Analysing Syntactic Regularities in Ontologies

Eleni Mikroyannidi, Nor Azlinayati Abdul Manaf, Luigi Iannone, Robert Stevens

University of Manchester, United Kingdom,
email:{mikroyannidi|abdulman|iannone|stevens@cs.manchester.ac.uk}

Abstract. Syntactic regularities are repetitive structures of axioms in the asserted form of an ontology. The Regularity Inspector for Ontologies (RIO) is a framework for detecting such regularities in ontologies using cluster analysis. Detection of syntactic regularities can be used to identify parts of an ontology that have a similar syntactic structure, and could therefore provide an intuition of their construction. In this paper, we introduce *uniformity* in regularities, meaning the degree of diversity of regularities in an ontology. Based on this notion, we present an analysis of syntactic regularities in a variety of ontologies by applying RIO. The selected ontologies are mainly biomedical ontologies; processable BioPortal ontologies and SKOS vocabularies that represent biomedical concepts, gathered from the Web. Our analysis aims to show how syntactic regularities are formulated when a different knowledge representation language (OWL, SKOS) is used. The results have shown that the selected SKOS vocabularies were more uniform in terms of their syntactic regularities; smaller homogeneous clusters were found, and with few generalisations, but of high abstraction level and cluster coverage. Compared to SKOS vocabularies, BioPortal ontologies were regular, but more complex and less uniform. The analysis of syntactic regularities and uniformity of regularities can be helpful for gaining an intuition of the ontology design and its complexity.

1 Introduction

Advancements in ontology engineering should lead to the adoption of more systematic methods and more advanced tools for ontology development. The construction of ontologies has become a collaborative process that is often based on patterns of different granularity. These can be conceptual schemas, general guidelines, spreadsheets for collecting knowledge and populating the ontology, and so on [11,7]. The instantiation of these patterns should give rise to repeating regularities in the use of entities and axioms. The recognition of regularities is important when authoring an ontology in order to understand it and to assure that it conforms to guidelines and agreed patterns.

A *syntactic regularity* is defined as a set of axioms with reoccurring (regular) syntactic structure. We presented RIO in [9]; a framework for detecting such regularities. A regularity can be expressed with a *generalisation*, which is an axiom that allows for variables to replace entities. For example, given the following axioms:

American *SubClassOf* hasTopping **some** TomatoTopping
LaReine *SubClassOf* hasTopping **some** HamTopping
Margherita *SubClassOf* hasTopping **some** TomatoTopping
Fiorentina *SubClassOf* hasTopping **some** SpinachTopping

Then the syntactic regularity of these axioms can be given by the following generalisation:

?Pizza *SubClassOf* hasTopping **some** ?PizzaTopping

where ?Pizza, ?PizzaTopping are variables holding the corresponding similar entities. Such a framework can be used when authoring an ontology, in order to pinpoint repetitive information.

We distinguish the notion of syntactic regularity from the notion of *pattern*. Patterns are used in the literature with multiple meanings. In ontology engineering, they can be interpreted as design patterns, meaning solutions to design problems [4,2,3]. Patterns of axioms, however, can exist throughout an ontology without being an accepted design pattern. In general, we regard patterns as the modelling templates that the ontology engineer follows when developing the ontology. Patterns can be represented with different forms, such as conceptual models, OPPL scripts[1] scripts [5,6], text descriptions etc [10] etc. The correct use of patterns will produce syntactic regularities, i.e., axioms of similar syntax [9]; on the other hand, a syntactic regularity does not necessarily coincide with a pattern. However, the recognition of syntactic regularities should be helpful in understanding the composition of the ontology, as it can reveal parts of the ontology that were designed in similar ways. This should enable the user to complete tasks, such as extension of the ontology, its integration with other ontologies, quality assurance and so on.

