Metadata for Web Ontologies and Rules: Current Practices and Perspectives

Carlos Tejo-Alonso, Diego Berrueta, Luis Polo, and Sergio Fernández

Fundación CTIC
Gijón, Asturias, Spain
{carlos.tejo,diego.berrueta,luis.polo,sergio.fernandez}@fundacionctic.org
http://www.fundacionctic.org

Abstract. The Semantic Web contains a number of different knowledge artifacts, including OWL ontologies, RIF rule sets and RDF datasets. Effective exchange and management of these artifacts demand the use of metadata and prompt availability of accurate reference documentation. In this paper, we analyze the current practices in metadata usage for OWL ontologies, and we propose a vocabulary for annotating RIF rules. We also introduce a software tool –Parrot– that exploits these annotations and produces reference documentation for combinations of ontologies and rules.

1 Introduction and Motivation

One of the goals of the Semantic Web is to leverage the web infrastructure for exchanging machine-readable knowledge. A full stack of technologies has being developed to this end, including a framework for resource descriptions (RDF [7]), some schema-definition language such as RDF Schema [5] and OWL [14], and the RIF family of rule interchange languages [4]. Some of these W3C standards have gained wide adoption. Particularly, OWL ontologies and RDF Schema vocabularies are being effectively exchanged on the web. A large amount of "linked data" has flourished in the last few years [3], although structured descriptions of the corresponding datasets seem to be one step behind. At the moment of this writing, one year after the RIF specifications reached maturity, rule interchange on the web is still marginal.

We believe that some of the burdens that prevent the take-off of RIF documents interchange are the lack of companion tools and the absence of guidelines for adding metadata to the rules. The purpose of this paper is to make contributions to both fronts: firstly, we propose a metadata scheme for RIF rules; secondly, we introduce Parrot, a software tool that produces human-oriented reference documentation for combinations of OWL and RIF.

The rest of the paper is structured as follows: in the next section, we examine the state of the art regarding vocabularies and tools for web artifacts metadata management. We learn from the study of current practices of metadata usage in Section 3, and we apply our findings to propose a metadata scheme for RIF rules (Section 4). We introduce Parrot in Section 5, and finally, Section 6 closes the paper with conclusions and some insights into future work.

E. García-Barriocanal et al. (Eds.): MTSR 2011, CCIS 240, pp. 56–67, 2011.

2 State of the Art

We split our review of the state of the art in two parts. On the one hand, some vocabularies for expressing metadata annotations are listed. On the other one, some software tools that use these vocabularies for generating documentation are evaluated. These two visions are complementary, as tools are supported by vocabularies.

2.1 Metadata Vocabularies

Some initiatives have produced schemas to annotate different kinds of resources with metadata, and some of these schemas are available as RDF vocabularies:

Dublin Core[1] is the result of an initiative to provide a small and fundamental group of metadata elements for annotating documents. It has two flavors, the older Dublin Core Elements[2] and the newer Dublin Core Terms[3].

RDF Schema, or RDFS, is an application of RDF to the description of RDF vocabularies. It includes a basic set of properties for metadata, such as `rdfs:label` and `rdfs:comment`.

OWL introduces a few properties for capturing versioning information and compatibility notes. As OWL is built on top of RDF Schema, authors are encouraged to also use RDFS metadata properties.

SKOS, the Simple Knowledge Organization System [13], is a common data model for sharing and linking knowledge organization systems on the web. It provides a basic vocabulary for associating lexical labels to any kind of resource. It introduces the distinction to among preferred (`skos:prefLabel`), alternative (`skos:altLabel`) and "hidden" (`skos:hiddenLabel`) lexical labels.

VANN introduces terms for annotating descriptions of vocabularies with examples and usage notes [8].

Beyond these vocabularies for general-purpose metadata, there are some others specially designed for describing a concrete domain or artifact, such as datasets. Because of the large amount of data that is becoming available on the web, new issues arise. A common need is to publish meta-descriptions of the data stored on datasets. To this end, some of the most relevant proposals are VoID [1], DCat[4] and voidp[5].

Furthermore, there are some vocabularies, such as FOAF [6] or OpenGraph [15], that are commonly found in metadata annotations even if they were not introduced with this purpose.

