# Current practices and perspectives for metadata on web ontologies and rules

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

Fundación CTIC
Ada Byron 39, Edificio Centros Tecnológicos, 33203
Gijón, Asturias, Spain
Email: {carlos.tejo,diego.berrueta,luis.polo,sergio.fernandez}@fundacionctic.org
Web: http://www.fundacionctic.org

**Abstract:** The Semantic Web contains a number of knowledge artifacts, including OWL ontologies, RIF rule sets and RDF datasets. Effective exchange and management of these artifacts demands the use of metadata and prompt availability of accurate reference documentation. In this article, 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.

**Keywords:** Metadata; Vocabularies; Ontologies; Rules; Annotations; OWL; RIF;

**References** to this article should be made as follows: Tejo-Alonso, C., Berrueta, D., Polo, L. and Fernández, S. (2012) 'Current practices and perspectives for metadata on web ontologies and rules', *Int. J. of Metadata, Semantics and Ontologies*, Vol. X, Issue. X/X, pp. XXX–XXX.

**Biographical notes:**
Carlos Tejo-Alonso holds a B.Sc. in Computer Science from the University of Oviedo. Since he joined Fundación CTIC in October 2007 he has worked as a research assistant of the Semantic Technologies research unit in the R&D Department, and has taken part in several national and European R&D projects and initiatives. He has experience in various areas of Computer Science such as semantic web, human-computer interaction or web accessibility. He has received awards for his academic performance.

Diego Berrueta holds an M.Sc. in Computer Science from the University of Oviedo, where he graduated top of his class. In March 2005 he joined Fundación CTIC to coordinate the Semantic Technologies research unit, where he is involved in national and European research projects. While participating in the W3C Semantic Web Deployment Working Group, he edited two Technical Reports and reviewed the RDFa specifications. Diego is one of the developers of Vapour, a linked data validation tool; Parrot, an OWL and RIF documentation tool; and RDFaDeveloper, a Firefox plugin that facilitates the annotation of web pages with RDFa.

Luis Polo holds a Master Degree in Linguistics from the University of Oviedo. He also received an Advanced Studies Diploma from the School of Philosophy at the University of Oviedo. At present, he is pursuing his PhD degree. Since March 2005, he has been developing his professional career as a member of the R&D Department at Fundación CTIC in the Semantic Technologies research unit. He is involved in several national and European research projects, such as FP7 ONTORULE and SERENOA.

Sergio Fernández holds an M.Sc on Web Engineering from the University of Oviedo (2010). His bachelor degree dissertation, SWAML, received an award at the first Free Software University Contest (Seville, 2007). Since 2006, he has worked as a Research Assistant in the Semantic Technologies research unit at Fundación CTIC, R&D Department, and has taken part in several national and European research projects and related initiatives. He has done an internship in DERI NUI Galway (Ireland) while participating in the SIOC project. He has published some twenty publications on the topics of Data Integration, Social Semantic Web and Linked Data.

*This article is a revised and expanded version of the paper entitled "Metadata for web ontologies and rules: current practices and perspectives" presented at the 5th Metadata and Semantics Research Conference (MTSR 2011) at Izmir, Turkey, 12-15 October 2011.*

## 1 Introduction and motivation

One of the goals of the Semantic Web is to leverage the web infrastructure to exchange machine-readable knowledge. A full stack of technologies has been developed to this end, including a framework for resource descriptions (RDF (Carroll and Klyne, 2004)), schema-definition languages such as RDF Schema (Brickley and Guha, 2004) and OWL (Motik et al., 2009), and the RIF family of rule interchange languages (Boley and Kifer, 2010). Some of these W3C standards have been widely adopted; notably, OWL ontologies and RDF Schema vocabularies are being effectively exchanged on the web. A great amount of "linked data" has flourished in the recent years (Berners-Lee, 2006), 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 document interchange are the lack of companion tools and the absence of guidelines for adding metadata to the rules. The purpose of this article is to make contributions to both fronts: firstly, we propose a metadata scheme for rules; secondly, we introduce Parrot, a software tool that produces human-oriented reference documentation for combinations of OWL and RIF.

The article 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 apply our findings to propose a metadata scheme for rules (Section 4). Section 5, describes an interpretation of RIF documents as RDF graphs, with a focus on annotations as well as the structure of rules and groups of rules. Parrot is introduced in Section 6, and finally, Section 7 presents our conclusions and insights into future work.

