontologies in which substitutions could be generated. These two ODPS are TypesOfEntities and GOTop.

(5) The SubstitutionEntailmentCheck did not result in the generation of additional candidate substitutions.

5.2 Experiment 2: Testing necessary conditions

The AxiomTypeCheck shows for both conditions (a) and (b) that 75% of ontologies do not exhibit the necessary number of different axioms types occurring in ODPs from catalogue (1). Similarly, 80% of ontologies do not contain the necessary number of different axioms types for the majority (37 out of 47) of ODPs from catalogue (2).

However, the NAryRelationCheck reveals that nearly half (168/360) of all ontologies in the corpus fulfil the tested necessary for the existence of an n-ary relationship. Lastly, the `DisjointUnionCheck` finds evidence for disjoint unions in 24 ontologies.

6 Discussion

The results of our investigation provide very little support for influences of ODPs in biomedical ontologies. The negative results of our `ImportCheck` show that a given ODP's component $\mathcal{P}$ is not reused in practise as originally envisioned by the NeOn project. Furthermore, the negative results of our `IRICheck` indicate that even parts of reusable components $\mathcal{P}$ do not directly influence the ontology engineering tasks in practise.

Even though we could not find explicit evidence for any ODP being reused by import, we did find evidence by the mere presence of the `AgentRole` pattern in the corpus of ontologies. Through manual inspection of the original data set for the used BioPortal snapshot, we notice that the `AgentRole` pattern is located in an archive file for the ontology ICPS. This archive also contains another pattern, namely `Person`. However, the ontology ICPS has been excluded during the process of the ontology corpus construction for the study. This observation raises the question whether our results are skewed by our ontology exclusion criteria for constructing the ontology corpus. We can invalidate this concern due to the following. First, we acquire a version of the used BioPortal snapshot in which each ontology has been merged with its import closure. Then, we treat all ontologies as simple text files and performed another `NamespaceCheck`. Still, there is no positive finding to be reported.

Inspecting the positive evidence found by the `NamespaceCheck`, it is quite clear that practitioners create their own entities instead of reusing IRIs from ODPs directly. Nevertheless, it remains unclear whether this is owed to a conscious modelling decision, mere personal preference, lack of know-how, or lack of tool support for ODPs.

Yet, there is a caveat with respect to reusing IRIs from ODPs. Some ODPs published on `http://ontologydesignpatterns.org` are said to be "extracted from upper level ontologies". However, interestingly, their respective reusable components $\mathcal{P}$ are often self-contained ontologies not bearing any relation to upper level ontologies. This suggests that $\mathcal{P}$ is a somehow reimplemented fragment of the upper level ontology. Clearly, this gets practitioners into the predicament of choosing between aligning their ontologies to an upper level ontology or an ODP (if they are so inclined in the first place). Hence, it is possible that practitioners prefer to work with the original upper level ontology rather than the extracted ODPs thereof.

Irrespective of any matter of renaming, the findings of our `AxiomTypeCheck` suggest that modelling features exhibited by most reusable components of ODPs are not highly prevalent in ontologies of the biomedical domain. It has been noted before that an ODP's required language expressivity is outside of the popular EL profile many biomedical ontologies conform to [15]. Moreover, it seems that a fair amount of published ODPs seem to propose property centric modelling approaches whereas ontologies in the biomedical domain follow a concept centric design.

Since a high percentage of ontologies do not contain at least the same number of axioms or axioms types of a given ODP, it is unsurprising to find a limited number of candidates under the `SubstitutionContainmentCheck`. Likewise, it is equally unsurprising to find a limited number of candidates under the `SubstitutionEntailmentCheck` because a fair number of ODPs make use of modelling techniques that are not expressible in the EL profile to which a lot of biomedical ontologies conform.

Given the above observation with respect to axiom types and differences in language requirements, we considered to relax the conditions of our substitution checks. Instead of requiring a substitution for *all* axioms $\alpha \in \mathcal{P}$, we only require a substitution for some subset $S \subseteq \mathcal{P}$ such that $\sigma(\alpha) \in \mathcal{O}$ holds for all $\alpha \in S$. Essentially, this corresponds to some notion of a *partial* reuse of $\mathcal{P}$. Allowing for arbitrary subsets $S \subseteq \mathcal{P}$ results in the generation of a large amount of spurious data due to our liberal lexical association procedure. Imposing some lower bound on the size of S is not straightforward as an ODP's $\mathcal{P}$ is often quite small to begin with. Limiting the search space for lexical associations in the target ontology $\mathcal{O}$ by some heuristics seems to be the most promising approach. For example, given a match between some $e \in \widetilde{\mathcal{P}}$ and $e' \in \widetilde{\mathcal{O}}$, limit the search for further lexical associations of elements in $\widetilde{\mathcal{P}}$ to the set $\{\alpha \in \mathcal{O} \mid e' \in \widetilde{\alpha}\}$ and proceed recursively. However, slight variations in heuristic search strategies result in drastic effects for the number of generated lexical associations. Overall, generating meaningful data for partial reuse of a given ODP's $\mathcal{P}$ turns out to be a non-trivial research endeavour in and of itself.

