Bounds: Expressing Reservations about Incoming Data

Martin G. Skjæveland[1] and Audun Stolpe[2]

[1] Department of Informatics, University of Oslo, Norway
martige@ifi.uio.no
[2] Norwegian Defence Research Establishment (FFI)
Audun.Stolpe@ffi.no

Abstract. This paper introduces the Boundz vocabulary, an RDF vocabulary for expressing reservations about incoming data. We argue that the need for such a vocabulary is real and pressing, and that it is a useful validation tool for any recipient of RDF data that wishes to formulate restrictions on amendments in terms of the data it is already holding. The Boundz vocabulary has a simple mathematical theory that can be expressed in terms of bounded homomorphisms between RDF graphs. We present the basics of this theory, and show that bounded homomorphisms implement conservative extensions over a restricted class of ontology languages, but can also prevent cases of ontology hijacking. We additionally present a prototype implementation with promising evaluation results.

1 Introduction

Information attainable through the Web is unique, not only in terms of its scale and diversity, but also in its manner of production, being as it is characterised by collaborative accumulation of data and a lack of central authority and editorial control. This open, distributed and flat nature of the Web is often *the* essential ingredient that ensures the liveness of web data, exemplified by community curated databases such as Wikipedia, Wikidata and Freebase. Nevertheless, it does have implications for trust, data quality and interface design that may require data publishers to protect themselves from unwanted, independent third-party contributions [5]. There is, of course, no answer to which amendments that ought to be considered harmful in general. Rather, harmfulness is in the eye of the beholder and will depend upon the intended uses of a dataset and/or its associated schema; it may concern the terminology that is used to encode the data or it may concern only the data itself. The following three examples illustrate both cases.

Ontology hijacking. Ontology hijacking is the contribution of statements about classes and/or properties from a non-authoritative publisher that affects the logical properties, and thus also the reasoning, of those classes and properties. A third-party contributor could, for instance, subsume the `dcterms:subject` property from the Dublin Core vocabulary, say, under its own concept of a `ex:topic`, but would then, in the terminology of [5], be 'hijacking' `dcterms:subject`. If subsequently

reasoning were to be applied to the recipient of the data, this hijacking would result in (at least) one (extra) statement using `ex:topic` being inferred for each explicitly asserted or inferred statement using `dcterms:subject`.[3] Thus, ontology hijacking is harmful insofar as it can increase the amount of data that is inferred from the ontology of the recipient considerably. Of course, hijacking can also affect inference over data provided by other parties, parties that may be relying on the terminology of the recipient to stay fixed.

Ontology-driven faceted browsing. The idea behind faceted search is to analyse and index search items along multiple orthogonal taxonomies that are called subject *facets* [16]. From the end-users viewpoint, searching is reduced to the selection of categories along these. In *semantic* faceted search, the facets are based on ontologies and may be generated by reasoning [16]. This makes the design of an interface and the user-experience of interacting with the system vulnerable to terminological changes, whence prudence and predictiveness dictates that one does not allow just any third-party to make assertions about classes and properties in the ontology that generates the facets, even though they may be allowed to contribute instance data.

Closed topics. In recent years the concept of open government data has evolved into a febrile research area which has catalysed major public investments into data dissemination and reuse. The concept has also made its way into international law, e.g., the European Public Sector Information Directive. The access to open government data has been spearheaded by official government websites such as UK's `data.gov.uk` and its US analogue `data.gov`. There are also notable examples such as `openelectiondata.org`, which, although it is not a government initiative, has gained official endorsement. Government data often contains what may be called *closed topics*, that is, data that once it is published should not be altered or amended. Election results is a case in point. Thus, although a data hub serving government data may wish to remain distributed and collaborative, it may wish to 'seal off' certain subsets of the data while keeping others open.

In this paper we introduce an RDF vocabulary for expressing reservations about incoming data such as exemplified above. The vocabulary has an appealingly simple theory and admits an efficient implementation. We present one such implementation, together with some preliminary test results that show the feasibility of our approach. The paper is organised as follows. Section 2 recapitulates the theoretical background as set out in [13, 15], where the central concept is that of a *bounded homomorphism*. We relate bounded homomorphisms to the concept of a logical conservative extension by showing that the co-domain of a homomorphism under the weakest bound is a conservative extension of the domain, given that the homomorphism relates saturated ontologies in which each axiom is expressed as a single triple. However, we also argue as a flip side of the same coin, that bounded homomorphism in general can not be expressed by ontologies—nor need they concern terminological axioms. Even when they do concern axioms, e.g., when protecting a vocabulary against hijacking, ontologies cannot in general

[3] Consult [5] for a formal definition of ontology hijacking and an evaluation illustrating how it may have significant unintentional, hence possibly harmful, effects.

express them. In Section 3 we introduce and explain the Boundz vocabulary, and present an example of using it. We describe a prototype implementation together with some tentative evaluation results in Section 4. Section 5 contains related work, and we conclude in Section 6.

