Chapter 5
The Semantic Publishing and Referencing Ontologies

Abstract One of the main research areas in semantic publishing is the development of semantic models that fit the requirements of authors and publishers. Although several models and metadata schemas have been developed in the past, they do not fully comply with the vocabulary used by publishers or they are not adequate for describing specific topics (e.g., characterisation of bibliographic citations, definition of publishing roles, description of publishing workflows, etc.). In this chapter I introduce the Semantic Publishing and Referencing (SPAR) Ontologies, a suite of orthogonal and complementary OWL 2 DL ontology modules for the creation of comprehensive machine-readable RDF metadata for every aspect of semantic publishing and referencing. In particular, I show the characteristics and benefits of all the SPAR ontologies, and support the entire discussion with several examples of Turtle code describing a particular reference of the legal discipline, namely Casanovas et al.'s "OPJK and DILIGENT: ontology modelling in a distributed environment".

The *development of semantic models* (vocabularies and ontologies) that fit the requirements of authors and publishers is one of the main research areas in semantic publishing. As I described in Chap. 2, several recent works have proposed metadata schemas, vocabularies and ontologies to describe the publishing domain. However those models show some limitations. Some of them (e.g., Dublin Core Metadata Terms (Dublin Core Metadata Initiative 2012) define bibliographic objects by means of abstract concepts that do not fully comply with the vocabulary used by publishers. Others (e.g., the Bibliographic Ontology (D'Arcus and Giasson 2009)) have been developed to describe parts of the publishing domain, but are not adequate for describing specific topics (e.g., characterisation of bibliographic citations, definition of agent's publishing roles, description of publishing workflows) and are not interoperable with other models (e.g., FRBR (International Federation of Library Associations and Institutions Study Group on the Functional Requirements for Bibliographic Records 2009)).

It appears clear that the development of a set of models that aim to describe the main part of the publishing domain must pass through the adoption of established methodologies for ontology modularisation and development (Rector 2003; Shotton et al. 2010). These activities may be eventually supported by the use of statistical

S. Peroni, *Semantic Web Technologies and Legal Scholarly Publishing,*
Law, Governance and Technology Series 15, DOI 10.1007/978-3-319-04777-5_5,

clustering techniques, e.g., Cimiano and Volker (2005), though I did not explicitly used them in this work. Moreover, the following principles[1] should apply:

- There should be an extensive dialogue with publishers and members of academic communities to clarify their requirements.
- Any area of interest of the publishing domain (bibliographic description of documents, characterisation of citations, person's roles, etc.) should be covered by separate yet interoperable ontologies.
- Logical constraints, for example domain and range constraints on properties, should be added only where they are strictly required, to allow maximum reusability of each ontology module.
- Where well-known and widely shared vocabularies covering parts of the domain already exist, these should be properly imported and re-used.
- Alongside the development of the ontologies, tools that assist people to understand and to use each ontology with minimum effort, without having to know the specific technical language in which the ontology is implemented, should be used or opportunely developed.

In this chapter, I will describe the principles and architecture of eight ontologies, which enable the semantic description of bibliographic entities according to the publisher' language: **SPAR**, the *Semantic Publishing and Referencing Ontologies*[2], a suite of orthogonal and complementary OWL 2 DL ontology modules. These ontologies permit the creation of comprehensive machine-readable RDF metadata for every aspect of semantic publishing and referencing: document description; type of citation and related contexts; bibliographic references; document parts and status; agents' roles and workflow processes, etc.

Table 5.1 on page 123 shows briefly all the metadata schemas, vocabularies and ontologies (including SPAR)—some of which were introduced in Sect. 2.3, and others will be presented in the following sections—according to four main characteristics:

- *domain*, that indicates the main domain for which the model has been developed;
- *standard*, that indicates whether the model is acknowledged as a proper standard according to some international organisations;
- *RDF*, that indicates whether there exists an official/unofficial implementation of the model in RDF or RDFS;
- *OWL 2 DL*, that indicates whether there exits an official/unofficial implementation of the model that is compliant with OWL 2 DL.

In addition to the peculiarities introduced in Table 5.1, which sets SPAR as one of the few models developed originally for the description of bibliographic entities through OWL 2 DL, the main characteristics of SPAR, that distinguish it from previous contributions, are firstly the creation of ontologies of sufficient expressivity to

[1] All these principles are derived from my personal experience in developing ontologies for a specific domain (i.e., publishing) and for specific end-users (primarily, publishers and authors).

[2] The SPAR (Semantic Publishing and Referencing) Ontologies: http://purl.org/spar.

Table 5.1 The models, introduced in this book, that can be used for describing the publishing domain

Model	Domain	Standard	RDF	OWL 2 DL
BIBO	Bibliographic entities	No	Yes	No
Dublin core	Generic resources	Yes	Yes	no
FRBR	Bibliographic entities	Yes	Yes	Yes
MARC 21	Bibliographic entities	Yes	Yes	No
Medium-grained structure	Scientific scholarly articles	No	No	No
ORB	Scientific scholarly articles	No	Yes	Yes
PRISM	Bibliographic entities	Yes	Yes	No
RDA	Bibliographic entities	Yes	Yes	No
SRO	Scholarly articles	No	Yes	Yes
SKOS	Generic resources	Yes	Yes	Yes
SPAR	Bibliographic entities	No	Yes	Yes
SWAN	Scientific scholarly articles	No	Yes	Yes

meet the requirements of academic authors and publishers, and secondly the development of accompanying presentation technologies, *LODE* (introduced in Sect. 6.2) and *Graffoo* (presented in Sect. 6.4), that enable the ontologies to be easily understood by potential users such as academic researchers, publishers and librarians who, while expert in their own domains, lack skills in ontology modelling and knowledge formalisation.

The starting point for SPAR was version 1.6 of *CiTO*, the *Citation Typing Ontology*, described in Shotton et al. (2010). Despite the fact that this work was both preliminary and incomplete, it contained, within one single ontology, terms for handling bibliographic document descriptions, properties to enable the characterisation of citations, as well as terms which allowed the recording of the number of citations to a given article, both within the citing paper and globally.

A simple architectural diagram of the eight SPAR ontologies is shown in Fig. 5.1 on page 124. As the diagram indicates, the eight principal SPAR ontologies are supported by three other OWL 2 DL ontologies that the SPAR ontologies import as required–*FRBR* in OWL 2 DL, *DEO*, the *Discourse Elements Ontology*[3], and the *Error Ontology*[4]. They are also supported by the external *FOAF Essentials*[5] and *SWAN Collections*[6] ontologies, by three *Ontology Design Patterns* ontology modules (*Time-indexed situation*[7], *Sequence*[8], *Participation*[9]), and by the *Patterns Ontology*[10] for document structures.

[3] DEO, the Discourse Elements Ontology: http://purl.org/spar/deo.

[4] The Error Ontology: http://www.essepuntato.it/2009/10/error.

[5] FOAF essentials in OWL: http://purl.org/swan/2.0/foaf-essential.

[6] CO, the Collections Ontology: http://swan.mindinformatics.org/ontologies/1.2/collections.owl.

[7] *Time-indexed situation* pattern: http://www.ontologydesignpatterns.org/cp/owl/timeindexedsituation.owl.

[8] *Sequence* pattern: http://www.ontologydesignpatterns.org/cp/owl/sequence.owl.

[9] *Participation* pattern: http://www.ontologydesignpatterns.org/cp/owl/participation.owl.