In addition, patterns can be represented as an aggregation of generalisations. For example, the pattern that describes all pizzas is:

?Pizza *SubClassOf* hasTopping **some** ?PizzaTopping
?Pizza *SubClassOf* hasTopping **only** ?PizzaTopping
DisjointClasses:'set(?Pizza.VALUES)'

A pattern can be represented not only by a single generalisation but, also as a set of generalisations.

In this paper, we also define the notion of *uniformity* in syntactic regularities. The level of uniformity in regularities is an indication of the diversity or degree of regularity of an ontology. An ontology can be regular, but not uniform, meaning that there are

[1] http://oppl2.sourceforge.net

many different types of regularities, which cover significant parts of the ontology. This may also give an intuition for the compositional complexity of an ontology. It should show whether different design decisions were taken for describing different portions of the ontology. On the other hand, an ontology with very low uniformity and instantiation coverage of its regularities is an indication of an irregular ontology. We define metrics based on the RIO framework for measuring uniformity of regularities in ontologies.

In addition, we perform an analysis of the syntactic regularities of the BioPortal corpus and of selected SKOS vocabularies. Our comparison focuses on level, uniformity and impact of the syntactic regularities. The results have shown that many SKOS vocabularies have syntactic regularities which reflect patterns on predicates such as annotations and object properties.

However, the resulting clusters are few in number (on average, only four clusters per ontology were found), and they are homogeneous with few generalisations, but with high abstraction level and cluster coverage. Compared to SKOS vocabularies, BioPortal ontologies were regular but more complex, with more clusters and less uniformity. The analysis of syntactic regularities and uniformity of regularities can be helpful for gaining an intuition of the ontology design and its complexity. A regular ontology with low uniformity can indicate the existence of a general pattern, with a few variations. Such an analysis should be useful when authoring or extending an ontology.

2 RIO framework

The RIO [9] framework spots syntactic regularities in ontologies using cluster analysis. RIO enables the partitioning of a set of entities in an ontology according to a similar usage in the axioms of the ontology.

The detection of syntactic regularities is based on the following two general steps:

1. Computation of clusters of similar entities in the ontology.
2. Provision of a synthetic view of all the axioms that contribute to the generation of an entity cluster.

The purpose of cluster analysis is to partition data into groups (clusters) that are meaningful, useful, or both [12]. In the second step, the description of the cluster is shown with *generalisations*, which are axioms with entities represented by variables. The variables represent the corresponding clusters of similar entities and the syntactic regularities in the ontology are expressed with the generalisations.

2.1 Clustering

Algorithm 1 shows the steps that are followed for the computation of clusters in an ontology. The role of the placeholder replacement function is described later in this section. The HIERARCHICAL($M_{i,j}, P$) is the function that performs hierarchical agglomerative clustering and has as parameters the generated proximity matrix $M_{i,j}$ and a stopping criterion P. Details on how the hierarchical agglomerative clustering works can be found in [12].

Algorithm 1 RIO `Clustering`

Require: A placeholder replacement function ϕ, J the set of axioms in $\mathcal{O}$.
Ensure: A set of clusters S.

1: $\Sigma \leftarrow Sig(J)$
2: $M_{i,j}, 0 \leq i, j < |\Sigma|$
3: **for all** $(\sigma_i,\sigma_j) \in \Sigma \text{x} \Sigma$ **do**
4: Get axioms $\mathsf{Ax}(\sigma_i)$, $\mathsf{Ax}(\sigma_j) \in J$
5: $\mathsf{A}_i \leftarrow \phi(\mathsf{Ax}(\sigma_i))$,$\mathsf{A}_j \leftarrow \phi(\mathsf{Ax}(\sigma_j))$ ▷ Transform axioms
6: $d(\sigma_i, \sigma_j) \leftarrow \frac{|\mathsf{A}_i \cup \mathsf{A}_j| - |\mathsf{A}_i \cap \mathsf{A}_j|}{|\mathsf{A}_i \cup \mathsf{A}_j|}$ ▷ Calculate distance
7: $m_{i,j} \leftarrow d(\sigma_i, \sigma_j)$ ▷ Build proximity matrix
8: **end for**
9: $S \leftarrow \textsc{Hierarchical}(M_{i,j}, P)$ ▷ P: Stopping criterion
10: **return** S