We note that there is not any specific vocabulary for rules, at least with a significant adoption. Moreover, the RIF specification[6] suggests 9 properties to be used in annotations. In Section 4 we extend this set to 31 properties in order to expand its coverage to areas such as legal rights or related multimedia objects.

1 `http://dublincore.org/`
2 `http://purl.org/dc/elements/1.1/`
3 `http://purl.org/dc/terms/`
4 `http://www.w3.org/egov/wiki/Data_Catalog_Vocabulary`
5 `http://www.enakting.org/provenance/voidp/`
6 `http://www.w3.org/TR/rif-prd/#Annotation`

2.2 Tools

A number of tools can generate reference documentation for RDFS and OWL ontologies:

OWLDoc[7] generates JavaDoc-like HTML pages from an OWL ontology. OWLDoc works together with Protégé-OWL.

SpecGen[8], is an off-line, ontology specification generator. It combines a template with static text with an index of the vocabulary terms and the detailed views of each one. It has been used to generate the companion documentation of some popular vocabularies, such as FOAF or SIOC.

VocDoc[9] is a Ruby script which produces documentation for RDFS/OWL ontologies and vocabularies. It is inspired by SpecGen, and it adds the LaTeX output to make it easier to include the report in larger documents, such as project deliverables or technical reports.

Neologism[10] is a web-based RDF Schema vocabulary editor and publishing system [2]. The main goal of Neologism is to dramatically reduce the time required to create, publish and modify vocabularies for the web of data, and to provide companion documentation.

All the aforementioned tools deal exclusively with vocabularies and ontologies. Regarding rules, commercial rule management systems such as IBM WebSphere ILOG JRules or ontoprise OntoStudio, can generate documentation about the rules in their particular proprietary formats. However, to the best of our knowledge, there is not any solution for documenting standard web rules. The RIF specification suggests some properties to be used for metadata, but there is no evidence of any previous tool supporting these properties.

Expanding the horizons of this analysis of the state of the art, we note that generic "linked data" browsers[11] can be used to visualize (and therefore, to document) any kind of resource published on the web. However, due to their general approach and their orientation to instances (as opposed to vocabularies), they provide limited help to grasp an ontology or a rule.

3 Analysis of Vocabulary Metadata in the Wild

In this section, we present a survey of the actual usage of metadata in ontologies/vocabularies publicly available on the web. For this study, we examine 23 of the most popular RDFS/OWL vocabularies according to the metrics available from reference web sites. The list is assembled as the union of the top-25 popular vocabulary list from `prefix.cc`[12] and top-18 from `pingthesemanticweb.com`[13].

[7] `http://protegewiki.stanford.edu/wiki/OWLDoc`
[8] `http://forge.morfeo-project.org/wiki_en/index.php/SpecGen`
[9] `http://kantenwerk.org/vocdoc`
[10] `http://neologism.deri.ie`
[11] `http://www.w3.org/wiki/TaskForces/CommunityProjects/LinkingOpenData/SemWebClients`
[12] `http://prefix.cc/popular/all`
[13] `http://pingthesemanticweb.com/stats/namespaces.php`

A number of vocabularies are not considered due to their redundancy (some of them belong to a family of vocabularies, such the DBPedia family of namespaces). Additionally, the “time” and “creativecommons” vocabularies have been cherry-picked due to their obvious relevancy to metadata, even if they do not appear in the top positions of the popularity ranking.

We exclude large ontologies from our analysis (e.g., Yago, SUMO or WordNet). At this stage, we focus on small, highly reused vocabularies. We plan to extend our study to these ontologies in the future.

Each one of the RDFS/OWL documents that define these vocabularies has been manually examined, and a comprehensive list of all the metadata properties in use has been collected. The results are captured in Table 1. Metadata properties are described in rows, and are sorted by decreasing usage frequency. Tick marks in this table indicate that the vocabulary of the column uses at least once the metadata property of the row (typically, to annotate one of its classes or properties). For the sake of conciseness, vocabularies in columns are identified by their usual prefix[14].