## 2 State of the art

We focus our review of the state of the art on two areas: vocabularies for expressing metadata annotations and software tools that use these vocabularies for generating documentation. These two visions are complementary, as tools are supported by vocabularies.

### 2.1 Metadata vocabularies

A number of 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 version information and compatibility notes. As OWL is built on top of RDF Schema, the use of RDFS metadata properties is also encouraged.

*SKOS*, the Simple Knowledge Organization System (Miles and Bechhofer, 2009), 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 distinctions between "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 (Davis, 2005).

Beyond these vocabularies for general-purpose metadata, there are others especially 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*[4] (Alexander et al., 2011), the Vocabulary of Interlinked Datasets, which is an RDFS vocabulary for expressing metadata about RDF datasets and how they are interlinked.

*DCat*[5], the Data Catalog vocabulary, which is an RDF vocabulary for the description and exchange of data catalogs. Although its primary purpose is the description of government data catalogues, its design can accommodate any kind of data.

*voidp*[6] is a RDF vocabulary that defines concepts to describe provenance relationships of data in linked datasets. It builds on and extends the VoID ontology and is designed to be lightweight. VoID describes linked datasets, and voidp describes linked dataset provenance, enabling easy information discovery and usage.

Furthermore, there are some vocabularies, such as FOAF (Brickley and Miller, 2010) or OpenGraph (Zuckerberg and Taylor, 2010), that are commonly found in metadata annotations although they were not introduced for this purpose.

There is no widely adopted, specific and comprehensive vocabulary for rules. The RIF specification[7] suggests 9 properties to be used in annotations, see Table 1. In Section 4 we extend this set to 31 properties in order to expand its coverage to new areas, such as legal rights or related multimedia objects.

**Table 1** The nine annotation properties suggested in the RIF specification, grouped by vocabulary.

| | |
|---|---|
| RDFS | `rdfs:label` |
| | `rdfs:comment` |
| | `rdfs:seeAlso` |
| | `rdfs:isDefinedBy` |
| DC | `dc:creator` |
| | `dc:description` |
| | `dc:date` |
| FOAF | `foaf:maker` |
| OWL | `owl:versionInfo` |

### 2.2 *Tools*

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

*OWLDoc*[8] generates JavaDoc-like HTML pages from an OWL ontology. OWLDoc works together with Protégé-OWL.

*SpecGen*[9] is a script that produces ontology specifications. It combines a template containing static text with an index of the vocabulary terms and a list of detailed descriptions for each term. SpecGen has been used to generate the companion documentation of some popular web vocabularies, such as FOAF or SIOC.

*VocDoc*[10] is a Ruby script which produces documentation for RDFS/OWL ontologies and vocabularies. It is inspired by SpecGen, and adds a LaTeX output option, that facilitates the inclusion of reports in larger documents, such as project deliverables or technical reports.

*Neologism*[11] is a web-based RDF Schema vocabulary editor and publishing system (Basca et al., 2008). 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 document the rules in their particular proprietary formats. However, to the best of our knowledge, there is no tool that supports standard web rules natively. Moreover, there is no evidence of any previous tool supporting the 9 annotation properties enumerated in the RIF specification.

With respect to datasets, it seems clear that it would be useful for them to be documented. Although there are ongoing efforts to manually collect meta-information about publicly available datasets (for instance, the initiative carried by the Linking Open Data community on CKAN[12], a registry of open knowledge), there is not yet any specific service to generate reference documentation for linked datasets.

Expanding the horizons of this analysis of the state of the art, we note that generic "linked data" browsers[13] 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 in grasping 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 of vocabularies under study is the union of the top-25 popular vocabulary list from `prefix.cc`[14] and the top-18 from `pingthesemanticweb.com`[15]. Some vocabularies have been discarded as redundant (they belong to a family of vocabularies, such the DBPedia family of namespaces). Additionally, the "time" and "creativecommons" vocabularies have been cherry-picked because of their obvious relevancy to metadata, even if they do not appear in the top positions of the popularity ranking.