The stopping criterion P is selected according to the minimal or maximal differences between pairs of entities whose distance is computed. As defined in step 6 of Algorithm 1, the value of $d(\sigma_i, \sigma_j)$ is in the interval [0,1]. We select $P(1)$, thus the algorithm will stop agglomerations when the distances between all possible pairs of elements for all clusters is greater than 1.

2.2 Placeholder replacement policy

Placeholder replacement policy. The axioms in step 5 of algorithm 1 are transformed into more abstract forms using a placeholder replacement function ϕ, which is based on a heuristic approach. It enables comparison between pairs of entities and control of the distance granularity. The placeholder replacement policy used by ϕ defines when an entity should be replaced by a placeholder.

More formally, given an ontology $\mathcal{O}$, we define Φ={ ?owlClass, ?owlObjectProperty, ?owlDataProperty, ?owlAnnotationProperty, ?owlIndividual, ?*} a set of six symbols that do not appear in the signature[2] of $\mathcal{O}$ - $\mathsf{sig}(\mathcal{O})$. A placeholder replacement is a function $\phi : \mathsf{sig}(\mathcal{O}) \rightarrow \mathsf{sig}(\mathcal{O}) \cup \Phi$ satisfying the following constraints: Consider an entity $e \in \mathcal{O}$ then $\phi(e) =$

- e or ?* or ?owlClass if e is a class name;
- e or ?* or ?owlObjectProperty if e is an object property name;
- e or ?* or ?owlDataProperty if e is a data property name;
- e or ?* or ?owlAnnotationProperty if e is an annotation property name;
- e or ?* or ?owlIndividual if e is an individual property name.

In previous work we have demonstrated the usage of different replacement policies [9,8]. In this paper, we will use a replacement policy that is based on the popularity of the entities in axioms [9].

[2] For signature here we mean the set of class names, data/object/annotation property names, individuals referenced in the axioms of an ontology $\mathcal{O}$.

For example, given the following axioms from the AminoAcid[3] ontology

$$\alpha_1 = \textsf{A } \textsf{\textit{SubClassOf}} \textsf{ hasSize } \textsf{\textbf{some}} \textsf{ Tiny}$$
$$\alpha_2 = \textsf{A } \textsf{\textit{SubClassOf}} \textsf{ hasPolarity } \textsf{\textbf{some}} \textsf{ Non-polar}$$
$$\alpha_3 = \textsf{A } \textsf{\textit{SubClassOf}} \textsf{ hasCharge } \textsf{\textbf{some}} \textsf{ Neutral}$$
$$\alpha_4 = \textsf{C } \textsf{\textit{SubClassOf}} \textsf{ hasSize } \textsf{\textbf{some}} \textsf{ Small}$$
$$\alpha_5 = \textsf{C } \textsf{\textit{SubClassOf}} \textsf{ hasPolarity } \textsf{\textbf{some}} \textsf{ Polar}$$
$$\alpha_6 = \textsf{C } \textsf{\textit{SubClassOf}} \textsf{ hasCharge } \textsf{\textbf{some}} \textsf{ Neutral}$$