6.1 Limitations

Despite our intention to maximise the recall of our detection mechanism, there are a few limitations. Some patterns in our corpus are not intended to be directly reused via some reusable component $\mathcal{P}$. The ODP `UpperLevelOntology` [5] is such an example. This pattern motivates to align a given ontology to a chosen upper level ontology. Since all our detection techniques are agnostic to influences of upper level ontologies and only target lexical as well as structural modelling features, the prevalence of ontologies aligned to upper level ontologies is not determined and our negative results are inconclusive.

Another limitation is the manner in which we try to establish lexical associations between entities of ODPs and entities of domain ontologies. Entities of ODPs are arguably of general nature and might not easily associated with domain specific entities on a purely lexical basis. Instead, one might need to consider lexical relationships based on hyponyms and hypernyms. However, doing so would require an overall sophisticated lexical matching procedure to prevent spurious associations.

6.2 Related Work

Ontology enrichment has motivated one of the first attempts to automatically identify the use of ODPs in ontologies [19]. Here, it is argued that the identification of partial instantiations of ODPs may allow for ontology refinement by completing the missing parts of a pattern. The proposed detection mechanism heavily depends on a lexical association procedure that is based on a number of heuristics. However, a large scale evaluation of the proposed techniques is left for future work.

WordNet has been considered to provide background knowledge for establishing lexical associations between entities of ODPs and domain ontologies [17]. A detection mechanism primarily based on lexical string matching is proposed and a first empirical evaluation is performed. Contrary to our findings, a large number of frequently detected ODPs is reported. However, the produced results are described as "probably not reliable". The background knowledge from WordNet seems to produce spurious results and skews the data towards patterns including a certain signature.

Query languages such as SPARQL and OPPL have been considered to probe an ontology for structural aspects of ODPs. Work on detection mechanisms to combine both lexical and structural aspects of ODPs is still preliminary [26]. However, a large scale study using a structural detection mechanism that deliberately disregards notions of ODP reuse under lexical variations has been conducted [18]. A small number of structurally simple ODPs are reported to be reused in biomedical ontologies. Otherwise, little evidence for ODPs reuse in biomedical ontologies is found.

7 Conclusion

The results of our empirical evaluation corroborate the findings of previous studies to the extent that there is only scant evidence for influences of ODPs in the biomedical domain [18]. Even liberal notions for ODP reuse which can only be considered suggestive of a given ODP's influence do not allow for a different conclusion. While this negative finding appears unconstructive, we will qualify its implications in light of the nature of our chosen detection techniques.

The results of our `AxiomTypeCheck` and `DisjointUnionCheck` indicate that modelling solutions proposed by ODPs differ significantly compared with ontologies authored by practitioners in the biomedical domain. The design of most biomedical ontologies are concept centric and do not contain a lot of disjoint unions, whereas the design of ODPs published in catalogues (1) and (2) place an emphasize on roles and disjoint unions respectively. The lack of evidence under the `IRICheck` also shows that ODPs are not partially reused by omission of unwanted axioms. Moreover, even in cases in which there is some evidence that practitioners in the biomedical domain are aware of ODPs, they seem to limit the reuse of ODPs to the realm of annotations as the data collected by our `NamespaceCheck` suggests.

Overall, it seems that currently, ODPs do not provide solutions for *common* ontology design tasks in the biomedical domain.

These findings can serve as a motivation for a data driven approach to automatically generate or at least inform the development of practically relevant ODPs. In such a scenario, detection techniques such as the ones presented in this paper can serve as some kind of quality measure. The desire for such work has already been expressed [13].