At this stage, we focus on small, highly reused vocabularies, excluding large ontologies from our analysis (e.g., Yago, SUMO or WordNet). We plan to extend our study to large 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 2. Metadata properties are described in rows, and are sorted in decreasing usage frequency. Tick marks in this table indicate that the vocabulary of the column uses the metadata property of the row at least once (typically, to annotate one of its classes or properties). For the sake of brevity, vocabularies in columns are identified by their usual prefix[16].

The results reveal that RDF Schema annotation properties are massively popular. Labels and comments are by far the most frequent metadata associated to vocabulary artifacts. 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

**Table 2** Metadata properties used in popular web vocabularies, sorted in 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 | | | | | | | | | | | | | | | | | | | | ✓ | | | |

properties, such as the ones used to express attribution (creator and contributor). Legal rights and license information are only present in a minority of the vocabularies.

Version 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 version 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 links (`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 noteworthy that SKOS labeling properties are barely used. In fact, even in those cases that indicate preferred labels, there is no alternative label. 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 explicitly indicate the language of the string literals, which leads to ambiguity. Only two 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 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 separate different aspects of the description of the same ontology ).

In the case of the duality between Dublin Core Terms and Dublin Core Elements, it seems that at least for some cases this can be explained by the fact that DC Terms is 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 metadata, 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 ontology documentation also apply to rule sets. Technical and business people, such as consultants or domain experts, often bear different interests in the usage of these artifacts. Moreover, their backgrounds are diverse and logical training cannot be assumed for business-oriented profiles. Metadata provides a practical mechanism to organize collections of rules without interfering the domain semantics. In addition, 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.

It is worth remembering that it is possible to reuse existing ontologies and vocabularies on the web for annotating rules and rulesets. As both artifacts share requirements with respect to metadata, they provide the opportunity to reuse the same resources for both ontologies and rules, without introducing new elements. The left of Table 3 shows the kind of metadata that can be used to describe rules. The right column contains our suggestions as to which properties can be applied to this end. The reader is encouraged to check the range of the recommended properties in their normative specifications.

## 5 RDF interpretation of RIF documents

This section is divided in two parts. The first part describes a proposed interpretation of the annotations in RIF documents as RDF graphs. The second part discusses a mapping from the structure of a RIF document to RDF statements. Further discussion about this interpretation can be found in González-Moriyón et et al. (2012).

### 5.1 Proposed RDF interpretation of annotations in RIF documents

RIF is a standard for exchanging rules among rule systems, in particular among web-oriented systems. Technically, RIF is a family of languages, called *dialects*, covering different kind of rules: from logic-programming (Kifer and Boley, 2010) to production rules (De Sainte Marie et al., 2010). The syntax and semantics of each dialect is rigorously and formally

**Table 3** Recommended list of metadata properties for documenting 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.<br>Recommended properties are:<br>`skos:prefLabel` ≻ `skos:altLabel` ≻ `dct:title` ≻ `dc:title` ≻ `rdfs:label`. |
| *Authoring* | Typically several entities are associated to a rule or a ruleset, but playing different roles. For instance, a person in a company is the creator of a rule although people from other departments may have contributed to its definition. Ultimately, the organization itself is responsible for its publication and distribution.<br>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.<br>Recommended properties are: `dct:description` ≻ `dc:description` ≻ `rdfs:comment`, `skos:definition`, `skos:example` and `skos:note`. |
| *Multimedia* | Descriptions of rules may be provided by means of multimedia contents, such as images, videos, graphical tables, etc.<br>Recommended properties are: `foaf:depiction` and `og:video`. |
| *Versioning* | Rules, like other knowledge artifacts, are subject to evolution and timeline modifications, which should be tracked.<br>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 proprietary or private license.<br>Recommended properties are: `dc:rights`, `dct:license`. It is also suggested to use RDF descriptions of licenses, such as those 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.<br>Recommended properties are: `dc:date` (for generic purposes), `dct:issued`, `dct:modified`. |
| *Documentation* | Another aspect of rules is the relationship with the sources from which the knowledge has been extracted, typically business documents. Moreover, a rule can be linked to other kinds of resources that provide additional information about it.<br>Recommended properties are: `dct:source`, `rdfs:seeAlso`, `rdfs:isDefinedBy` |

specified, trying to reuse as much machinery as possible, including the mechanism for annotations.