The results reveal that RDF Schema annotation properties are massively popular. By far, the most frequent metadata associated to vocabulary artifacts are labels and comments. Titles and descriptions are common too, with two namespaces being used for equivalent purposes (Dublin Core Terms and Dublin Core Elements). This duality is also present in other properties, such as the ones used to express attribution (creator and contributor). Legal rights and license information is only present in a minority of the vocabularies.

Versioning information is often limited to simple textual annotations that use `owl:versionInfo` and `dct:hasVersion`, some of them automatically generated by the VCS (version control system) used by the vocabulary authors. It has been observed that some vocabularies convey versioning information in their comments. This practice may be convenient for manual management of the vocabularies, but it is a hindrance to automated management. Some versioning information is sometimes provided by means of time references. Our study reveals that the generic property `dc:date` is commonly used, while more specific properties such as `dct:issued` and `dct:modified` are limited to the vocabularies controlled by Dublin Core.

The absence of some metadata is also interesting. There is a complete lack of multimedia resources associated to the vocabularies, although many vocabularies include very generic pointers (`rdfs:seeAlso`) to other resources. Moreover, the VANN vocabulary, which was designed with the purpose of annotating other vocabularies, is completely absent from the selected sample.

The SKOS vocabulary is sometimes used to introduce definitions, examples and notes. Regarding linguistic information, it is noticeable that SKOS labeling properties are barely used. In fact, even in those cases that indicate preferred labels, there are no alternative labels. The use of SKOS in this context is pointless, as the same semantics could be simply conveyed by `rdfs:label`. Moreover, it has been observed that approximately half of the sampled vocabularies do not

[14] The full namespace URI can be retrieved by means of a query to prefix.cc.

Table 1. Metadata properties used in popular web vocabularies, sorted by decreasing frequency of use

	dbpedia	foaf	dc	rdf	rdfs	owl	geonames	skos	geo	sioc	dct	gr	doap	wot	rss	vcard	cc	mo	time	bibo	geospecies	txn	dctype
rdfs:label	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓
rdfs:comment		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
rdfs:isDefinedBy		✓	✓	✓	✓	✓		✓		✓	✓	✓	✓	✓	✓			✓		✓	✓	✓	✓
rdfs:seeAlso			✓	✓	✓	✓		✓		✓	✓			✓		✓		✓	✓	✓	✓	✓	
dc:title		✓		✓	✓	✓			✓			✓	✓	✓		✓		✓			✓		
dc:description		✓		✓					✓				✓	✓		✓		✓			✓	✓	
dct:title			✓					✓		✓	✓	✓					✓			✓			✓
owl:versionInfo	✓					✓	✓			✓		✓						✓		✓			
dc:date									✓					✓		✓		✓			✓	✓	
dc:creator												✓	✓			✓					✓		
dct:creator								✓												✓		✓	✓
dct:description								✓		✓	✓									✓			
vs:terms_status		✓												✓				✓		✓			
dct:issued			✓								✓												✓
dct:hasVersion			✓								✓												✓
dct:modified			✓								✓												✓
dc:rights												✓	✓										
dct:publisher											✓												✓
foaf:maker													✓					✓					
skos:definition							✓	✓															
skos:example								✓												✓			
skos:note			✓								✓												
skos:prefLabel							✓														✓		
skos:scopeNote								✓												✓			
dc:contributor												✓											
dc:identifier																					✓		
dc:subject												✓											
dct:contributor								✓															
dct:license												✓											
foaf:homepage												✓											
skos:changeNote																				✓			

explicitly indicate the language of the string literals, which leads to ambiguity. Only a couple of vocabularies contain multilingual metadata.

The results table clearly reflects the fact that some metadata annotations can be captured by different properties, and there is a lack of consensus about which is the preferred one. For instance, the semantics of `rdfs:comment`, `dc:description`, `dct:description` and `skos:definition` are very similar (at least when applied to vocabularies). The choice among them is mainly a matter of the preferences of the author, and it is not exclusive. Some vocabularies use more than one.

Moreover, it has been observed that they are sometimes multivalued (e.g., multiple `rdfs:comments` are attached to the same ontology to separate different aspects of the description).

In the case of the duality between Dublin Core Terms and Dublin Core Elements, it seems that at least for some cases it can be explained by DC Terms being a relatively new specification. It is assumed that newer vocabularies may prefer DC Terms.