2 Theoretical Background

Let U, B and L respectively denote pairwise disjoint, fixed and infinite sets of *URIs*, *blank nodes* and *literals*. Fix $\mathcal{U} = U \cup B \cup L$ as the set of *elements*. Define the set of *(RDF) triples* as the set $\mathcal{T} = (U \cup B) \times U \times \mathcal{U}$. A triple is commonly written as a sequence of its elements, $\boldsymbol{t} = \langle s, p, o \rangle$, where s, p and o are called respectively the *subject*, *predicate* and *object* of the triple. An *(RDF) graph* G is a finite set of triples. If G is a graph, then $\mathcal{U}(G)$ is the set of elements occurring in G.

The design of the Boundz vocabulary is based on the notion of a *bounded RDF homomorphism*, which was first introduced in [15].

Definition 1 (Homomorphism). *Let G and H be graphs. A homomorphism $h : G \to H$ is a function $h : \mathcal{U}(G) \to \mathcal{U}(H)$ satisfying the condition; for all $s, p, o \in \mathcal{U}$:*

$$\langle s, p, o \rangle \in G \quad \Rightarrow \quad \langle h(s), h(p), h(o) \rangle \in H\,.$$

RDF homomorphisms, as homomorphisms, reflect the structure of the domain in the co-domain, but they do not in general reflect the structure of the co-domain back into the domain. This is evident since the co-domain of a homomorphism may be a strict superset of the image of the homomorphism, whence the co-domain is not in general addressed by the homomorphism. Nevertheless, properties of the co-domain can be expressed in terms of a homomorphism by placing restrictions on the class of homomorphisms one is willing to consider. In [15] these restrictions are called *bounds*:

Definition 2 (Bounded Homomorphism). *Let $h : G \to H$ be a homomorphism. A simple* bound *is one of following conditions; for all $s, p, o \in \mathcal{U}$:*

$$\begin{aligned}
\langle h(s), p, o \rangle \in H \quad &\Rightarrow \quad \langle s, p, o \rangle \in G && (\mathsf{S}) \\
\langle s, h(p), o \rangle \in H \quad &\Rightarrow \quad \langle s, p, o \rangle \in G && (\mathsf{P}) \\
\langle s, p, h(o) \rangle \in H \quad &\Rightarrow \quad \langle s, p, o \rangle \in G && (\mathsf{O}) \\
\langle s, p, o \rangle \in H \quad &\Rightarrow \quad \langle s, p, o \rangle \in G && (\top)
\end{aligned}$$

New bounds may be built from the simple bounds by combining them conjunctively and/or disjunctively. A bounded homomorphism *is a homomorphism that satisfies a bound. If h satisfies the bound β, we call h a β-map.*

The essential idea in [15] is to control the amendment of a dataset by interlocking two graphs in a reciprocal simulation of varying degrees of strength by combining the homomorphism condition with a bound. The two graphs in question represent

the recipient of the data before and after a contribution is made, and the relationship between the recipient and the contributor is regulated by requiring the existence of an RDF homomorphism that reflects the structure of each in the other. That is, suppose G is some community-curated data set encoded in RDF, and that H is an amendment contributed by some peer. Then the reservations that G may have about H may be expressed in terms of conditions on an RDF homomorphism h of G into $G \cup H$ that ensures that $G \cup H$ and H simulate each other to some extent deemed sufficient from the point of view of G.

In [15] it is required of a homomorphism h that it be the identity function on subjects and objects of triples. In other words, the only elements that are allowed to vary from an RDF graph to its homomorphic image are the predicates. It is an important property of this class of RDF homomorphisms that the problem of checking whether a bounded instance exists among them is in P [15, Theorem 9]. In the present paper, we shall be even more restrictive and require h to be the identity function on all elements of its domain. This yields a purely *morphological* notion of simulation where the only variations a homomorphism talks about are variations of form. We record this under the name of *bounded extension*.