According to the specification, an annotation can be attached to any term or formula within a RIF document (in RIF PRD dialect, this also includes groups of rules). Annotations are optional, and only one annotation per element is allowed. Although XML is the normative syntax for RIF, in this article we will use the informative, human-readable RIF Presentation Syntax (PS). An annotation has the form `(* id` $\varphi$ `*)`, where `id` represents the identifier of the annotated syntactic element (an URI), and $\varphi$ is a RIF formula capturing the metadata. In particular, $\varphi$ 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 Figure 1. Notice that RIF web-oriented design makes it possible to reuse existing vocabularies, such as Dublin Core or even RDFS annotation properties.

```
Import (
 <http://xmlns.com/foaf/spec/>
 <http://www.w3.org/ns/entailment/RDF> )

(* ex:age-ruleset
   ex:age-ruleset
   [ rdfs:label -> "Age Ruleset"@en]
*)
Group rif:forwardChaining 100 (

  (* ex:under18-rule
     ex:under18-rule
     [ rdfs:comment
       -> "This rule checks if a person
           is under 18 years old"@en
       dc:date
       -> "2011-08-29"
       dc:creator
       -> "Carlos Tejo Alonso"
       dc:title
       -> "Under 18 years rule"@en
       rdfs:label
       -> "Checks age under 18 years"@en
       dct:publisher
       -> <http://ontorule-project.eu>
       owl:versionInfo
       -> "1.0-intial draft"]
  *)
  Forall ?person ?age (
    If And(
        ?person#foaf:Person
        ?person[foaf:age -> ?age]
        External(pred:numeric-less-than(?age 18))
    ) Then Do( Assert(?person#ex:Under18Person))
  )
```

**Figure 1** A snippet of an annotated rule in RIF Presentation Syntax.

The RIF machinery for annotations is very flexible and offers much syntactic freedom, which is a hindrance for the correct interpretation of metadata. For instance, the identifier (`id`) of the rule is an optional element in the annotation expression. Moreover, there may be frames in $\varphi$ which do not describe 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 enables cross-references between rules and other elements of a RIF document, and links between rules and arbitrary RDF resources (for instance, from the Linked Data cloud).
2. The formula $\varphi$ must contain at least one frame where the subject is the identifier of the annotation, i.e., the RIF element to which the annotation is attached.

We propose an RDF interpretation of RIF annotations, so all the annotations of a RIF document are comprehensively collected in a single RDF graph. By having the metadata available as an RDF graph, it is possible to query the metadata with SPARQL, and to reuse descriptions according to the principles of "linked data". Table 4 describes the mapping between RIF metadata expressions ($\varphi$) and RDF triples. Notice that $\varphi$ 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 $\varphi$ and the RDF syntax for RIF proposed by W3C (Hawke and Polleres, 2011). In our case, semantically-equivalent $\varphi$ expressions for annotations are provided following (De Bruijn, 2010) (i.e., there exists a direct correspondence between a frame and a triple), while Hawke and Polleres (2011) describes an RDF-serialization for its frame-based syntax. Although the latter is more expressive because it can capture complete RIF documents, it is also notoriously difficult to query using SPARQL. In many practical cases, the simplicity of the annotations does not justify using this complex RDF syntax for RIF.

**Table 4** Interpretation of RIF annotations as RDF graphs.

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

In addition to the metadata that can be obtained from RIF annotations, there is additional information describing rules and groups that can also be merged in the RDF graph. This information includes the priority of the rules and the conflict resolution strategy.

### 5.2 *Proposed mapping from RIF structure to RDF*

The RIF document conveys important information about the hierarchy of rules and groups. The structure of a RIF document can be conceptually interpreted as a tree comprised by two kinds of nodes that match two types of RIF entities. The inner nodes are rulesets (also known as groups) and the leaves are rules.

In order to translate this tree-shaped structure of a RIF document to RDF, we make use of the Rulz (Rules in the Web Zoo) vocabulary [17]. This RDF vocabulary allows the description of rules in the web. The result of the mapping is a set of RDF statements which translate RIF entities into Rulz-typed resources. `GROUP` entities directly translate into `rulz:Ruleset` instances while `RULE` entities direct translate to `rulz:Rule`, as shown in Table 5.

Moreover, relations between these new instances are created using Rulz properties. The property `rulz:subset` connects logically partitioned subsets, while the property `rulz:inRuleset` attaches rules to the ruleset they belong to.