4 Proposed Vocabulary for Rule Metadata

This section presents our proposal to describe rules and rule sets with medatata, identifying documentation requirements and relevant vocabularies based on the previous work of Section 3.

Rules, like ontologies, are knowledge-based artifacts that capture domain information in a formal way. They declaratively express the dynamic conditions comprising business logic built upon a data model, which describes the entities of the domain. In other words, a ruleset specifies how a system works. In the web, rules and rule sets can be interchanged using RIF, while data models are typically OWL ontologies. The same concerns arising for documenting ontologies apply to rule sets as well. Technical and business people, such as consultants or domain experts, often bear different interests regarding the usage of these artifacts. Moreover, their background may also diverge and logical training cannot be assumed for business-oriented profiles. Metadata provide, on the one hand, a practical mechanism to organize collections of rules without interfering the domain semantics. On the other one, they help lay and nonprofessional users to understand the vision of the world encoded in knowledge-based systems, for instance, by means of natural language expressions.

RIF is an standard for exchanging rules among rule systems, in particular among web-oriented ones. Technically, RIF is a family of languages, called *dialects*, covering different kind of rules: from logic-programming [12] to production rules [10]. The syntax and semantics of each dialect is rigorously and formally specified, trying to reuse as much machinery as possible, such as the mechanism for annotations.

According to the specification, an annotation can be attached to any term and formula within a RIF document (in RIF PRD dialect, this also includes group of rules). Annotations are optional, and only one annotation is allowed per element.

Although XML is the normative syntax for RIF, in this paper we will use the informative, human-readable RIF Presentation Syntax (PS). An annotation is of the form `(* id` φ `*)`, where `id` represents the identifier of the annotated syntactic element (an URI), and φ is a RIF formula capturing the metadata. In particular, φ is a frame (an expression of the form `s[p ->o]`) or a conjunction of frames (i.e., $\texttt{And(s}_1\texttt{[p}_1\texttt{ ->o}_1\texttt{] , ... , s}_n\texttt{[p}_n\texttt{ ->o}_n\texttt{])}$). An example of a RIF annotation is shown in Listing 1.1. Notice that RIF web-oriented design enables the reutilization of existing vocabularies for annotations, such as Dublin Core or even RDFS annotation properties.

Listing 1.1. A snippet of an annotated rule in RIF format

```
(* ex:rule ex:rule [
      rdfs:label      -> 'Example Rule'@en
      dc:creator      -> 'Luis Polo'
      dc:date         -> '1981-01-20'
      og:video        -> <http://youtu.be/5h10QHpA5EU>
      dct:publisher   -> <http://ontorule-project.eu>
 ] *)
```

Nevertheless, the RIF machinery for annotations is very flexible and offers a lot of syntactic freedom, which difficults the correct interpretation of rule and rule sets metadata. For instance, the identifier (`id`) of the rule is an optional element in the annotation expression. Moreover, there could be frames in φ not describing the annotated element. Therefore, we propose some additional restrictions on RIF annotations in order to simplify their management, on the one hand, and to guarantee some integrity on rule metadata, on the other:

1. It is mandatory to declare an identifier (`id`) of the rule, providing an identity on the web of data. The identifier of the rule not only enables cross-references between rules and other elements of a RIF document, but also to establish links between rules and any RDF resource (for instance, in the Linked Data cloud).
2. Metadata φ must contain at least one frame where the subject is the identifier of the annotation, i.e., the RIF element being described.

Coming back to vocabularies for rules and rule sets annotations, it is worth reminding that it is possible to reuse existent ontologies and vocabularies on the web for this purpose. As both artifacts share requirements with respect to needed metadata, they bring the opportunity to reuse the same resources for both ontologies and rules, without introducing new elements. Table 2 sums up, on the one hand, the kind of metadata required to describe rules and, on the other, our suggestions on which properties can be applied to this end. The reader is encouraged to check the range of the recommended properties in their normative specifications.