[10] The Patterns Ontology: http://www.essepuntato.it/2008/12/pattern.

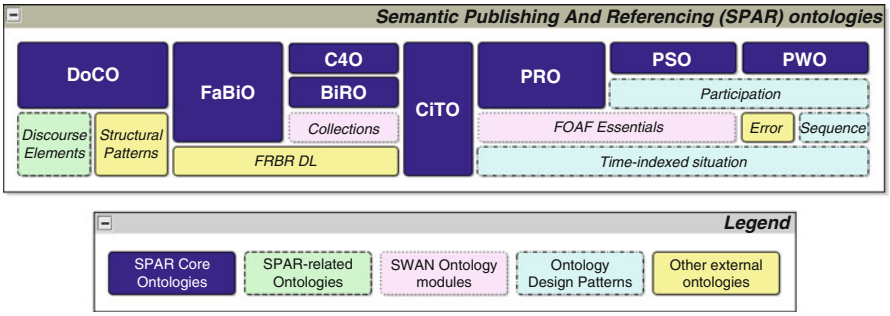

Fig. 5.1 A simple architectural diagram showing the interactions and dependencies between the component ontologies of SPAR

The characteristics and benefits of all the SPAR ontologies will be outlined in the following sections, which provide a comprehensive picture of the scope of SPAR. Where appropriate, I will also show how to integrate SPAR semantic data with documents defined through EARMARK (introduced in Chap. 3). The entire discussion will be supported by several examples of Turtle code (Prud'hommeaux and Carothers 2013) describing a particular reference of the legal discipline, i.e., Casanovas et al.'s "OPJK and DILIGENT: ontology modelling in a distributed environment" (Casanovas et al. 2007).

5.1 Representing Bibliographic Information Using FaBiO

The current well-known and commonly used vocabularies, which I described in Sect. 2.3, are either meagre in concepts or shallow, therefore preventing the description of publishing reality accurately. I will illustrate this by considering the representation of a typical bibliographic reference first using Dublin Core, then BIBO and finally FRBR. I will then show how this information can be accurately described using FaBiO, which incorporates elements of all these three vocabularies. Consider the following typical bibliographic reference describing (Casanovas et al. 2007):

> Pompeu Casanovas, Núria Casellas, Christoph Tempich, Denny Vrandečić, Richard Benjamins (2007). OPJK and DILIGENT: ontology modeling in a distributed environment. Artificial Intelligence and Law, 15 (2): 171–186. June (2007). Springer. DOI: 10.1007/s10506-007-9036-2. Print ISSN 0924-8463. Online ISSN 1572-8382. Published online (PDF) May 31, 2007.

From the previous description we can extract the following information:

1. The document is an academic research article—deducible from the journal in which it is published.
2. Pompeu Casanovas, Núria Casellas, Christoph Tempich, Denny Vrandečić, and Richard Benjamins are the authors of the article.
3. The article was published in 2007.
4. The article is entitled "OPJK and DILIGENT: ontology modeling in a distributed environment".
5. It was published in the 2nd issue of the 15th volume of *Artificial Intelligence and Law*.
6. The DOI of the article is "10.1007/s10506-007-9036-2".
7. The Print ISSN of the journal is "0924-8463".
8. The Online ISSN of the journal is "1572-8382".
9. The PDF version of the article was published online on May 31, 2007.
10. The journal issue within which the printed version of the article was published bears the publication date June 2007.
11. The page range of the article within the printed version is "171–186".
12. The publisher of the journal is Springer.

5.1.1 Bibliographic Reference Metadata Encoding Using DC Terms

In the following RDF encoding example[11], we attempt to describe all these facts using only terms from the DC Terms vocabulary (Dublin Core Metadata Initiative 2012):

[11] This and the following RDF encodings are written in Turtle (Prud'hommeaux and Carothers 2013).

```
@prefix : <http://www.example.com/> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix text: <http://purl.org/NET/mediatypes/text/> .
@prefix application: <http://purl.org/NET/mediatypes/
   application/> .

:opjk-and-diligent a dcterms:BibliographicResource
  ; dcterms:creator :casanovas , :casellas,
    :tempich, :vrandecic, :benjamins
  ; dcterms:title "OPJK and DILIGENT: ontology modeling
     in a distributed environment"
  ; dcterms:issued "2007"^^xsd:gYear
  ; dcterms:issued "2007-06"^^xsd:gYearMonth
  ; dcterms:identifier "doi:10.1007/s10506-007-9036-2"
  ; dcterms:extent [ a dcterms:SizeOrDuration
    ; dcterms:description "171-186" ]
  ; dcterms:hasFormat :pdf
  ; dcterms:isPartOf [ dcterms:identifier "2"
    ; dcterms:description "Issue"
    ; dcterms:isPartOf [ dcterms:identifier "15"
      ; dcterms:description "Volume"
      ; dcterms:isPartOf [
        dcterms:title
          "Artificial Intelligence and Law"
        ; dcterms:publisher :springer ] ] ] .

:pdf a dcterms:BibliographicResource
  ; dcterms:format application:pdf
  ; dcterms:issued "2007-05-31"^^xsd:date .

:casanovas a dcterms:Agent
  ; dcterms:description "Pompeu Casanovas" .
:casellas a dcterms:Agent
   ; dcterms:description "Nuria Casellas" .

:tempich a dcterms:Agent
   ; dcterms:description "Christoph Tempich" .

:vrandecic a dcterms:Agent
   ; dcterms:description "Denny Vrandečić" .

:benjamins a dcterms:Agent
   ; dcterms:description "Richard Benjamins" .

:springer a dcterms:Agent
   ; dcterms:description "Springer" .
```

There are some obscure points that emerge from the preceding formalisation:

- There is *no clear characterisation* of the entities involved. We are able to speak about a general "bibliographic resource" (*dcterms:BibliographicResource*) and an "agent" (*dcterms:Agent*), but not about a journal article, a journal, a volume, or an issue of a journal, nor about persons, authors, etc.
- Some of the statements are *too generic*. E.g., the property *dcterms:issued* that is used to represent the various dates associated with the publication of this article, is also employed in conjunction with three different date formats, e.g., `"2007-05-31"^^xsd:date`, `"2007-06"^^xsd:gYearMonth`, and `"2007"^^xsd:gYear`.
- Some of the statements *hide the semantics within the textual content* of the statement. E.g., the statement *dcterms:identifier "doi:10.1007/s10506-007-9036-2"* implicitly says that the character string "10.1007/s10506-007-9036-2" is a Digital Object Identifier, i.e., a special type of identifier used to identify journal articles. Similarly `"171–186"` implicitly says that the *printed version* (**only**) of the article starts at page "171" and ends at page "186". While these implied facts are understandable to human readers, they are not available to computational agents processing the metadata.
- The relationships between the various formats of the article are not clear. For example, the manner in which the resource "*:opjk-and-diligent*" relates to the resource "*:pdf*" is not specified. Do the latter represents the content of the former in a different format, or there is something more to it?

5.1.2 Bibliographic Reference Metadata Encoding Using BIBO

Some of these points are addressed by BIBO (D'Arcus and Giasson 2009). BIBO is the first OWL ontology specifically designed to address the domain under discussion, and expands the DC Terms vocabulary with terms which are specific for bibliographic metadata, with particular regards to legal documents, and for various types of event. It also includes PRISM (Hammond 2008) and FOAF (Brickley and Miller 2010) terms.