Definition 3 (Bounded extension). *Let G and H be graphs. If there is bounded homomorphism $h : G \to G \cup H$, and $h(u) = u$ for all $u \in \mathcal{U}(G)$, then $G \cup H$ is a* bounded extension *of G.*

There are 19 different non-equivalent bounds for homomorphisms. If we let '$\bot$' designate 'no bound'—making a $\bot$-map an *unbounded* homomorphism—we can arrange all 19 bounds and $\bot$ in a lattice according to logical implication; this is done in Figure 1. Here, if we have $\beta_1 \leq \beta_2$ for two bounds β_1, β_2, it means that β_2 is at least as strong as β_1—meaning that any β_2-map is also a β_1-map. The weakest bounded homomorphism is the $(\mathsf{S} \wedge \mathsf{P} \wedge \mathsf{O})$-map, while $\top$-map is the strongest. Figure 2 offers a compact explanation of the patterns of new triples that the target is willing to accept under the different bounds, i.e., the permissible triples in $H \setminus G$ for a homomorphism $h : G \to G \cup H$. In the figure, the patterns use n ('new') to indicate that an element in this position *must* be new to G ($\mathsf{n} \notin G$), while a ('any') specifies that *any* element is allowed ($\mathsf{a} \in \mathcal{U}$). Multiple patterns in a position of the lattice mean that a triple matching any of the patterns satisfies the corresponding bound.

Bounded homomorphisms can themselves be combined to yield new bounded homomorphisms:

Theorem 1. [15, Lemma 7] If h_1, h_2 are homomorphisms bounded by β_1 and β_2 respectively, and $h_1(u) = h_2(u)$ for all $u \in \mathsf{dom}(h_1) \cap \mathsf{dom}(h_2)$. Then $h_1 \cup h_2$ is a bounded homomorphism satisfying the infimum of $\{\beta_1, \beta_2\}$.

What this means in practice is that the 19 different bounds in the lattice may be used exercise detailed control over incoming data—if desirable down to the level of the individual vocabulary element and be combined into one homomorphism. We shall work through an example in Section 3. For now, we only pause by the bound labelled $\mathsf{S} \wedge \mathsf{P} \wedge \mathsf{O}$. This is the weakest non-trivial bound in the lattice, and

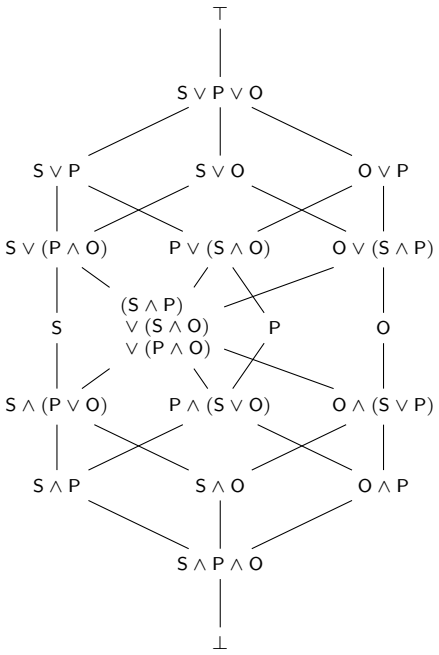


Fig. 1. Bounds.

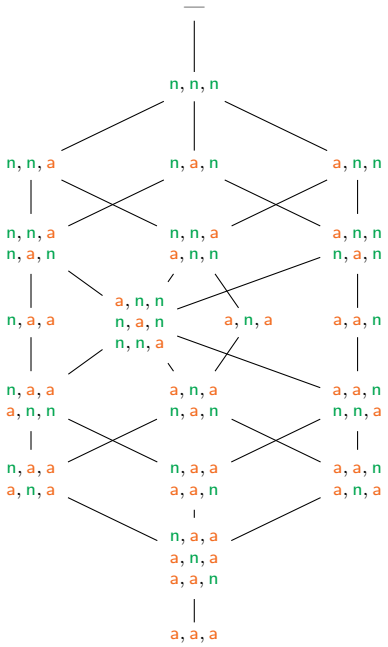


Fig. 2. Permissible triple patterns.

it says that resources known to the recipient cannot be put in known relationship to one another, if they do not already stand in those relationships. Since it is the weakest, it follows that every other bound enforces the same restriction. It is interesting therefore that the $(\mathsf{S} \wedge \mathsf{P} \wedge \mathsf{O})$-bound simulates conservative extensions for a restricted class of ontologies—a not insignificant fact that we record next.