One apparent limitation of the normative XML syntax of RIF annotations is that the scope of each metadata expression is constrained to the annotated element, even if the machinery enables references to other rules or terms. This is

Table 2. Recommended list of metadata properties for documenting RIF rules

Metadata	Recommended properties
Labeling	Rules are usually referenced by a label, such as “rule for identifying defects”. These labels can be captured by several properties. An important aspect is to appropriately capture multilingualism. Recommended properties are: `dc:title`, `dct:title`, `rdfs:label`, `skos:prefLabel` and `skos:altLabel`.
Authoring	Typically several entities are associated to a rule or a ruleset, but playing a different role. For instance, someone in a company is the creator of the rule and people from other department may have contributed to its definition. Finally, the organization itself is the responsible for its publication and distribution. Recommended properties are: `foaf:maker`, `dc:creator`, `dc:contributor`, `dc:publisher`, `dct:creator`, `dct:contributor` and `dct:publisher`.
Description	Natural language descriptions of rules are useful in order to provide a human-readable expression of its meaning. Recommended properties are: `dc:description`, `dct:description`, `rdfs:comment`, `skos:definition`, `skos:example` and `skos:note`.
Multimedia	Description of rules may be provided by means of multimedia contents, such as images, videos, graphical tables, etc. Recommended properties are: `foaf:depiction` and `og:video`.
Versioning	Rules, as other knowledge artifacts, are subject to evolution and timeline modifications, which should be tracked. Recommended properties are: `owl:versionInfo`, `dct:hasVersion` and `skos:changeNote`.
Rights	Rules are specifications of IT systems, which might be protected by copyright and distributed under a propietary or private license. Recommended properties are: `dc:rights`, `dct:license`. It is also suggested to use RDF descriptions of licenses, such as the ones available from CreativeCommons.
Dates	Apart from versioning, it is important to capture other temporal stamps relevant for rules, such as rule creation or modification dates. Recommended properties are: `dc:date` (for generic purposes), `dct:issued`, `dct:modified`.
Documentation	Another aspect about rules is the relationship with the sources from which the knowledge has been extracted, typically business documents. Moreover, a rule can also be linked to other kind of resources that provide additional information about it. Recommended properties are: `dct:source`, `rdfs:seeAlso`, `rdfs:isDefinedBy`

Table 3. Interpretation of RIF annotations as RDF graphs

Annotation φ	$\pi(\varphi)$
`s [p -> o]`	`{ s p o }`
$\texttt{s[p}_1\texttt{->o}_1\ \ldots\ \texttt{p}_n\texttt{->o}_n\texttt{]}$	$\{\texttt{s p}_1\ \texttt{o}_1\ ;\ \cdots\ ;\ \texttt{p}_n\ \texttt{o}_n\ \}$
$\texttt{And(F}_1\texttt{, ... , F}_n\texttt{)}$	$\{\pi(\texttt{F}_1)\} \cup \cdots \cup \{\pi(\texttt{F}_n)\}$.

a hindrance for reusing descriptions. To overcome this limitation, we propose an RDF interpretation of RIF annotations, so all the annotations of a RIF document are comprehensively collected in a single RDF graph. Moreover, this interpretation makes it possible to execute SPARQL queries over RIF metadata, and it also fosters information reuse based on the principles of "linked data". Table 3 describes the mapping between RIF metadata expressions (φ) and RDF triples. Notice that φ cannot contain variables and that identifiers have a straightforward translation because both sides use URIs for this purpose. It is worth remarking the divergence between annotations translated by φ and the RDF syntax for RIF proposed by W3C [11]. In our case, semantically-equivalent φ expressions for annotations are provided following [9] (i.e., there exists a direct correspondence between a frame and a triple), while [11] describes an RDF-serialization for its frame-based syntax. Although the latter is more expressive, enabling to capture complete RIF documents, it is notoriously difficult to be queried using SPARQL. The simplicity of common annotations does not justify using this complex RDF syntax for RIF.

5 Parrot: Generating Reference Documentation for Ontologies and Rules

One of the applications of ontology and rule metadata is to produce human-oriented reference documentation. We implemented Parrot, a tool that generates documentation for ontologies, rules and combinations of both of them. In this sense, it is a superset of the tools that have been examined in Section 2. To the best of our knowledge, it is the first implementation of a documentation generator for combinations of OWL and RIF.

The input to Parrot are ontology and rule documents compliant with W3C standards, namely OWL and RIF. Typically these documents are available at public web locations and are identified by their URI. Parrot can retrieve them from the web, although it also supports direct file upload.