In the following RDF encoding example, the information given in the bibliographic reference cited above is encoded using BIBO:

```
@prefix bibo: <http://purl.org/ontology/bibo/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

:opjk-and-diligent a bibo:AcademicArticle
  ; bibo:authorList ( :casanovas :casellas
    :tempich :vrandecic :benjamins )
  ; dcterms:title "OPJK and DILIGENT: ontology modeling
     in a distributed environment"
  ; dcterms:issued "2007"^^xsd:gYear
  ; dcterms:issued "2007-06 "^^xsd:gYearMonth
  ; bibo:doi "10.1007/s10506-007-9036-2"
  ; bibo:pageStart "171"
  ; bibo:pageEnd "186"
  ; dcterms:hasFormat :pdf
  ; dcterms:isPartOf [ a bibo:Issue
    ; bibo:issue "2"
    ; bibo:volume "15"
    ; dcterms:isPartOf [ a bibo:Journal
      ; dcterms:title
        "Artificial Intelligence and Law"
      ; dcterms:publisher :springer ] ] .

:pdf a bibo:AcademicArticle
  ; dcterms:format application:pdf
  ; dcterms:issued "2007-05-31 "^^xsd:date .

:casanovas a foaf:Person
  ; foaf:givenName "Pompeu"
  ; foaf:familyName "Casanovas" .

:casellas a foaf:Person
  ; foaf:givenName "Nuria"
  ; foaf:familyName "Casellas" .

:tempich a foaf:Person
   ; foaf:givenName "Christoph"
   ; foaf:familyName "Tempich" .

:vrandecic a foaf:Person
   ; foaf:givenName "Denny"
   ; foaf:familyName "Vrandécic" .

:benjamins a foaf:Person
   ; foaf:givenName "Richard"
   ; foaf:familyName "Benjamins" .

:springer a foaf:Organization
   ; foaf:name "Springer" .
```

As this example shows, BIBO resolves many of the semantic ambiguities present in the DC version–the DOI is specifiable through the specific data property *bibo:doi*; the article is identified as a *bibo:AcademicArticle*; the authors and the publisher are respectively *foaf:Persons* and *foaf:Organization*, etc. However, other ambiguities are still unresolved. The relationships between the various formats are still not clear, and the date properties continue to be too generic. In addition, new issues emerge:

- BIBO specifies that the property for listing authors (*bibo:authorList*) must have, as its range, either an *rdf:List* or an *rdf:Seq*.Since these RDF classes are not supported by OWL 2, this has the disadvantage of making that model non-compliant with the decidable and computable OWL 2 DL, and thus preventing OWL 2 DL reasoners from inferring new axioms from a current knowledge base encoded using BIBO[12].
- BIBO can record a volume number through the data property *bibo:volume*, but, although BIBO has the classes *bibo:AcademicArticle*, *bibo:Issue* and *bibo:Journal*, it lacks the concept of "Volume" as a distinct class among other bibliographic classes that have a hierarchical partitive relationship to one another (e.g., Journal Article > Issue > Volume > Journal).
- Furthermore, because it lacks the layered structure of FRBR (International Federation of Library Associations and Institutions Study Group on the Functional Requirements for Bibliographic Records 2009), it does not have the flexibility to distinguish between concepts at these various levels, for example an academic paper (a FRBR Work) and the various possible Expressions of that paper as a journal article, a conference paper or a book chapter. The class *bibo:AcademicArticle* is in fact a conflation of the concepts "academic paper" and "journal article".

5.1.3 Bibliographic Reference Metadata Encoding Using FRBR

It is possible to resolve the third of the issues raised above by adopting the more structured FRBR model (International Federation of Library Associations and Institutions Study Group on the Functional Requirements for Bibliographic Records 2009), as expressed in the FRBR Core ontology, together with DC terms for textual statements (i.e., those statements having a literal string as their object). This is illustrated in the following example:

[12] For a more detailed explanation of why RDF collections and containers are neither usable nor interpreted correctly by OWL 2 DL, consult http://hcklab.blogspot.com/2008/12/moving-towards-swan-collections.html.

```
@prefix frbr: <http://purl.org/vocab/frbr/core#> .

:opjk-and-diligent a frbr:Work
  ; frbr:creator :casanovas , :casellas,
    :tempich, :vrandecic, :benjamins
  ; dcterms:title "OPJK and DILIGENT: ontology modeling
     in a distributed environment"
  ; frbr:realization :version-of-record .

:version-of-record a frbr:Expression
  ; dcterms:issued "2007"^^xsd:gYear
  ; dcterms:identifier "doi:10.1007/s10506-007-9036-2"
  ; frbr:embodiment :printed , :pdf
  ; frbr:partOf [ a frbr:Expression
    ; dcterms:identifier "2"
    ; dcterms:description "Issue"
    ; frbr:embodiment :printed-issue
    ; frbr:partOf [ a frbr:Expression
      ; dcterms:identifier "15"
      ; dcterms:description "Volume"
      frbr:partOf [ a frbr:Expression
        ; dcterms:title
          "Artificial Intelligence and Law" ] ] ] .

:printed-issue a frbr:Manifestation
  ; frbr:producer :springer
  ; dcterms:issued "2007-06"^^xsd:gYearMonth
  ; frbr:part :printed .

:printed a frbr:Manifestation
   ; frbr:producer :springer
   ; dcterms:issued "2007-06"^^xsd:gYearMonth
   ; dcterms:extent [ a dcterms:SizeOrDuration
     ; dcterms:description "171-186" ] .

:pdf a frbr:Manifestation
   ; frbr:producer :springer
   ; dcterms:format application:pdf
   ; dcterms:issued "2007-05-31"^^xsd:date .

:casanovas a frbr:Person
   ; dcterms:description "Pompeu Casanovas" .

:casellas a frbr:Person
   ; dcterms:description "Nuria Casellas" .

:tempich a frbr:Person
   ; dcterms:description "Christoph Tempich" .
```

```
:vrandecic a frbr:Person
   ; dcterms:description "Denny Vrandečić" .

:benjamins a frbr:Person
   ; dcterms:description "Richard Benjamins" .

:springer a frbr:CorporateBody
   ; dcterms:description "Springer" .
```

Although it is possible to use FRBR in this manner to give a structured and unambiguous description of all the bibliographic entities, the example makes it clear the severe limitations of FRBR. These are caused by the lack of terms in the FRBR Core ontology which allow publications to be described in normal everyday language.

5.1.4 Bibliographic Reference Metadata Encoding Using FaBiO

FaBiO, the *FRBR-aligned Bibliographic Ontology*[13] (Peroni and Shotton 2012), was developed precisely to address all the issues raised by the previous examples, while re-using the previous fundamental work in this domain (so as not to re-invent the wheel). In particular, DC Terms, PRISM, FRBR and SKOS terms are all included in FaBiO.