2.1 Relation to Conservative Extensions of Ontologies

The notion of a conservative extension has received a fair bit of attention in the description logic literature in recent years, cf., [3, 7], as it provides a mathematical handle on what it means to amend an ontology without compromising the set of conclusions that that ontology already licenses. On the face of it, this ambition is somewhat similar to ours, so it is natural to consider the relationship between the two notions. To be sure, the dissimilarities are at least as obvious: the concept of a conservative extensions is a logical notion, whereas the concept of a bounded homomorphism is purely graph theoretic. For essentially the same reason, the former applies primarily to ontologies, whereas the latter can apply to any formalism represented as graphs. Nevertheless, there are circumstances under which the two concepts coincide.

Henceforth, an *ontology* is a set of ontological axioms formulated in a language representable in OWL. The *signature* of an ontology or axiom χ, $\mathsf{sig}(\chi)$, is the set of concept, role and individual names occurring in χ, and $\models$ will represent

the standard entailment relation for OWL semantics. Furthermore, let $\mathsf{RDF}(\chi)$ be the RDF representation of an ontology or axiom χ as defined in [9], and set the following two definitions.

Definition 4 (Conservative extension). *Let $\mathcal{O}_1$ and $\mathcal{O}_2$ be ontologies such that $\mathcal{O}_1 \subseteq \mathcal{O}_2$. We say that $\mathcal{O}_2$ is a conservative extension of $\mathcal{O}_1$ if for every axiom α with $\mathsf{sig}(\alpha) \subseteq \mathsf{sig}(\mathcal{O}_1)$ we have $\mathcal{O}_2 \models \alpha$ iff $\mathcal{O}_1 \models \alpha$.*

Definition 5 (Saturated, single triple ontology). *An ontology $\mathcal{O}$ is a* saturated, single triple ontology *if 1) for all $\alpha \in \mathcal{O}$, $\mathsf{RDF}(\alpha)$ is a single triple, and 2) if $\mathcal{O} \models \alpha$ then $\alpha \in \mathcal{O}$.*

Theorem 2. *Let $\mathcal{O}_1$ and $\mathcal{O}_2$ saturated, single triple ontologies, then $\mathcal{O}_2$ is a conservative extension of $\mathcal{O}_1$ iff $\mathsf{RDF}(\mathcal{O}_2)$ is a bounded extension of $\mathsf{RDF}(\mathcal{O}_1)$.*

Proof. To simplify the proof we will use the following simple lemma: G is a bounded extension of H iff $H \subseteq G$ and if $\boldsymbol{t} \in G \setminus H$, then $u \notin \mathcal{U}(H)$ for some $u \in \boldsymbol{t}$. If $\mathcal{O}_1 \not\subseteq \mathcal{O}_2$, then, trivially, $\mathcal{O}_2$ is not a conservative extension of $\mathcal{O}_1$ and $\mathsf{RDF}(\mathcal{O}_2)$ is not a bounded extension of $\mathsf{RDF}(\mathcal{O}_1)$, so assume otherwise. Let $h : \mathsf{RDF}(\mathcal{O}_1) \to \mathsf{RDF}(\mathcal{O}_2)$ be a bounded homomorphism, and α be axiom such that $\mathsf{sig}(\alpha) \subseteq \mathsf{sig}(\mathcal{O}_1)$. If $\mathcal{O}_2 \models \alpha$, then by Definition 5, $\boldsymbol{t} = \mathsf{RDF}(\alpha) \in \mathsf{RDF}(\mathcal{O}_2)$, where $\boldsymbol{t}$ is a single triple. Since $\mathsf{sig}(\alpha) \subseteq \mathsf{sig}(\mathcal{O}_1)$, we can apply the lemma and get that $\mathsf{RDF}(\alpha) \in \mathsf{RDF}(\mathcal{O}_1)$, thus, by Definition 5 again, $\mathcal{O}_1 \models \alpha$. If $\mathcal{O}_1 \models \alpha$, then $\mathcal{O}_2 \models \alpha$, by Definition 5 and the fact that h maps identically from $\mathsf{RDF}(\mathcal{O}_1)$ to $\mathsf{RDF}(\mathcal{O}_2)$. The other direction of the proof is similar.