After parsing the input documents, Parrot builds an in-memory model of the artifacts they describe, mainly ontologies, classes, properties, instances, rules and rule sets. Then, direct and inverse references between the artifacts are established. For instance, classes do not contain references to rules, but rules do use classes in their definitions; therefore bi-directional references are introduced between pairs of classes and the rules. These references manifest as navigable hyperlinks in the final document.

Parrot also builds indexes of the artifacts. These indexes are later transformed into tables of contents, summaries and glossaries in the generated documentation.

The main part of the reference documentation comprises detail views of each artifact. Figure 1 depicts the detailed view of a RIF rule. Note that different aspects of the metadata are visually separated, and can be individually displayed or hidden by the user. Parrot pays special attention to abstract the complexity of the metadata and the underlying OWL and RIF documents, in an effort to make the knowledge accessible to a larger audience. Moreover, Parrot supports user profiles with different skills and interests. For instance, the "business profile" is tailored to users without technical expertise, but operative knowledge of the domain.

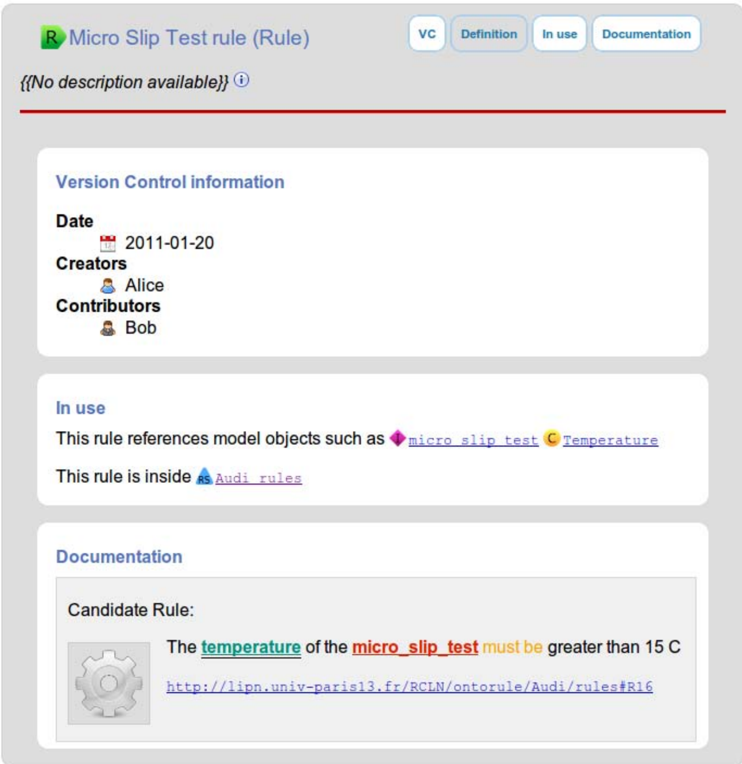

Fig. 1. Screenshot of the detailed view of a rule in Parrot

Parrot has to deal with the fact that a wide range of properties are used in metadata annotations, as was found in Section 3. Therefore, mechanisms are in place to deal with conflicting and redundant annotations. For instance, in the case of annotations that convey essentially the same semantic information, a priority-driven iterative strategy is implemented. Similarly, Parrot handles multilingual annotations and lets the user choose the language used for the documentation. Currently, Parrot supports most of the properties in Tables 1 and 2 and adds a few more[15].

[15] For a complete listing of all the metadata properties supported by Parrot, please check `http://ontorule-project.eu/parrot/help`

This tool is available as a web service[16] and is distributed as open-source [17]. It has been implemented is Java and reuses a number of components such as Jena[18] and Java-RDFa[19]. One remarkable dependency is RIFle[20], a Java toolkit for managing RIF documents. Among other tasks, RIFle parses the annotations in a RIF document and exposes them as a simple RDF graph, according to the mapping described in Table 3.

6 Conclusions and Future Work

We expect this work to have an impact on the quality and quantity of the metadata annotations associated to web ontologies and rules. Firstly, we believe that the guidelines proposed in this paper, as well as the lessons learned from analysing the metadata embedded in publicly available vocabularies, will help the community to be more precise with the metadata they include. This is especially true for rules, because of the current lack of best practices and the vague guidelines provided by the specifications.