Considering again the previous bibliographic reference example, a possible FaBiO formalisation would be:

[13] FaBiO, the FRBR-aligned Bibliographic Ontology: http://purl.org/spar/fabio.

```
@prefix fabio: <http://purl.org/spar/fabio> .
@prefix prism: <http://prismstandard.org/namespaces/
   basic/2.0/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

:opjk-and-diligent a fabio:ResearchPaper
  ; dcterms:creator :casanovas , :casellas,
    :tempich, :vrandecic, :benjamins
  ; dcterms:title "OPJK and DILIGENT: ontology modeling
     in a distributed environment"
  ; frbr:realization :version-of-record .

:version-of-record a fabio:JournalArticle
  ; fabio:hasPublicationYear "2007"^^xsd:gYear
  ; prism:doi "10.1007/s10506-007-9036-2"
  ; frbr:embodiment :printed , :pdf
  ; frbr:partOf [ a fabio:JournalIssue
    ; prism:issueIdentifier "2"
    ; frbr:embodiment :printed-issue
    ; frbr:partOf [ a fabio:JournalVolume
      ; prism:volume "15"
      frbr:partOf [ a fabio:Journal
        ; dcterms:title
          "Artificial Intelligence and Law" ] ] ] .

:printed-issue a fabio:Paperback
  ; dcterms:publisher :springer
  ; prism:publicationDate "2007-06"^^xsd:gYearMonth
  ; frbr:part :printed .

:printed a fabio:PrintObject
  ; dcterms:publisher :springer
  ; prism:publicationDate "2007-06"^^xsd:gYearMonth
  ; prism:startingPage "171"
  ; prism:endingPage "186" .

:pdf a fabio:DigitalManifestation
  ; dcterms:publisher :springer
  ; dcterms:format application:pdf
  ; prism:publicationDate "2007-05-31"^^xsd:date .
```

```
:casanovas a foaf:Person
   ; foaf:givenName "Pompeu"
   ; foaf:familyName "Casanovas" .

:casellas a foaf:Person
   ; foaf:givenName "Nuria"
   ; foaf:familyName "Casellas" .

:tempich a foaf:Person
   ; foaf:givenName "Christoph"
   ; foaf:familyName "Tempich" .

:vrandecic a foaf:Person
   ; foaf:givenName "Denny"
   ;  foaf:familyName "Vrandečić" .

:benjamins a foaf:Person
   ; foaf:givenName "Richard"
   ; foaf:familyName "Benjamins" .

:springer a foaf:Organization
   ; foaf:name "Springer" .
```

With FaBiO, it thus becomes possible:

- To write semantic descriptions of a wide variety of bibliographic objects, including research articles, journal articles and journal volumes, using terms that closely resemble the language used in everyday speech by academics and publishers[14].
- To employ FRBR categories to define clear separations between each part of the publishing process, which involves different people (authors, publishers, readers), depending on which aspect of the bibliographic entity we are considering: the high-level conceptualisation of the research paper, the version of record of that paper forming a journal article: the publication of the article in various formats, and the individual physical or electronic exemplars of the published article that people may read and own.
- To include with ease elements from other vocabularies which describe particular entities involved in the publishing process that are not specified by FaBiO itself, such as those from FOAF for persons and organisations.

Other advantages of FaBiO will be outlined in the following sections.

[14] This has been achieved through many meetings with a number of academics and publishers that we have undertaken in order to understand their working practices and requirements.

5.1.4.1 Using External Models

As already mentioned, FaBiO has been developed so to limit any restriction to its classes as well as the domains and ranges of its properties. This flexibility has the great advantage of allowing FaBiO to be used together with other models. We have already seen how FOAF can be used to describe agents. Another common requirement would be to specify the order of components in a list, e.g., authors in an author list. This can be achieved in a manner that is compliant with the decidable and computable OWL 2 DL, unlike the *bibo:authorList* as described above, by combining FaBiO with the *Collections Ontology* (*CO*)[15] (Ciccarese and Peroni 2013), an OWL 2 DL ontology specifically designed for defining orders among items. In particular:

```
@prefix co: <http://purl.org/co/> .

:opjk-and-diligent a fabio:ResearchPaper
  ; dcterms:creator :listOfAuthors .

:listOfAuthors a co:List
  ; co:firstItem [ co:itemContent :casanovas
  ; co:nextItem [ co:itemContent :casellas
  ; co:nextItem [ co:itemContent :tempich
  ; co:nextItem [ co:itemContent :vrendecic
  ; co:nextItem [ co:itemContent :benjamins ] ] ] ] ] .

:casanovas a foaf:Person
  ; foaf:givenName "Pompeu"
  ; foaf:familyName "Casanovas" .

:casellas a foaf:Person
  ; foaf:givenName "Nuria"
  ; foaf:familyName "Casellas" .

:tempich a foaf:Person
  ; foaf:givenName "Christoph"
  ; foaf:familyName "Tempich" .

:vrandecic a foaf:Person
  ; foaf:givenName "Denny"
  ; foaf:familyName "Vrandečić" .

:benjamins a foaf:Person
  ; foaf:givenName "Richard"
  ; foaf:familyName "Benjamins" .
```

In this way we can still keep the model in OWL 2 DL. Additionally, because the ranges of *dcterms:creator* and other properties within FaBiO have intentionally been left unspecified, FaBiO guarantees a level of interoperation with other models without incurring in any undesirable side effects, such as ontology inconsistencies or generation of undesired inferences.

[15] CO, the Collections Ontology: http://purl.org/co.

5.1.4.2 Extending FRBR Within FaBiO

One of the explicit requests from publishers and end-users was to be able to create shortcuts between FRBR endeavours (work, expression, manifestation, item) that were not part of the original FRBR model. Let me introduce an example to illustrate this requirement, by marginally changing the bibliographic reference we introduced earlier:

> Pompeu Casanovas, Núria Casellas, Christoph Tempich, Denny Vrandečić, Richard Benjamins (2007). OPJK and DILIGENT: ontology modeling in a distributed environment. http://link.springer.com/content/pdf/10.1007%2Fs10506-007-9036-2.pdf.

In this reference, we have one FRBR work—the paper by Casanovas et al.—and the URL for a specific FRBR item that portrays that work—the PDF version of the paper on the publishers' website. If I wished to link these concepts using the FRBR OWL ontology terms I have employed so far, I would be obliged to specify each intermediate FRBR endeavour, namely the expression and manifestation of that paper, even if we were not interested in doing that:

```
@prefix springer: <http://link.springer.com/content/pdf/> .
:opjk-and-diligent a frbr:Work
  ; frbr:creator :casanovas , :casellas,
    :tempich, :vrandecic, :benjamins
  ; dcterms:title "OPJK and DILIGENT: ontology modeling
     in a distributed environment"
  ; frbr:realization [ a frbr:Expression
    ; frbr:embodiment [ a frbr:Manifestation
      ; frbr:exemplar springer:10.1007%2Fs10506
         -007-9036-2.pdf ] ] .
```

To avoid this long-windedness, it is possible to use the new FaBiO properties, shown in Fig. 5.2[16], to link a work directly to its manifestations *(fabio:hasManifestation)* or to its items (*fabio:hasPortrayal*), or to link an expression directly to its items (*fabio:hasRepresentation*).

Evidently, these added properties allows us to treat these cases quite easily and in a less verbose way:

```
@prefix springer: <http://link.springer.com/content/pdf/> .
:opjk-and-diligent a frbr:Work
  ; frbr:creator :casanovas , :casellas,
    :tempich, :vrandecic, :benjamins
  ; dcterms:title "OPJK and DILIGENT: ontology modeling
     in a distributed environment"
  ; fabio:hasPortrayal springer:10.1007%2Fs10506
     -007-9036-2.pdf .
```

[16] This and the following diagrams comply with the *Graphic framework for OWL ontologies* (*Graffoo*), introduced in Sect. 6.4. A legend for all Graffoo diagrams can be found in Fig. 6.13 on page 227.