The conditions of Definition 5 put strong requirements on such ontologies. The first condition requires that all axioms are represented in the RDF mapping of the OWL ontology as singleton triples. This restricts the permissible ontology language, but leaves a well-identified and still useful subset. The set of OWL axioms expressible using a single triple is listed in [9] and is also used to define the *OWL LD* profile [4] (LD for linked data). This profile is the subprofile of the standardised *OWL RL* profile [8] restricted to single triple axioms, and is especially designed for the Linked Data community after evaluating the use of ontological constructs in the web of data. It turns out that this profile covers the better portion of the language that is actually in use. Roughly, the profile contains all of the RDFS vocabulary and the different "equality/inequality" axioms for classes, properties and individuals from OWL 2, e.g., `owl:disjointWith`, `owl:equivalentProperty`, `owl:sameAs` and `owl:differentFrom`, and additionally property types like `owl:FunctionalProperty` and `owl:TransitiveProperty`. Important omissions from the profile are `owl:someValuesFrom`, `owl:allValuesFrom`, cardinality axioms, and `owl:unionOf` and `owl:intersectionOf`. The second requirement of the definition states that the ontology must be completely saturated, i.e., all consequences must be explicitly stated in the ontology. In general, this would be an impossible problem for most ontology languages as the set of all consequences would be infinite. However, for the profile we are restricted to by the first requirement of Definition 5, the size of a completely saturated ontology,

when excluding datatype support, is bounded by $|C|^3$, where C is the number of resources occurring in the ontology and entailment ruleset [4]. We believe that this shows that for our purposes the notion of a saturated, single triple ontology is still a useful one.

It should be emphasised that the case where conservative extensions and bounded homomorphisms coincide has been carefully circumscribed, and that the similarities do not stretch all that far. Conservative extensions cannot in general be simulated by bounded homomorphisms as we have defined them. Conversely, adding `dcterms:subject rdfs:subPropertyOf ex:topic` to an ontology that does not already contain `ex:topic` is conservative, and so is adding a new election result given that the election result is codified in the recipient's terms. Hence, conservative extensions do not offer the detailed level of control required to prevent the kind of cases that were described in the introduction. Yet, these cases are easy to express with bounded homomorphisms. A related fact is that it is not in general possible to express bounds with ontologies. One reason is the close connection between description logics and the guarded fragment of first-order logic, which does not make it possible to express dependencies between two variables of the kind necessary for formulating bounds. Moreover, OWL is not 'directional' in the sense that a homomorphism is, and does therefore not distinguish between elements from the source and the target. We conclude that it is natural to construct a special purpose vocabulary and software to represent and manage such relations.

3 Vocabulary

The Boundz vocabulary[4] comprises 32 classes, 34 properties and 3 individuals; an informal and simplified overview of its most important top-level classes and properties is depicted in Figure 3. The central class of the vocabulary is `Bound`. All the bounds in Figure 1 are represented as subclasses of this class and in the same hierarchical structure as in the figure. The atomic bounds are `S`, `P` and `O` (and $\top$ and $\bot$), all other bounds are defined from these. Bounds may be assigned to graphs, making the graph a `BoundedGraph` which should be taken to mean that any incoming data to this graph must satisfy the bounds in order to be accepted by the graph. If multiple bounds are overlapping in scope, the strongest bound overrides weaker bounds, i.e., the accepted exchange should always satisfy all bounds. An `ExchangeSchema` is a different way of placing bounds on graphs. It is a specification for a data exchange from a set of source graphs to one target graph that must satisfy the bounds in the schema. The result is an `Exchange` which contains the payload, i.e., the set of triples that successfully passed the bounds, and a set of `Violations`, which contains the triples not meeting the requirements of the bounds. An exchange schema can also specify whether of not to require that the sources and target are saturated by a reasoner, and if the payload and/or violations should be listed in the resulting exchange instance. The latter is convenient to control if one just wants to check whether a set of

[4] Vocabulary URI: `http://sws.ifi.uio.no/vocab/boundz`

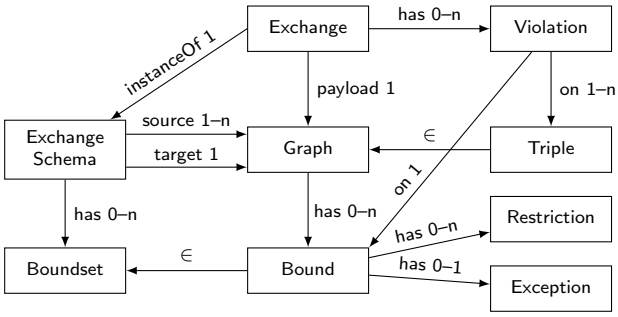


Fig. 3. Boundz vocabulary, an informal and simplified excerpt.

bounds are satisfied, and not to replicate the sources into the exchange. The vocabulary also includes the possibility of placing `Restriction`s and `Exception`s on bounds. A restriction gives a way of constraining the scope of the bound to only concern triples with elements of a specified value, of a given type, or belonging to a certain namespace. An exception applies when the conditions posed by a bound are not met and prescribes how the data that did not break the bound should be handled. The current possibilities are to abort the exchange altogether, ignore all data from the data source causing the violation, or ignore only the problematic triples and accept the remaining triples regardless of the data source.