For calculating the distance $d(\mathsf{A}, \mathsf{C})$ the axioms are transformed as:

$$A_1 = \textsf{?* } \textsf{\textit{SubClassOf}} \textsf{ hasSize } \textsf{\textbf{some}} \textsf{ ?owlClass}$$
$$A_2 = \textsf{?* } \textsf{\textit{SubClassOf}} \textsf{ hasPolarity } \textsf{\textbf{some}} \textsf{ Non-polar}$$
$$A_3 = \textsf{?* } \textsf{\textit{SubClassOf}} \textsf{ hasCharge } \textsf{\textbf{some}} \textsf{ Neutral}$$
$$A_4 = \textsf{?* } \textsf{\textit{SubClassOf}} \textsf{ hasSize } \textsf{\textbf{some}} \textsf{ ?owlClass}$$
$$A_5 = \textsf{?* } \textsf{\textit{SubClassOf}} \textsf{ hasPolarity } \textsf{\textbf{some}} \textsf{ Polar}$$
$$A_6 = \textsf{?* } \textsf{\textit{SubClassOf}} \textsf{ hasCharge } \textsf{\textbf{some}} \textsf{ Neutral}$$

The transformation is done according to the popularity replacement policy (e.g. Neutral and Polar classes are popular, hence are not replaced by a placeholder). Entities A, C have two axioms in common ($A_1 = A_4, A_3 = A_6$), thus, according to Algorithm 1, step 6, $d(\mathsf{A}, \mathsf{C})$=(4-2)/4=0.5.

2.3 Generalisations

Algorithm 1 will return a set of clusters, whose description is given by *generalisations*. Generalisations provide a synthetic view of all the axioms that contribute to generate a cluster of entities. They also express the detected semantic regularities in the ontology. Each of these axioms can be regarded as an *instantiation* of a generalisation, as they can be obtained by replacing each variable in the generalisation with entities in the signature of the ontology. The syntax for the variables is borrowed from OPPL[4], a declarative language for manipulating OWL ontologies [5].

For example, RIO will produce for 14 clusters for the AminoAcid ontology, which will include a cluster with all the Amino Acids (20 classes), and smaller clusters including the physicochemical properties and different types of Amino Acids (e.g. PolarAminoAcid, Non-PolarAminoAcid).

[3] `http://www.co-ode.org/ontologies/amino-acid/`
[4] `http://oppl2.sourceforge.net`

2.4 Grouping the generalisations

We provide a more synthetic view of the generalisations, by grouping generalisations of similar structure and representing them with generalisations of higher abstraction. For the grouping of the generalisations, the following parameters in the generalisations are considered:

- Similar structure of generalisations
- Position of the representative cluster variable in generalisations of similar structure

The super generalisations that are created have variables whose values can be also variables of more fine-grained generalisations. In addition, the name of the variable is selected according to the commonalities of the entities that it holds. The name of the variables is selected to be the least common subsumer of the values that are covered. If this is the top entity (owl:Thing), then a general placeholder will be selected. For example,

$$hg = \text{?hematologic\_evaluation } \textit{SubClassOf} \text{ ?cluster}_{13} \; \textbf{only} \text{ ?unit\_of\_measurement}$$
$$g_1 = \text{platelet\_function\_analyzer\_100 } \textit{SubClassOf} \text{ ?cluster}_{13} \; \textbf{only} \text{ ?unit\_of\_measurement}$$
$$g_2 = \text{?hematologic\_evaluation } \textit{SubClassOf} \text{ ?cluster}_{13} \; \textbf{only} \text{ ?unit\_of\_measurement}$$

generalisations g_1, g_2 are folded under super generalisation hg.

2.5 Measuring regularity

We define the following metrics for measuring the level of regularity in an ontology:

Definition 1 (Mean Generalisations per Cluster (MG)). $MG = \sum_{i=0}^{n} \frac{g_i}{N}$, *where* g_i *is a generalisation and* N *is the number of detected clusters. It is a measure intended to show the level of abstraction for each generalisation.*

Definition 2 (Mean Instantiations per Generalisation (MI)). $MI = \sum_{i=0}^{n} \frac{a_i^{g_i}}{g_N^i}$, *where* a_i *is an axiom (instantiation) covered by a generalisation* g_i *and* g_N^i *is the total number of generalisations in cluster* i*. It is a measure intended to show the level of abstraction for each generalisation.*

Definition 3 (Total Mean Cluster Coverage per generalisation (TMCC)). *Given a set of clusters* c_N *of size* N*, in which every cluster* c_i *holds* e_n *entities described by* g_n *generalisations, if each generalisation* g_i *covers* e_m *entities in the cluster, then* $TMCC = \sum_{i=0}^{n} \frac{e_m^{g_i}}{e_n g_n N}$.