**Table 5** Structure alignment between RIF and Rulz.

| RIF entity | Rulz concept |
|---|---|
| ⟨`GROUP`⟩ | `rulz:Ruleset` |
| ⟨`RULE`⟩ | `rulz:Rule` |

## 6 Parrot: generating reference documentation for ontologies and rules

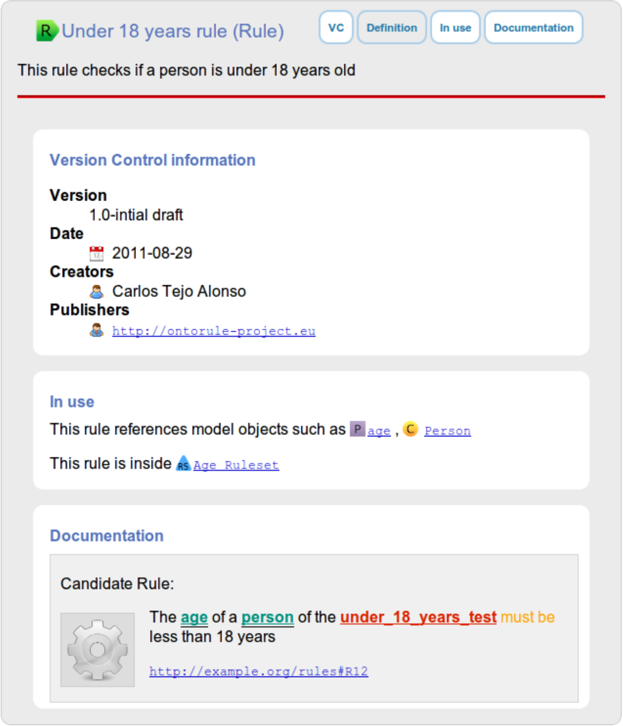
One of the applications of ontology and rule metadata is to produce human-oriented reference documentation. We implemented Parrot, a tool that generates documentation for ontologies, rules and combinations of both. In this sense, it is a superset of the tools that have been examined in Section 2. To the best of our knowledge, this 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. These artifacts are then bidirectionally linked. Direct references are those that explicitly appear as named resources (URIs) in the definition of the artifacts. For instance, in Figure 1 the rule contains a direct reference to the ontology class `foaf:Person` and the property `foaf:age`. Parrot automatically derives inverse references, i.e., a link is created from foaf:Person to the rule `ex:under18-rule`. These references appear 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 detailed views of each artifact. Figure 2 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. In an effort to make the knowledge accessible to a larger audience, Parrot pays special attention to abstracting the complexity of the metadata and the underlying OWL and RIF documents. Moreover, Parrot supports user profiles with different skills and interests. For instance, the "business profile" is tailored to users without technical expertise, but with operative knowledge of the domain. Depending on the user profile, some information is prominently displayed, while other fragments of the report are initially hidden. Nevertheless, the user can adjust the information displayed to suit his preferences.



Under 18 years rule (Rule)

This rule checks if a person is under 18 years old

Version Control information

**Version**
1.0-intial draft
**Date**
2011-08-29
**Creators**
Carlos Tejo Alonso
**Publishers**
http://ontorule-project.eu

In use

This rule references model objects such as P age , C Person

This rule is inside RS Age Ruleset

Documentation

Candidate Rule:

The age of a person of the under_18_years_test must be less than 18 years

http://example.org/rules#R12

**Figure 2** 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. Sometimes, multiple annotations may convey essentially the same semantic information. In the example in Figure 1, two properties `dc:title` and `rdfs:label` are used to label the rule. Parrrot manages this situation by assigning priorities to the properties. According to the priority chain shown in Table 3, only the value of `dc:title` is retained, while the value of `rdfs:label` is discarded, see Figure 2.

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 2 and 3 and adds a few more[18].

This tool is distributed as open-source [19] and it is available in different formats: as a web service[20], as an Eclipse plugin[21] (shown in Figure 3) and as a command line interface (CLI). It has been implemented is Java and reuses a number of components such as Jena[22] and Java-RDFa[23]. One remarkable dependency is RIFle[24], 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 4.

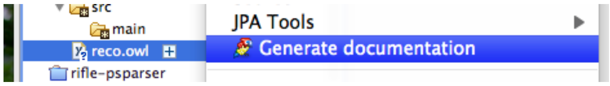


**Figure 3** Screenshot of Parrot as Eclipse plugin.

## 7 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 article, as well as the lessons learned from analyzing 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 of complete reference documentation at no cost should encourage 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 article 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 exchanging 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, it can be used with any rule language that associates its artifacts to a named RDF resource (URI).