In the same spirit as the vocabulary R2R [2], with which RDF dataset mappings can be specified and published for sharing and re-use, we believe that the Boundz vocabulary can be used to formulate and share bounds for vocabularies. With this in mind we have published a set of bounds which restricts the use of vocabulary elements in the RDFS vocabulary and the OWL LD profile in those cases where one wants protection from ontology hijacking.[5] We believe that this library can grow by adding useful specialised bounds which have natural interpretations for popular vocabularies.

Example 1. The BBC Music dataset contains, amongst other things, data about artists and their record releases, represented in part using the FOAF vocabulary and the Music Ontology.[6] A `mo:MusicArtist` is related to his or her `mo:Record`s by the `foaf:made` relation and may have many `mo:fanpage`s. A record may be of a certain `mo:Genre`. Suppose the BBC wishes to protect its dataset by requiring that amendments meet the following requirements:

1. The vocabulary that the BBC uses must not be hijacked by adding new superclasses or superproperties.
2. Adding new `foaf:made` relationships is not tolerated, unless both artist and record is new to the BBC dataset; their current library is regarded as complete

[5] Vocabulary URI: `http://sws.ifi.uio.no/vocab/boundzLibrary`

[6] See `http://datahub.io/dataset/bbc-music` and sample `http://www.bbc.co.uk/music/artists/79239441-bfd5-4981-a70c-55c3f15c1287.rdf`.

with respect to the albums of enlisted artists, but is open for extensions with new artists.
3. More fanpages may be added, but an existing fanpage cannot be related to more artists.
4. No new information about existing genres may be added.
5. Also, assume the BBC keeps a special dataset about the Beatles which is not under their management, so they want to disallow any new information using only elements from this dataset. However, new information may *relate* to the Beatles dataset.

These requirements are enforced by the following bounds:

```
1  ex:bbcmusic a bz:BoundedGraph ;
2    bz:hasBound bzs:RDFS ,
3                [ a bz:Aso ;  bz:predicateValue foaf:made ] ,
4                [ a bz:o ;  bz:predicateValue mo:fanpage ;
5                            bz:hasException bz:ignoreViolations ] ,
6                [ a bz:T ;  bz:subjectClass mo:Genre ] ,
7                [ a bz:T ;  bz:objectClass mo:Genre ] .
8  ex:beatles a bz:BoundedGraph ;
9    bz:hasBound [ a bz:KKspo ] .
```

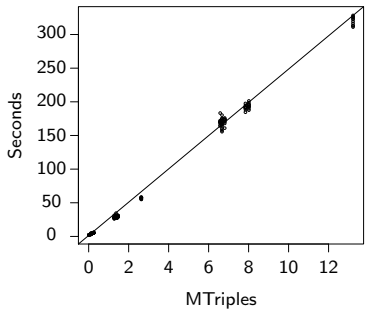
The Boundz vocabulary identifies bounds with URIs using the bounds' label from Figure 1 written in prefix notation[7] as the localname of the URI, e.g., `bz:KKspo` identifies the bound $\mathsf{S} \wedge \mathsf{P} \wedge \mathsf{O}$, `bz:Aso` is $\mathsf{S} \vee \mathsf{O}$, `bz:s` is S, and `bz:T` is $\top$. The example bounds specification above talks about two bounded graphs, `ex:bbcmusic` and `ex:beatles`. The latter graph is bounded by `bz:KKspo` (line 9) which assures that requirement 5 is met; no new triple may re-arrange elements in the Beatles dataset. However, it allows triples where at least one element is not in the receiving dataset, hence, adding triples that use only in part elements from the dataset is permitted. The remaining bounds concern the BBC Music's graph. The bound on line 2 is defined in the `boundzLibrary` vocabulary and protects the `ex:bbcmusic` dataset from being hijacked by RDFS axioms, i.e., axioms that superimpose new superclasses, superproperties, and domain and range definitions onto existing concepts and properties. Requirement 3 is specified with the bound on line 3 which disallows adding new `foaf:made` relationships unless both the subject and object of the triple is new to the receiving target. Line 4 contains a bound which allows adding fanpages if the object of the triple, i.e., the fanpage resource, is new to the target. This bound is equipped with an exception which ignores the violating triples of this bound, and allows other triples to pass. All other bounds in the listing will reject the complete incoming dataset if their conditions are not satisfied. The bounds on lines 6 and 7 require that new triples where the subject or object is of type Genre in the BBC dataset cannot be added, thus making sure that nothing new can be said about genres, i.e., they are write-protected.