References

1. BioPortal Snapshot 30.03.2017. https://zenodo.org/record/439510#.XKK-Nt-YVhE (Accessed on 30.04.2018).
2. Nary Relationship pattern. http://odps.sourceforge.net/odp/html/Nary_Relationship.html (Accessed on 24.05.2019).
3. Ontology Design Patterns (ODPs) Public Catalog. http://odps.sourceforge.net/odp/html/index.html (Accessed on 24.05.2019).
4. Semantic Web portal dedicated to ontology design patterns. http://ontologydesignpatterns.org (Accessed on 24.05.2019).
5. Upper Level Ontology Pattern. http://odps.sourceforge.net/odp/html/Upper_Level_Ontology.html (Accessed on 24.05.2019).
6. OWL 2 Web Ontology Language. Structural Specification and Functional-Style Syntax (Second Edition). 2012. http://www.w3.org/TR/owl2-syntax/ (Accessed on 24.05.2019).
7. Eva Blomqvist. Ontology Patterns: Typology and Experiences from Design Pattern Development. In *The Swedish AI Society Workshop May 20-21; 2010; Uppsala University*, number 048, pages 55–64. Linköping University Electronic Press, 2010.
8. Eva Blomqvist and Kurt Sandkuhl. Patterns in Ontology Engineering–Classification of Ontology Patterns. In *ICEIS 2005: proceedings of the Seventh International Conference on Enterprise Information Systems, Miami, USA, May 25-28, 2005*, 2005.
9. Peter Clark. Knowledge Patterns. In *EKAW*, volume 5268 of *Lecture Notes in Computer Science*, pages 1–3. Springer, 2008.
10. Ricardo de Almeida Falbo, Giancarlo Guizzardi, Aldo Gangemi, and Valentina Presutti. Ontology Patterns: Clarifying Concepts and Terminology. In *WOP*, volume 1188 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2013.
11. Aldo Gangemi. Ontology Pesign Patterns for Semantic Web Content. In *International Semantic Web Conference*, pages 262–276. Springer, 2005.
12. Giancarlo Guizzardi. Theoretical Foundations and Engineering Tools for Building Ontologies as Reference Conceptual Models. *Semantic Web*, 1(1-2):3–10, 2010.
13. Karl Hammar, Eva Blomqvist, David Carral, Marieke van Erp, Antske Fokkens, Aldo Gangemi, Willem Robert van Hage, Pascal Hitzler, Krzysztof Janowicz, Nazifa Karima, Adila Krisnadhi, Tom Narock, Roxane Segers, Monika Solanki, and Vojtech Svátek. Collected Research Questions Concerning Ontology Design Patterns. In *Ontology Engineering with Ontology Design Patterns*, volume 25 of *Studies on the Semantic Web*, pages 189–198. IOS Press, 2016.
14. Karl Hammar and Valentina Presutti. Template-Based Content ODP Instantiation. In *The 7th Workshop on Ontology and Semantic Web Patterns*. IOS Press, 2017.
15. Matthew Horridge, Mikel Egaña Aranguren, Jonathan Mortensen, Mark A. Musen, and Natalya Fridman Noy. Ontology Design Pattern Language Expressivity Requirements. In *WOP*, volume 929 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2012.
16. Chih-Sheng Johnson Hou, Natalya Fridman Noy, and Mark A. Musen. A Template-Based Approach Toward Acquisition of Logical Sentences. In *Intelligent Information Processing*, volume 221 of *IFIP Conference Proceedings*, pages 77–89. Kluwer, 2002.
17. Muhammad Tahir Khan and Eva Blomqvist. Ontology Design Pattern Detection-Initial Method and Usage Scenarios. In *SEMAPRO 2010, The Fourth International Conference on Advances in Semantic Processing*, pages 19–24, 2010.
18. Jonathan Mortensen, Matthew Horridge, Mark A. Musen, and Natalya Fridman Noy. Modest Use of Ontology Design Patterns in a Repository of Biomedical Ontologies. In *WOP*, volume 929 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2012.

19. Nadejda Nikitina, Sebastian Rudolph, and Sebastian Blohm. Refining Ontologies by Pattern-Based Completion. In *WOP*, volume 516 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2009.
20. Valentina Presutti and Aldo Gangemi. Content Ontology Design Patterns as Practical Building Blocks for Web Ontologies. In *International Conference on Conceptual Modeling*, pages 128–141. Springer, 2008.
21. Valentina Presutti, Aldo Gangemi, Stefano David, G. Aguado de Cea, Mari-Carmen Suárez-Figueroa, Elena Montiel-Ponsoda, and M. Poveda. D2.5.1: A Library of Ontology Design Patterns: reusable solutions for collaborative design of networked ontologies. (Available at: http://www.neon-project.org/), 2008.
22. Jacqueline Renée Reich. Onthological Design Patterns for the Integration of Molecular Biological Information. In *German Conference on Bioinformatics*, pages 156–166, 1999.
23. Fabiano Borges Ruy, Cássio Chaves Reginato, Victor Amorim dos Santos, Ricardo de Almeida Falbo, and Giancarlo Guizzardi. Ontology Engineering by Combining Ontology Patterns. In *ER*, volume 9381 of *Lecture Notes in Computer Science*, pages 173–186. Springer, 2015.
24. Steffen Staab, Michael Erdmann, and Alexander Maedche. Engineering Ontologies using Semantic Patterns. In *OIS@IJCAI*, volume 47 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2001.
25. M. C. Suárez-Figueroa, S. Brockmans, A. Gangemi, A. Gómez-Pérez, J. Lehmann, H. Lewen, V. Presutti, and M. Sabou. D 5.1.1 NeOn Modelling Components. (Available at: http://www.neon-project.org), March 2007.
26. Ondrej Sváb-Zamazal, François Scharffe, and Vojtech Svátek. Preliminary Results of Logical Ontology Pattern Detection using SPARQL and Lexical Heuristics. In *WOP*, volume 516 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2009.
27. Vojtěch Svátek. Design Patterns for Semantic Web Ontologies: Motivation and Discussion. In *In: 7 th Conf. on Business Information Systems (BIS-04)*, 2004.
28. Denny Vrandecic. Explicit knowledge engineering patterns with macros. In *Proceedings of the Ontology Patterns for the Semantic Web Workshop a the ISWC 2005*, Galway, Ireland, 2005.