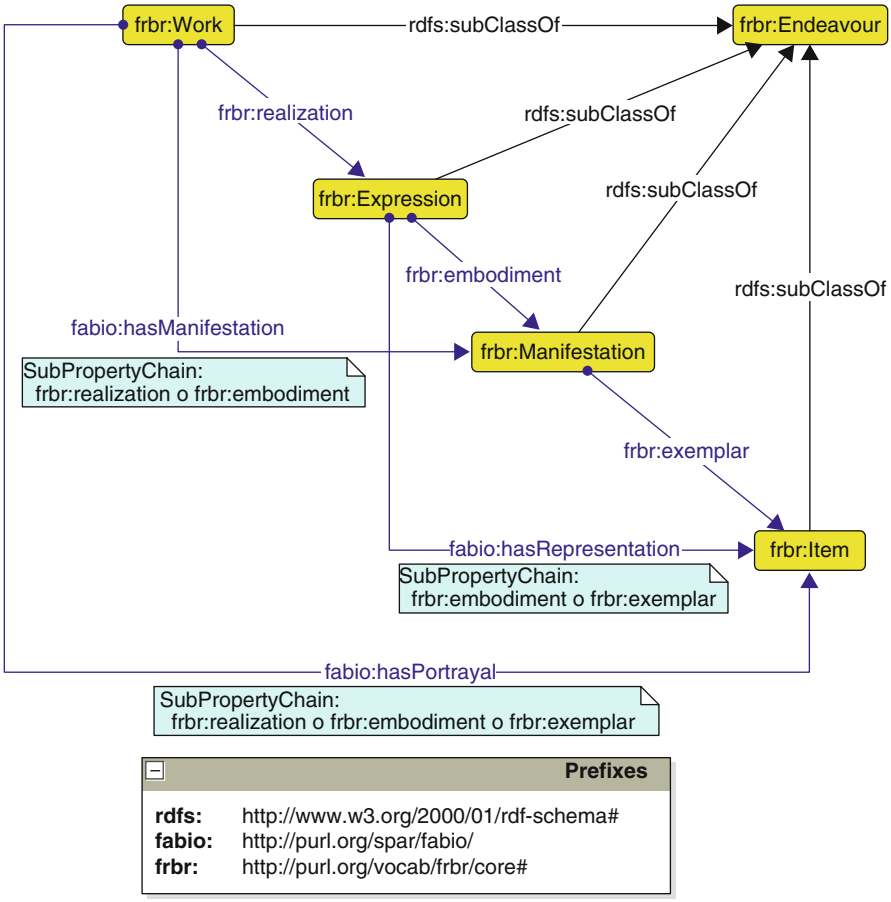

Fig. 5.2 The main FRBR object properties relating FRBR endeavours (work, expression, manifestation, item), and the related new object properties introduced by FaBiO (fabio:hasManifestation, fabio:hasRepresentation, fabio:hasPortrayal) to provide shortcuts between Work and Manifestation, Work and Item, and Expression and Item, respectively

5.1.4.3 Categorising Bibliographic Resources with SKOS

One of the most important needs for a publisher is to categorise each bibliographic entity it produces by adding free-text keywords and/or specific terms structured according to recognised classification systems and/or thesauri developed for specific academic disciplines. While through FaBiO the definition of keywords is possible using the PRISM property *prism:keyword*, terms from thesauri, structured vocabularies and classification systems are described using SKOS (Miles and Bechhofer 2009).

To facilitate this, FaBiO extends some classes and properties of SKOS as shown in Fig. 5.3.

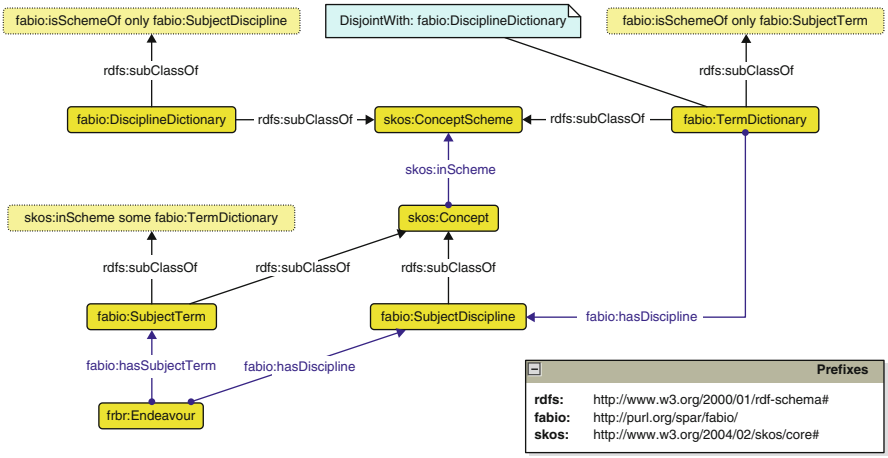

Fig. 5.3 The extension to the common SKOS classes and relations implemented in FaBiO

As shown, any FRBR endeavour can be associated (*fabio:hasSubjectTerm*) with one or more descriptive terms (*fabio:SubjectTerm*, a sub-class of*skos:Concept*) found in a specific dictionary (*fabio:TermDictionary*, a sub-class of *skos:ConceptScheme*) that is relevant to (*fabio:hasDiscipline*) particular disciplines (*fabio:SubjectDiscipline*, also a sub-class of*skos:Concept*) describing a field of knowledge or human activity such as computer science, biology, economics, cookery or swimming. At the same time, the subject disciplines can be grouped by an opportune vocabulary (i.e., *fabio:DisciplineDictionary*).

For instance, the previous example can be enriched in this way:

```
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix facet: <http://link.springer.com/facet/> .
@prefix dbpedia: <http://dbpedia.org/resource/> .

:opjk-and-diligent a fabio:ResearchPaper
  ; fabio:hasSubjectTerm facet:air-and-space-law ,
    facet:computational-linguistics ,
    facet:philosophy-of-law ,
    facet:legal-aspects-of-computing ,
    facet:artificial-intelligence-incl-robotics
  ; prism:keywords "legal ontologies" , "methodology" ,
    "ontology modeling" , "professional knowledge" ,
    "rhetorical structure theory" .

<http://link.springer.com/facet> a fabio:TermDictionary
  ; skos:prefLabel "Facet dictionary used in Springer
     library"
  ; fabio:hasDiscipline dbpedia:Computer_science ,
     dbpedia:Law .

facet:air-and-space-law a fabio:SubjectTerm
  ; skos:prefLabel "Air and Space Law"
  ; skos:inScheme <http://link.springer.com/facet> .

computational-linguistics a fabio:SubjectTerm
  ; skos:prefLabel "Computational Linguistics"
  ; skos:inScheme <http://link.springer.com/facet> .
...
```

5.2 Characterising Citations with CiTO

Bibliographic citation, i.e., the act of referring from a citing entity to the cited one, is one of the most important activities of an author in the production of any bibliographic work, since the acknowledgement of the sources used by the author stands at the very core of the scholarly enterprise. The network of citations created by combining citation information from many academic articles and books, is a source of rich information for scholars, and can be used by publishers to create new and interesting ways of browsing their material, as well as for calculating metrics which reflect the relative importance of a journal (i.e., the *impact factor*) or an author (i.e., the *h-index*).