[7] Also called polish notation; K (koniunkcja) means *conjunction* and A (alternatywa) means *disjunction*. We use this notation to avoid parenthesis (and other URL unfriendly characters) in the bound labels.

4 Implementation

To test the practical usefulness of the Boundz vocabulary we have implemented a test prototype which takes as input an RDF file containing one or more exchange schemata and computes and outputs an exchange instance for every schema. The prototype is written in Java using the Jena framework[8] and Pellet reasoner.[9] After the input file is read into memory, we apply reasoning to the exchange schemata to reveal possible inconsistencies and allow for a simpler parsing of the vocabulary model using, e.g., superproperties to discover the different schemata settings. For each exchange schema, the specified source and target graphs are read into memory, and saturated if this is specified. Exchanges are then computed and written to output according to settings in the schema. Bounds are checked by a simple algorithm which iterates through the specified bounds, searching for violating triples.

The prototype implementation is evaluated using the Lehigh University Benchmark data generator.[10] The data generator allows one to create datasets of different sizes and with different content by supplying a random seed. We have generated different combinations of source data ranging in total sizes from 15K triples to 13M triples and tested a single exchange schema specification with various bounds against one target of 6M triples. Each test was repeated 10 times on a regular desktop computer using a maximum of 8 GB of heap space. The running times for checking the bounds and producing output—disabling output of the payload and violation triples, and excluding the time to load the source and target graphs into memory—are presented in the graph above, the x-axis indicates the sum of the triples in the sources and the y-axis the time in seconds to complete the check. This simple evaluation shows promising results, the increase in time spent develops linearly against increase in size of input, and the running time for checking bounds with sources of 13M triples against a 6M triple sized target is ≈5 minutes. The prototype implementation, complete test and test results are published on `http://sws.ifi.uio.no/project/boundz`.



5 Related work

An important approach to RDF validation is based on expressing integrity constraints in an OWL ontology. Since OWL is designed to supplement rather than to validate data, this approach involves interpreting parts, or the whole of an ontology under a closed world semantics [10, 12, 17, 18]. Tools such as TrOWL

[8] `http://jena.apache.org/`
[9] `http://clarkparsia.com/pellet/`
[10] `http://swat.cse.lehigh.edu/projects/lubm/`

and Pellet ICV implement this approach, which has the virtue that constraints can be automatically inferred from the domain description in the ontology.

Another approach is represented by the SPIN SPARQL syntax which offers a vocabulary for encoding SPARQL queries in RDF [6]. The idea is to link class definitions with SPARQL queries to capture constraints and rules that formalise the expected behaviour of those classes.

The IBM Resource Shapes vocabulary [11] describes the properties that a resource of a given types is required to have. Validation over resource shapes can then be implemented as a set of ASK queries over the graph.

The current paper is a full version of [14].

6 Conclusion

We have presented a vocabulary that can be used for implementing the reservations a data hub might have against incoming data that is not under the control of that data hub itself, and we have presented elements of the theory behind it. The vocabulary expresses constraints on an incoming contribution in terms of what data the hub already contains. These constraints can be formalised as bounded homomorphisms from the consuming data hub into the union of the consumer and contribution.

The possible uses for bounds we have currently identified are automatic validation (or rejection) of incoming data, identifying conservative extensions of simple ontological schemata, write-protecting (parts of) datasets—with different degrees of strength, and simple implementations of trust, e.g., ignoring sources that do not meet specific bounds. Bound sets corresponding to these use cases can be published as independent RDF resources, and they can be combined and re-used for data hubs with similar needs. Checking conformance with a bound set is computationally tractable, and testing shows that it is practically feasible even over fairly large datasets. Our experimental evaluation indicates that the execution time grows linearly in the size of input data.