The union of the generalisations describes the cluster, hence a single generalisation might not be necessarily applicable for all the values in a cluster. Thus, TMCC measures the number of values in a cluster for which a generalisation is applicable.

Definition 4 (Cluster homogeneity). *Given a cluster c_i, the homogeneity h is defined as $h = 1 - Mean\ Internal\ Distance$. It is a measure that assesses how well formed the clusters are.*

2.6 Uniformity in regularities

The main question about detection of regularities is which strategy captures regularities, if they exist, in the most efficient way. Then a second question that arises is what is considered as an efficient way. An ontology can be one of the followings in terms of regularities:

- It can be irregular
- It can be regular with many different forms of regularities
- It can be regular with a few different forms of regularities

According to the RIO framework, we can characterise an ontology as irregular if all the generalisations in the ontology cover only single axioms. This is an extreme case, which is unlikely to happen for medium to large size ontologies. The reason is that the axioms are constructed following a syntax, thus they are expected to have some syntactic regularities.

We define *uniformity* in regularities as the *degree of diversity of regularities* in an ontology. The intuition behind uniformity is to define a characteristic that allows the assessment of the detected regularities. According to the previous states, an ontology can be regular, but have different forms of regularities; thus it has low uniformity. Uniformity can give an intuition of the composition complexity of an ontology. It shows that different design decisions were taken for describing different portions of the ontology. For example, a general pattern that has been chosen to describe a set of entities in the ontology, can have some deviations when it is applied in a different set of entities. On the other hand, an ontology can be regular with a high level of uniformity, meaning that the same form of regularity appears in most axioms of the ontology. On a second level, this reveals a low compositional complexity of the ontology.

According to the RIO framework, we can observe the following properties in order to assess the uniformity of an ontology:

- Number of generalisations
- Number of instantiations per generalisation
- Cluster coverage by generalisations
- Degree of homogeneity of clusters

An indicator of a regular and homogeneous ontology is the number of generalisations covering a high number of instantiations: the higher the number of instantiations covered by a generalisation, the more regular the ontology is; a side effect is that the number of such generalisations must be small. As a consequence, the cluster coverage by per generalisation will be close to 1. On the other hand, the existence of a high number of generalisations of similar structures gives the indication of a regular but not very uniform ontology. This can be also assessed by the number of grouped generalisations.

The high number of generalisations of similar structure having only a few differences in the variables, indicates the existence of a regularity with many deviations.

There can be two explanations for the deviations. The first one is that the distance approach which is selected does not capture similarities between entities in the most efficient way but the detected regularities is an approximation. The second one is that even though entities share common axioms, in which they play similar roles, their design deviates in other axioms.

3 Results

We applied RIO to 86 ontologies from BioPortal and in 76 SKOS vocabularies collected from the web [1]. The selected ontologies are processable and their clusters can be computed in less than two minutes. All the results can be found online[5]. Here, we will refer only to some interesting cases.