The reasons why an author cites other publications are varied. Usually, it is because the author has gained assistance of some sort, perhaps in the form of background information, ideas, methods or data, from a previously published work, and they

wish to acknowledge this. More rarely, citations may be made to review, critique or refute previous work. Most citations are direct and explicit (as in the reference list of a journal article). However, they can also be indirect (for example, by means of a citation to a more recent paper by the same research group on the same topic), or implicit (as in artistic quotations or parodies, or *in extremis*in cases of plagiarism).

Traditionally, scholars had well-developed methods for citing individual sections, paragraphs or verses of referenced works. In addition, it was not uncommon for the citing author to reproduce entire sections of the cited work in their own document, so that the reader could understand exactly the relationship of the earlier document to the present one, since the author could not be sure the reader would have ready access to the works of the cited authority (for example, Aristotle). In contrast, modern scientific practice takes the other extreme–a citation is made to the previously published paper as a whole, with little or no indication given as to why this paper has been cited or what portions of it are relevant to the discussion in hand, except for what the reader can glean from the citation context.

Previously developed models for describing bibliographic objects would normally allow for the existence of citations among bibliographic entities to be recorded. For instance, considering again the example of the article "OPJK and DILIGENT: ontology modeling in a distributed environment", using BIBO (D'Arcus and Giasson 2009) it is possible to declare citations as follows:

```
@prefix bibo: <http://purl.org/ontology/bibo/> .

:opjk-and-diligent bibo:cites
    :principles-design-ontologies
  , :ontological-engineering
  , :ontology-integration ...
```

Alternatively, it is possible to use the Discourse Relationships Module[17] in SWAN v1.2 (Ciccarese et al. 2008) in the same way:

```
@prefix disrel: <http://swan.mindinformatics.org/
   ontologies/1.2/discourse-relationships/> .

:version-of-record disrel:cites
    :principles-design-ontologies
  , :ontological-engineering
  , :ontology-integration ...
```

However, while the *cites* properties in these two ontologies, as well as the more generic property *dcterms:relation* in DC Terms, permit the bald **existence** of a citation to be recorded, they do not permit the author to invest the citation with any specific

[17] The Discourse Relationships Ontology: http://swan.mindinformatics.org/spec/1.2/discourserelationships.html.

factual and/or rhetorical meanings that would describe the **reasons** why the author decided to create such citation.

CiTO, the *Citation Typing Ontology* [18] version 2.0 (Peroni and Shotton 2012), seeks to improve upon this situation by making it possible for authors (or others) to capture their intent when citing a particular publication, as it allows them to create metadata describing citations that are distinct from metadata describing the cited works themselves. CiTO thus permits the motives of an author when referring to another document to be captured. The previous example would therefore be written using CiTO as follows:

```
@prefix cito: <http://purl.org/spar/cito/> .
:version-of-record
 cito:includesQuotationFrom
    :principles-design-ontologies
; cito:usesMethodIn :ontological-engineering
; cito:citesAsRelated :ontology-integration ...
```

The current version of CiTO, version 2.6.2, contains and extends the citation-specific object properties that were originally contained in CiTO version 1.6 (Shotton 2010), to the exclusion of those other original classes and properties within CiTO v1.6, which, as part of the modularisation we have undertaken, have been moved into FaBiO (Sect. 5.1) or into C4O (Sect. 5.3.2) and PSO (Sect. 5.5.7).

CiTO now contains just two main object properties, *cito:cites* and its inverse *cito:isCitedBy*, each of which has 41 sub-properties, plus four additional generic object properties–i.e. *cito:shareAuthorsWith*, *cito:sharesAuthorInstitutionWith*, *cito:sharesFundingAgencyWith* and *cito:likes*–that may be used even outside a citation act. As defined in *Functions of Citations* ontology[19], all these properties (and, consequently, their inverses) may be classified as rhetorical and/or factual, with the rhetorical properties being grouped in three sets depending on their connotation: *positive*, *informative* (or *neutral*) or *negative*.

When developing CiTO v2.6.2 from CiTO v1.6, we intentionally removed the domain and range constraints from the object properties, so that this ontology could be easily integrated with other models. Obviously, it can be successfully used in conjunction with FaBiO, so that descriptions of a bibliographic entity and its citations can be mixed within a single RDF graph:

[18] CiTO, the Citation Typing Ontology: http://purl.org/spar/cito.

[19] Functions of Citations ontology: http://www.essepuntato.it/2013/03/cito-functions.

```
:version-of-record a fabio:JournalArticle
  ; frbr:realizationOf :opjk-and-diligent
  ; cito:includesQuotationFrom
    :principles-design-ontologies
  ; cito:usesMethodIn :ontological-engineering
  ; cito:citesAsRelated :ontology-integration ...

:principles-design-ontologies a fabio:JournalArticle
  ; frbr:realizationOf [ a fabio:ResearchPaper
    ; dcterms:title "Toward principles for the design of
        ontologies used for knowledge sharing"
    ; dcterms:creator :gruber ]
  ; cito:providesQuotationFor :version-of-record .

:ontological-engineering a fabio:Book
  ; frbr:realizationOf [ a fabio:ReferenceWork
    ; dcterms:title "Ontological Engineering"
    ; dcterms:creator :gomez-perez , :fernandez-lopez ,
       :corcho ]
  ; cito:providesMethodFor :version-of-record .

:ontology-integration a fabio:ConferencePaper
  ; frbr:realizationOf [ a fabio:ResearchPaper
    ; dcterms:title "Ontology integration: Experiences
       with medical terminologies"
    ; dcterms:creator :gangemi , :pisanelli , :steve ]
  ; cito:isCitedAsRelatedBy :version-of-record .
...
```

5.3 Documents and Their Bibliographic References

The word "citation" is often subject of misinterpretations and misuse. The reason being that the word can be used to identify objects which have different purposes, at least in scientific literature. For instance, we often identify as "citation" the *act of citing* another work, a *bibliographic reference* put at the end of a paper (usually in a list), as well as particular *pointers* (e.g., "[3]") denoting that bibliographic reference.

In order to expand more on this topic, let we consider the following text from the article "OPJK and DILIGENT: ontology modeling in a distributed environment" (Casanovas et al. 2007) used in the previous examples:

> **DILIGENT Methodology**
>
> ...
>
> an ontology is defined as 'a shared specification of a conceptualization' (Gruber 1995). Although 'shared' is an essential feature, it is often neglected. In DILIGENT, experts exchange arguments while building the initial shared ontology in order to reach consensus;
>
> ...
>
> **References**
>
> ...

Gruber (1995). Toward principles for the design of ontologies used for knowledge sharing, International Journal of Human Computer Studies, 43(5–6): 907–928.

The above excerpt contains a particular paragraph from the section "DILIGENT Methodology" of the paper and a list item from the final "References" section. We can clearly identify six different kinds of objects that relate to this citation, all of them having different purposes:

1. The *citing article*, i.e., the article that contains such a text.
2. The *cited article*, i.e., the article that is being referred to by the text.
3. The *in-text reference pointer* which refers to a particular bibliographic reference (usually contained in the section "References"), e.g., the text "(Gruber 1995)". In scientific literature this can be presented in different forms—as an in-square-brackets number (e.g., "[3]"); as an in-square-brackets string with the first letter of each (at most three) authors' surname plus the last two digits of the publication year (e.g., "[RDS02]"); or as an in-round-brackets string with the first author's surname followed by "et al.," and the publication year (e.g., "(Renear et al. 2002)");
4. The *citation context*, the unit of the paper (sentence, paragraph, section, chapter, etc.) in which the in-text reference pointer appears.
5. The *bibliographic reference*, i.e., the list item at the end of the above excerpt that briefly summarises some metadata of a particular paper. It is explicitly denoted (by an in-text reference pointer) somewhere in the text.
6. The *act of citing*, i.e., a statement that connects two different articles (more precisely, the *citing document* and the *cited document*) for a particular reason, as described in Sect. 5.2.