Secondly, regarding the quantity of metadata annotations, we hope that the availability of an easy-to-use reference documentation tool will encourage authors to include more metadata. The prompt availability complete reference documentation at no cost should catalyze authors to add metadata. Moreover, it can foster knowledge reuse, by lowering the barrier to gain understanding of ontologies and rules found on the web.

The vocabulary proposed in this paper is not the only one for annotating rules. Some Business Rule Management Systems (BRMS) such as JRules and Drools have their own extensible schemas. Our proposed vocabulary could contribute to exchange metadata between BRMS by suggesting how expressive rule annotations can be captured and interpreted in intermediate RIF documents.

Although the primary target of the vocabulary proposed in Section 4 is RIF, potentially it can be used with any rule language that associates its artifacts to a named RDF resource (URI).

Our perspectives for future work include to extend our analysis to large ontologies. We also plan to assess our proposal for rule metadata by accounting the use of these properties in RIF documents as they gain popularity on the web. To this end, we plan to anonymously monitor the usage trends in the public instance of the Parrot web service. The development roadmap of Parrot also includes extending its coverage to other web resources, such as datasets and queries. Finally we aim to extend rule metadata with new properties to describe other business features, such as their scope and inter-rule relations.

16 http://ontorule-project.eu/parrot/
17 http://sourceforge.net/projects/parrot-project/
18 http://incubator.apache.org/jena
19 https://github.com/shellac/java-rdfa
20 http://rifle.sourceforge.net/

Acknowledgements. The work described in this paper has been partially supported by the European Commission under ONTORULE Project (FP7-ICT-2008-3, project reference 231875).

References

1. Alexander, K., Cyganiak, R., Hausenblas, M., Zhao, J.: Describing Linked Datasets with the VoID Vocabulary (March 2011)
2. Basca, C., Corlosquet, S., Cyganiak, R., Fernández, S., Schandl, T.: Neologism: Easy Vocabulary Publishing. In: Proceedings of the ESWC 2008 Workshop on Scripting for the Semantic Web (SFSW 2008) (June 2008)
3. Berners-Lee, T.: Linked Data design issues (July 2006), http://www.w3.org/DesignIssues/LinkedData.html
4. Boley, H., Kifer, M.: RIF Overview. Working Group Note, W3C (June 2010), http://www.w3.org/TR/rif-overview/
5. Brickley, D., Guha, R.: RDF Vocabulary Description Language 1.0: RDF Schema. Recommendation, W3C (February 2004), http://www.w3.org/TR/rdf-schema/
6. Brickley, D., Miller, L.: FOAF Vocabulary Specification (August 2010), http://xmlns.com/foaf/spec/
7. Carroll, J.J., Klyne, G.: Resource Description Framework (RDF): Concepts and Abstract Syntax. Recommendation, W3C (February 2004), http://www.w3.org/TR/rdf-concepts/
8. Davis, I.: VANN: A vocabulary for annotating vocabulary descriptions. Technical report (2005), http://vocab.org/vann/
9. de Bruijn, J.: RIF RDF and OWL Compatibility. Recommendation, W3C (June 2010), http://www.w3.org/TR/rif-rdf-owl/
10. de Sainte Marie, C., Hallmark, G., Paschke, A.: RIF Production Rule Dialect. Recommendation, W3C (June 2010), http://www.w3.org/TR/rif-prd/
11. Hawke, S., Polleres, A.: RIF In RDF. Working Group Note, W3C (May 2011)
12. Kifer, M., Boley, H.: RIF Basic Logic Dialect. Recommendation, W3C (June 2010), http://www.w3.org/TR/rif-bld/
13. Miles, A., Bechhofer, S.: SKOS Simple Knowledge Organization System Reference. Recommendation, W3C (August 2009), http://www.w3.org/TR/skos-reference/
14. Motik, B., Parsia, B., Patel-Schneider, P.F.: OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax. Recommendation, W3C (October 2009), http://www.w3.org/TR/owl2-syntax/
15. Zuckerberg, M., Taylor, B.: The Open Graph Protocol (April 2010), http://ogp.me/