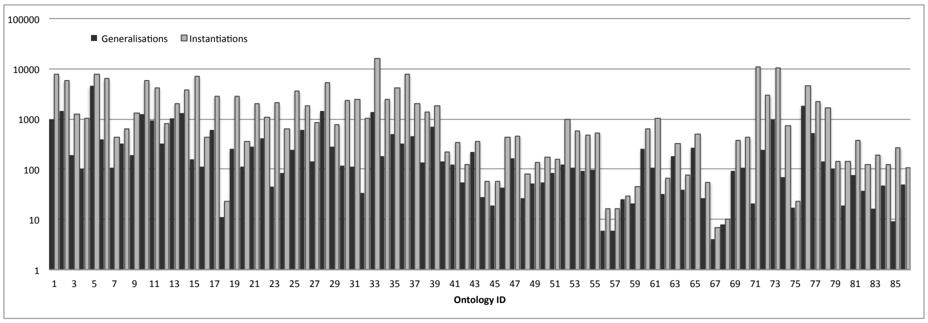
Figure 1(a) shows some selected clustering results that could give an intuition of the uniformity of regularities in the BioPortal corpus. The mean number of clusters for each ontology is 28, with a mean size of 10 entities per cluster. Mean generalisations per cluster is 12 with a minimum of 2 generalisations and maximum 125 generalisations. Mean cluster coverage (MCC) is 9.8%. The mean instantiations per generalisation for the corpus is 12 axioms, with a min value of two axioms per generalisation and a max value of 524 axioms (Ontology 72: human-developmental-anatomy-abstract.owl). This ontology is an example of one with a high regularity and uniformity. It consists of 6 clusters in total. The first cluster includes 1512 classes whose description is abstracted by two generalisations. The first refers to an annotation label and the second one is shown in Figure 2, which abstracts 1528 axioms.

The benefit from such uniform and regular design is that in an inspection task, the ontology engineer can inspect very few regularities and have an intuition of how the ontology is constructed. In addition, this can be an intuition of the ontology complexity; the ontology is based on very few patterns and constructs, thus the understanding of these patterns covers the construction of most of the ontology.

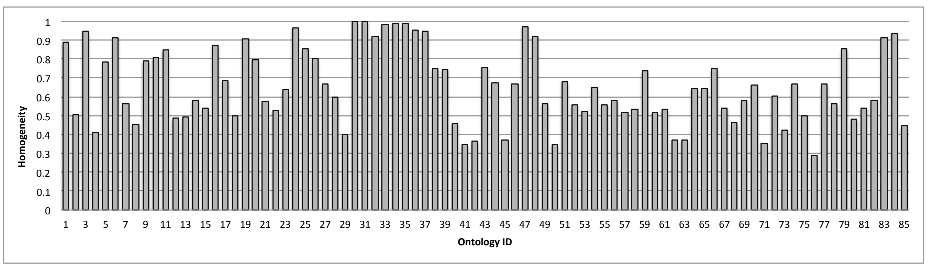
Also, Figure 1(b) shows that many ontologies that have a high level of abstraction (MI), have also a high level of homogeneity (e.g. Ontologies 32-36). This is a strong indication of a regular and uniform ontology. On the other hand, ontologies like 1, and 73 are regular but less uniform; there is a high number of instantiations (more than 1000) but there is also a quite high number of generalisations with a small mean cluster coverage (MCC=0.001%). In addition, the homogeneity of the clusters is more than 0.8, which means that the ontology is regular but not very homogeneous. This is an indication of an ontology which is regular but the regularities appear to have deviations, meaning many generalisations with similar structure. These deviations are either deliberate design or design errors. The intuition of uniformity can be helpful for also assessing the complexity of the ontology. For example, extending an ontology by following the initial design style is more difficult, since there are more than one options of regularities to select.

Figure 3 shows the corresponding clustering results for the SKOS vocabularies. On average, 4 clusters per ontology were detected, with a mean of 19 entities per cluster,

[5] `http://www.cs.man.ac.uk/~mikroyae/2012/owled/`



(a) Number of Instantiations and Generalisations



(b) Cluster homogeneity

Fig. 1. Graph showing selected clustering results for 86 BioPortal ontologies.

a mean of 6 generalisations per cluster and a mean 30 instantiations per generalisation. The mean cluster coverage (MCC) is 50.5%, which is much higher than BioPortal's corpus. Compared to the BioPortal corpus, SKOS vocabularies were regular, more uniform, with simpler and fewer generalisations, but with quite a high abstraction impact.