Ideas for future work include integrating the Boundz vocabulary with existing vocabularies for describing the content of RDF sources, for instance by using the VoID vocabulary [1] to capture the relationship between the receiving dataset and the exchange payload. We also plan to extend the theory and the vocabulary to cover a symmetric notion of bounds. Currently, our approach is based on regarding one graph as the receiver and the other as the contributor, and the bounds are designed to protect the content of the receiver from being distorted or skewed by the contributor. A natural generalisation is to consider both as peers and to redefine the payload as the uncontroversial subset of the union of the datasets. A potentially interesting further development is to add degrees of trust to the mix by ordering bound sets according to priority or the trustworthiness of its issuer.

References

1. K. Alexander et al. *Describing Linked Datasets with the VoID Vocabulary*. W3C Interest Group Note. W3C, 2011. URL: http://www.w3.org/TR/void/.
2. C. Bizer and A. Schultz. "The R2R Framework: Publishing and Discovering Mappings on the Web". In: *Proc. of the First Int. Workshop on Consuming Linked Data (COLD2010)*. 2010.
3. S. Ghilardi, C. Lutz, and F. Wolter. "Did I Damage my Ontology? A Case for Conservative Extensions in Description Logics". In: *Proc. of the 10th Int. Conference on Principles of Knowledge Representation and Reasoning (KR'06)*. 2006.
4. B. Glimm et al. "OWL: Yet to arrive on the Web of Data?" In: *Proc. of the WWW2012 Workshop on Linked Data on the Web (LDOW 2012)*. 2012.
5. A. Hogan, A. Harth, and A. Polleres. "Scalable Authoritative OWL Reasoning for the Web". In: *Int. Journal on Semantic Web and Information Systems* 5.2 (2009), pp. 49–90.
6. H. Knublauch. *SPIN - Modeling Vocabulary*. W3C Member Submission. W3C, 2012. URL: http://www.w3.org/Submission/spin-modeling/.
7. C. Lutz, D. Walther, and F. Wolter. "Conservative Extensions in Expressive Description Logics". In: *Proc. of the 20th Int. Joint Conference on Artificial Intelligence (IJCAI-07)*. 2007.
8. B. Motik et al., eds. *OWL 2 Web Ontology Language: Profiles (Second Edition)*. 2012. URL: http://www.w3.org/TR/owl2-profiles/.
9. P. F. Patel-Schneider and B. Motik, eds. *OWL 2 Web Ontology Language: Mapping to RDF Graphs (Second Edition)*. 2012. URL: http://www.w3.org/TR/owl2-mapping-to-rdf/.
10. Y. Ren, J. Z. Pan, and Y. Zhao. "Closed World Reasoning for OWL2 with NBox". In: *Tsinghua Science and Technology* 15.6 (2010), pp. 692 –701.
11. A. Ryman, A. L. Hors, and S. Speicher. "OSLC Resource Shape: A language for defining constraints on Linked Data". In: *Proc. of the WWW2013 Workshop on Linked Data on the Web (LDOW 2013)*. 2013.
12. E. Sirin and J. Tao. "Towards Integrity Constraints in OWL". In: *Proc. of the 6th Int. Workshop on OWL: Experiences and Directions (OWLED 2009)*. 2009.
13. M. G. Skjæveland and A. Stolpe. *Bounded RDF Data Transformations*. Tech. rep. University of Oslo, 2012. URL: http://hdl.handle.net/10852/9104.
14. M. G. Skjæveland and A. Stolpe. *Bounds: Expressing Reservations about Incoming Data*. Position paper for W3C's RDF Validation Workshop—Practical Assurances for Quality RDF Data. 2013.
15. A. Stolpe and M. G. Skjæveland. "Preserving Information Content in RDF Using Bounded Homomorphisms". In: *The Semantic Web: Research and Applications*. Vol. 7295. LNCS. Proc. of the 9th ESWC 2012. 2012, pp. 72–86.
16. O. Suominen, K. Viljanen, and E. Hyvänen. "User-Centric Faceted Search for Semantic Portals". In: *The Semantic Web: Research and Applications*. Vol. 4519. LNCS. Proc. of the 4th ESWC 2007. Springer, 2007, pp. 356–370.
17. J. Tao. "Adding Integrity Constraints to the Semantic Web for Instance Data Evaluation". In: *The Semantic Web – ISWC 2010*. Vol. 6497. LNCS. Springer, 2010, pp. 330–337.
18. E. Thomas, J. Z. Pan, and Y. Ren. "TrOWL: Tractable OWL 2 Reasoning Infrastructure". In: *The Semantic Web: Research and Applications*. Vol. 6089. LNCS. Proc. of the 7th ESWC 2010. 2010, pp. 431–435.