Ontologies that aim to describe such elements should be provided with appropriate entities (classes and properties) in order to prevent or minimise any ambiguity when modelling citing acts in documents.

Having a clear and unambiguous description of elements used in citations is particularly relevant for those applications that extract citation contexts in an automatic and semi-automatic ways. For instance, the *Citation-Sensitive In-Browser Summariser* (*CSIBS*) (Wan et al. 2010) is a tool for presenting a preview of possible excerpts from the cited document that are relevant to a particular in-text reference pointer in the citing document. One of this tool's expected future development would be to enable RDF-based descriptions of these elements.

In Sect. 5.2 I introduced the SPAR ontology for the description of factual and rhetorical aspects of citations. In the next section I will present two models which focus on the remaining aspects of the citing acts: the *Bibliographic Reference Ontology* (*BiRO*) and the *Citation Counting and Context Characterisation Ontology*(*C4O*). They are two SPAR ontologies developed for describing bibliographic lists, bibliographic references, in-text reference pointers, citation contexts and a mechanism for counting locally (within an article) or globally (by means of particular platforms, such as Google Scholar[20]) document citations.

[20] Google Scholar: http://scholar.google.com.

5.3.1 *Describing the Bibliographic Reference Lists of Articles with BiRO*

According to Shotton (2009), one of the rules that digital publishers should follow to participate actively to Semantic Publishing is at least to make available reference lists of articles in a machine-readable form. In principle, reference lists are the platform from where citation networks should be built. In order to accomplish this aim, ontologies which model article references and reference lists are needed. Besides offering flexible mechanisms for the description of references, these ontologies should also allow the user to link the reference to the particular semantic representation of the document being *referenced*.

This is an important point to understand: the reference in the reference list of a particular article **is not** the cited document. Rather, it is a compact description considered (usually) sufficient to make the reader aware of what document has been cited. Therefore, having references expressed in a machine-readable form should allow machines to make the inference step, i.e., to link automatically the article containing the reference (i.e., the citing document) to the article referenced by the reference itself (i.e., the cited document).

I have developed the *Bibliographic Reference Ontology*[21] (*BiRO*) to offer a standard model for the description of reference lists and references according to (machine-readable) Semantic Web standards. In particular, BiRO is an ontology structured according to the FRBR model Sect. 2.3.5 to define bibliographic records (as subclasses of FRBR Work) in relation to bibliographic references (as subclasses of FRBR Expression), and their compilation into bibliographic collections and ordered bibliographic lists, respectively (as shown in Fig. 5.4).

An individual bibliographic reference, such as one in the reference list of a published journal article, may exhibit varying degrees of incompleteness, depending on the formatting rules of the journal. For example, it may lack the title of the cited article, the full names of the listed authors, or indeed a full listing of all the authors. It will also lack other information that one would expect to find in the complete bibliographic description for that article.

BiRO provides a logical system for relating such an incomplete bibliographic reference:

- To the full bibliographic record for that cited article, which, in addition to any author and title fields missing from the reference, may also be expected to include the name of the publisher, and the ISSN or ISBN of the publication.
- To collections of bibliographic records, such as library catalogues.
- To ordered bibliographic lists, such as reference lists.

In order to understand how to use BiRO to describe reference lists, let use again the reference introduced above which refers to (Gruber 1995):

[21] BiRO, the Bibliographic Reference Ontology: http://purl.org/spar/biro.

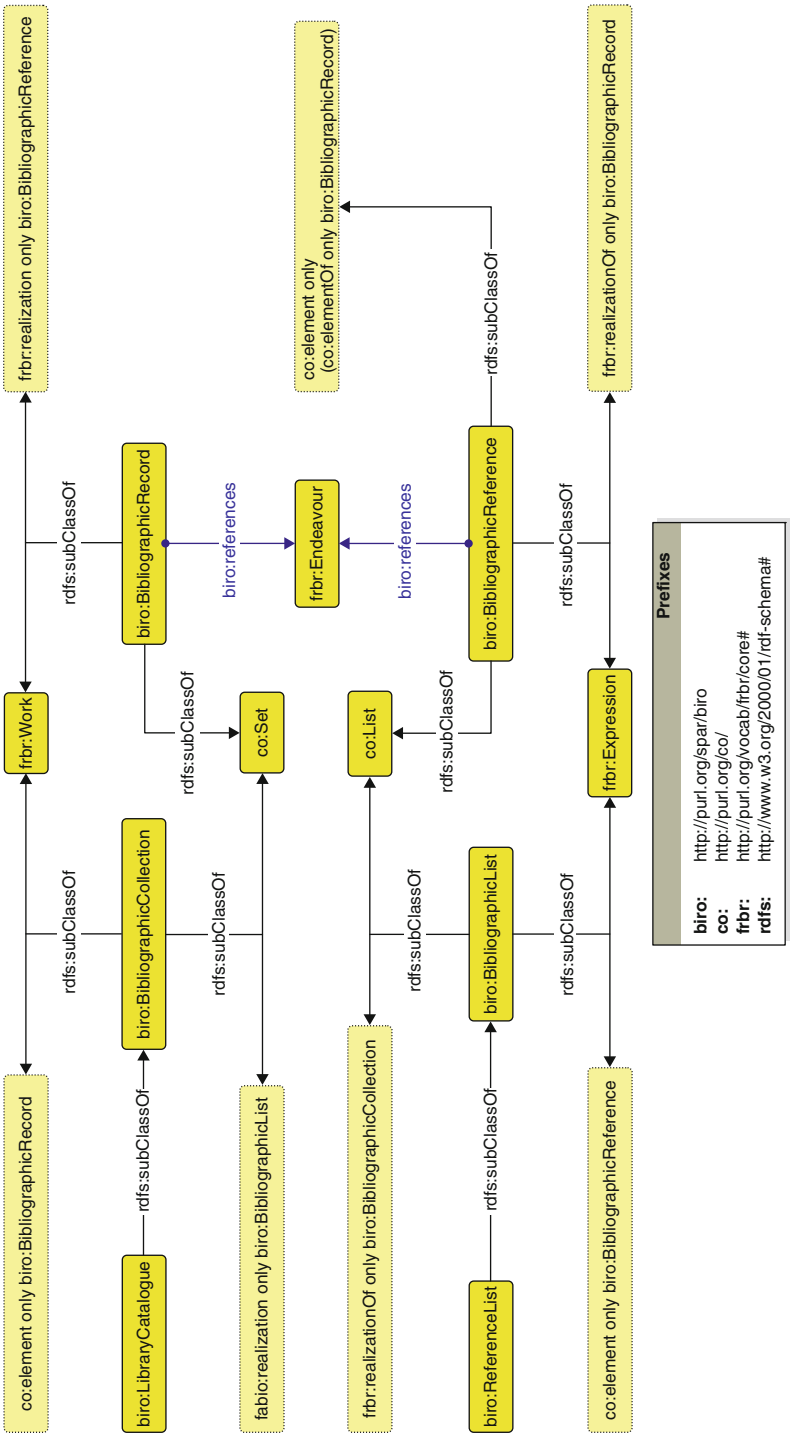

Fig. 5.4 Graffoo diagram summarising the Bibliographic Reference Ontology (*BiRO*)

Gruber (1995). Toward principles for the design of ontologies used for knowledge sharing, International Journal of Human Computer Studies, 43(5–6): 907–928.