The detected syntactic regularities are mainly individual types and annotations. Some example syntactic regularities are shown in Figure 4 from vocabulary 47 (GeoSci-CDTGVocabularyRelation200811.rdf). The results of the detected syntactic regularities in SKOS ontologies revealed that most detected patterns refer to predicates like annotation properties and object properties.

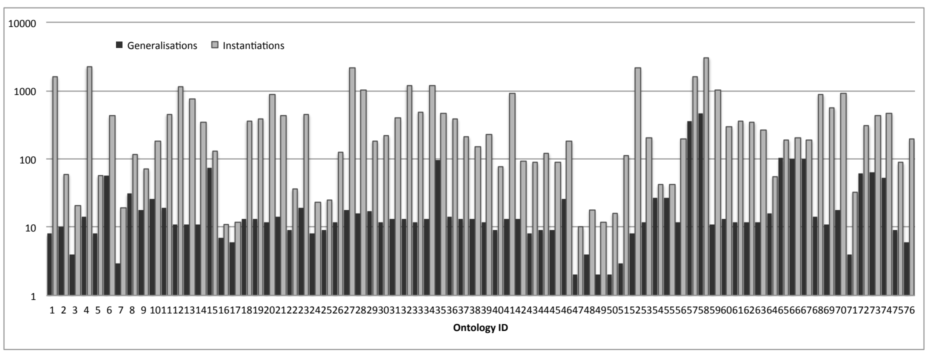
4 Conclusions

In this paper we have defined notions of uniformity in syntactic regularities, meaning the degree of diversity of regularities in an ontology. Based on this notion, we present an analysis of syntactic regularities in a variety of OWL ontologies and SKOS vocabularies by applying RIO.

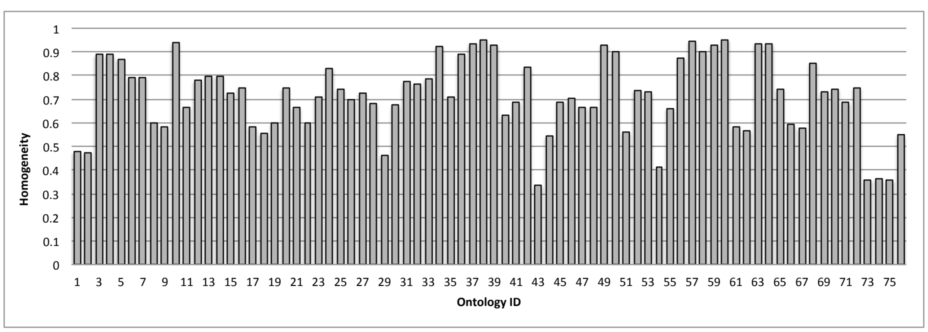
The selected targets are the processable BioPortal ontologies and SKOS vocabularies that represent biomedical concepts, gathered from the Web. The results have shown that the selected SKOS vocabularies were more uniform in terms of their syntactic regularities; smaller homogeneous clusters resulted with few generalisations, but of high

Fig. 2. Example syntactic regularity of the human-developmental-anatomy-abstract ontology with a high level of abstraction.

?cluster$_1$ *SubClassOf* HUMAN-DEV-ANAT-ABSTRACT_part_of **some** ?cluster$_2$
$Example\ Instantiation$:
EHDAA_2925 *SubClassOf* HUMAN-DEV-ANAT-ABSTRACT_part_of **some** EHDAA_2923
$where$
$\{$?cluster$_2$: CLASS=[EHDAA_2923], ?cluster$_1$: CLASS=[EHDAA_2925]$\}$



(a) Number of Instantiations and Generalisations



(b) Cluster homogeneity

Fig. 3. Graph showing selected clustering results for 76 SKOS vocabularies.

abstraction level and cluster coverage. Compared to SKOS vocabularies, BioPortal ontologies were regular but more complex and less uniform. The analysis can be used for assessing the design style of the ontologies and for quality insurance. For example, ontologies with low uniformity indicates a possibly complex ontology; the high number