Our perspectives for future work include extending 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 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.

## Acknowledgements

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

## References

K. Alexander, R. Cyganiak, M. Hausenblas, and J. Zhao. Describing Linked Datasets with the VoID Vocabulary, March, 2011.

C. Basca, S. Corlosquet, R. Cyganiak, S. Fernández, and T. Schandl. Neologism: Easy Vocabulary Publishing. In *Proceedings of the ESWC2008 workshop on Scripting for the Semantic Web (SFSW2008)*, June, 2008.

T. Berners-Lee. Linked Data design issues, July, 2006. http://www.w3.org/DesignIssues/LinkedData.html.

H. Boley and M. Kifer. RIF Overview. Working Group Note, W3C, June, 2010. http://www.w3.org/TR/rif-overview/.

D. Brickley and R. Guha. RDF Vocabulary Description Language 1.0: RDF Schema. Recommendation, W3C, February, 2004. http://www.w3.org/TR/rdf-schema/.

D. Brickley and L. Miller. FOAF Vocabulary Specification, August, 2010. http://xmlns.com/foaf/spec/.

J. J. Carroll and G. Klyne. Resource Description Framework (RDF): Concepts and Abstract Syntax. Recommendation, W3C, February, 2004. http://www.w3.org/TR/rdf-concepts/.

I. Davis. VANN: A vocabulary for annotating vocabulary descriptions. Technical report, 2005. http://vocab.org/vann/.

J. de Bruijn. RIF RDF and OWL Compatibility. Recommendation, W3C, June, 2010. http://www.w3.org/TR/rif-rdf-owl/.

C. de Sainte Marie, G. Hallmark, and A. Paschke. RIF Production Rule Dialect. Recommendation, W3C, June, 2010. http://www.w3.org/TR/rif-prd/.

G. González-Moriyón, L. Polo, D. Berrueta, C. Tejo-Alonso, and M. Iglesias. Assembling Rule Mashups in the Semantic Web. In *Proceedings of the 9th Extended Semantic Web Conference (ESWC2012)*, May, 2012.

S. Hawke and A. Polleres. RIF In RDF. Working Group Note, W3C, May, 2011.

M. Kifer and H. Boley. RIF Basic Logic Dialect. Recommendation, W3C, June, 2010. http://www.w3.org/TR/rif-bld/.

A. Miles and S. Bechhofer. SKOS Simple Knowledge Organization System Reference. Recommendation, W3C, August, 2009. http://www.w3.org/TR/skos-reference/.

B. Motik, B. Parsia, and P. F. Patel-Schneider. OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax. Recommendation, W3C, October, 2009. http://www.w3.org/TR/owl2-syntax/.

M. Zuckerberg and B. Taylor. The Open Graph Protocol, April, 2010. http://ogp.me/.

## Notes

[1] http://dublincore.org/

[2] http://purl.org/dc/elements/1.1/

[3] http://purl.org/dc/terms/

[4] http://vocab.deri.ie/void

[5] http://www.w3.org/egov/wiki/Data_Catalog_Vocabulary

[6] http://www.enakting.org/provenance/voidp/

[7] http://www.w3.org/TR/rif-prd/#Annotation

[8] http://protegewiki.stanford.edu/wiki/OWLDoc

[9] http://forge.morfeo-project.org/wiki_en/index.php/SpecGen

[10] http://kantenwerk.org/vocdoc

[11] http://neologism.deri.ie

[12] http://thedatahub.org/group/lodcloud

[13] http://www.w3.org/wiki/TaskForces/CommunityProjects/LinkingOpenData/SemWebClients

[14] http://prefix.cc/popular/all

[15] http://pingthesemanticweb.com/stats/namespaces.php

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

[17] http://vocab.deri.ie/rulz

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

[19] http://sourceforge.net/projects/parrot-project/

[20] http://ontorule-project.eu/parrot/

[21] http://ontorule-project.eu/resources/parrot/eclipse/

[22] http://incubator.apache.org/jena

[23] https://github.com/shellac/java-rdfa

[24] http://rifle.sourceforge.net/