Fig. 4. Example syntactic regularities of the GeoSciCDTGVocabularyRelation200811.rdf SKOS ontology

$$g_1 = \text{?cluster}_1 \text{ Type Concept}$$
$$g_2 = \text{?cluster}_1 \text{ definition "?constant"string}$$
$$g_3 = \text{?cluster}_1 \text{ historyNote "?constant"string}$$

of the generalisations of similar structure (only a variable changes in the structure of the generalisations) shows that there are deviations of the regularity, which might be intended design or design errors. Also, extending an ontology by following a regularity that has deviations (expressed with many generalisations) makes the task less straightforward since there is more than one option to select. Future work will involve the examination of alternative clustering or isomorphic approaches, since they may provide more well formed generalisations.

We have shown that RIO can highlight differences in syntactic regularity, homogeneity of clusters and the uniformity of an ontology or vocabulary written in OWL. The tooling of such notions made available in RIO offers users and authors of ontologies means by which overviews and characterisations of ontologies can be generated that could be used in quality assurance and control. The analysis of the design style and the construction of the ontology can be useful for tasks such as ontology authoring, maintenance and extension. Notions of regularity, homogeneity and uniformity in ontologies, coupled with the software adds a tool for inspecting ontologies that is not just a graph or the axioms themselves; it is an abstraction with the power to describe general properties of the ontology.

References

1. N. A. Abdul-Manaf, S. Bechhofer, and R. Stevens. The current state of SKOS vocabularies on the Web. In *Proceedings of the 9th Extended Semantic Web Conference (ESWC2012)*, May 2012.
2. M. Egaña, A. Rector, R. Stevens, and E. Antezana. *Applying Ontology Design Patterns in Bio-ontologies*, volume 5268 of *Lecture Notes in Computer Science*, pages 7–16. Springer Verlag, 2008.
3. A. Gangemi. Ontology design patterns for semantic web content. *The Semantic Web–ISWC 2005*, pages 262–276, 2005.
4. A. Gangemi and V. Presutti. Ontology design patterns. *Handbook on Ontologies*, pages 221–243, 2009.
5. L. Iannone, M. Egana, A. Rector, and R. Stevens. Augmenting the expressivity of the ontology pre-processor language. In *Proceedings of the Fifth OWLED Workshop on OWL: Experiences and Directions, OWLED*. Citeseer, 2008.

6. L. Iannone, A. Rector, and R. Stevens. Embedding knowledge patterns into OWL. *The Semantic Web: Research and Applications*, pages 218–232, 2009.
7. S. Jupp, M. Horridge, L. Iannone, J. Klein, S. Owen, J. Schanstra, K. Wolstencroft, and R. Stevens. Populous: a tool for building owl ontologies from templates. *BMC Bioinformatics*, 13(Suppl 1):S5, 2011.
8. E. Mikroyannidi, L. Iannone, R. Stevens, and A. Rector. Inspecting regularities and irregularities in SNOMED-CT. In *Proceedings of Semantic Web Applications and Tools for the Life Science 2011 (SWAT4LS)*, 2011.
9. E. Mikroyannidi, L. Iannone, R. Stevens, and A. Rector. Inspecting regularities in ontology design using clustering. *The Semantic Web–ISWC 2011*, pages 438–453, 2011.
10. B. Peters, A. Ruttenberg, J. Greenbaum, M. Courtot, R. Brinkman, P. Whetzel, D. Schober, S. Sansone, R. Scheuerman, and P. Rocca-Serra. Overcoming the ontology enrichment bottleneck with quick term templates. *Applied Ontology*, 2009.
11. S. Staab, M. Erdmann, and A. Maedche. Engineering ontologies using semantic patterns. In *Proceedings of the IJCAI-01 Workshop on E-Business & the Intelligent Web*, pages 174–185, 2001.
12. P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Addison-Wesley, 2005.