5.3.1.1 An URI for the Reference

A first, very quick, way for defining a simple machine-readable representation of that reference using BiRO is as follows[22]:

```
    :version-of-record frbr:part :reference-list .

    :reference-list a biro:ReferenceList
      ; co:firstItem [ co:itemContent :ayuso03
      ; co:nextItem [ co:itemContent :benjamins04 ...
      ; co:nextItem [
        co:itemContent :gruber95 ... ] ... ] ] .
:gruber95 a biro:BibliographicReference
  ; biro:references :principles-design-ontologies
  ; dcterms:bibliographicCitation "Gruber, T. R. (1995).
      Toward principles for the design of ontologies
      used for knowledge sharing, International Journal
      of Human Computer Studies, 43(5-6): 907-928." .
```

Evidently, this formal description does not achieve a substantial improvement, as I only assigned an URL to the reference list and to each of its references. The semantics beyond the string representing the reference is still obscure. For instance, at this stage I do make clear that the strings "Gruber", "T. R.", "1995", "Toward principles for the design of ontologies used for knowledge sharing" are, respectively, the surname of one of the authors, the first letters of his names, the year of publication and the title of the article.

5.3.1.2 Semantic Enhancement of Literal Elements in References

A way to enable the semantic enhancement of strings is to use literals as subjects of statements and assertions, which is not allowed by Semantic Web standards such as RDF and OWL. Recently, within the Semantic Web community, this topic, i.e., whether and how to allow literals to be subjects of RDF statements[23], has been actively discussed. However, this discussion has still failed to provide a unique and clear indication of how to proceed in that regard.

Although one of the suggestions coming out of the discussion was to explicitly include the proposal in a (future) specification of RDF, this idea is not in fact new,

[22] The prefix *swan* refers to entities defined in the old version of the Collection Ontology (SWAN Collection Ontology 1.2), currently imported in BiRO. The SWAN Collection Ontology is available at http://swan.mindinformatics.org/ontologies/1.2/collections.owl.

[23] Literals as subjects: http://www.w3.org/2001/sw/wiki/Literals_as_Subjects.

particularly in ontology modelling. The need to describe "typical" literals (especially strings) as individuals of a particular class has been addressed by many models in past, such as Common Tag[24] (through the class *Tag*), SIOC (Bojars and Breslin 2010) (through the classes *Category* and *Tag*), SKOS-XL (Miles and Bechhofer 2009) (through the class *Label*), and LMM (Picca et al. 2008) (through the class *Expression*). After considering the above-mentioned models, among others, I have developed–in collaboration with Aldo Gangemi and Fabio Vitali–a pattern called *literal reification*to address this issue. It allows one to express literal values as proper ontological individuals so that they can be used as subject/object of any assertion within OWL models.

By extending the pattern *region*[25] (Gangemi 2010b), the pattern *literal reification*[26] (Gangemi et al. 2010) promotes any literal as "first class object" in OWL by reifying it as a proper individual of the class *litre:Literal*. Individuals of this class express literal values through the functional data property *litre:hasLiteralValue* and can be connected to other individuals that share the same literal value by using the property *litre:hasSameLiteralValueAs*. Moreover, a literal may refer to, and may be referred by, any OWL individual through *litre:isLiteralOf* and *litre:hasLiteral* respectively.

It is to be noted that the pattern defines also a SWRL rule (Horrocks et al. 2004) that allows one to infer the (not explicitly asserted) literal value of a particular literal individual when it is connected to another literal individual via *litre:hasSameLiteralValueAs*:

```
litre:hasSameLiteralValueAs(x,y) ,
litre:hasLiteralValue(y,v)
  -> litre:hasLiteralValue(x,v)
```

This pattern allows one to use each reified literal as subject or object of any assertion, and it is able to address scenarios described, for example, by the following competency questions:

- What is the context in which entities refer to a particular literal value?
- What is the meaning of a particular value considering the context in which it is used?

Plausible scenarios of its application include:

- Modelling domains concerning descriptive tags, in which each tag may have more than one meaning depending on the context in which it is used.

[24] Common Tag: http://www.commontag.org.

[25] Region pattern: http://ontologydesignpatterns.org/cp/owl/region.owl. The prefix *region* refers to entities defined in it.

[26] Literal reification pattern: http://www.essepuntato.it/2010/06/literalreification. The prefix *literal* refers to entities defined in it.

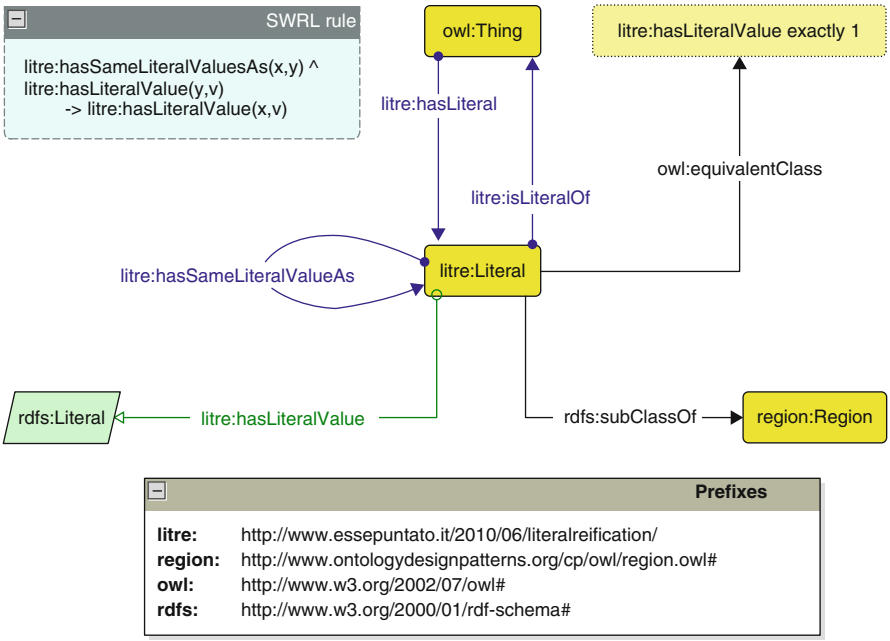

Fig. 5.5 Graffoo diagram summarising the *Literal Reification* pattern

- Extending the capabilities of a model by adding the possibility to make assertions on values, previously referred through data properties, without modifying it.

As briefly introduced above and also shown in Fig. 5.5, the pattern *literal reification* is composed by a class, a data property and three object properties, which are described as follows:

- class *litre:Literal*. It describes reified literals, where the literal value they represent is specified through the property *litre:hasLiteralValue*. Each individual of this class must always have a specified value;
- data property *litre:hasLiteralValue*. It is used to specify the literal value that an individual of *litre:Literal* represents;
- object property *litre:hasSameLiteralValueAs*. It relates the reified literal to another one that has the same literal value;
- object property *litre:hasLiteral*. It connects individuals of any class to a reified literal;
- object property *litre:isLiteralOf*. It connects the reified literal to the individuals that are using it.

By means of this pattern and of the OWL 2 capabilities in meta-modelling, it becomes possible to link specific strings in the references and to enhance them through semantic assertions according to specific vocabularies, as shown in the following excerpt:

```
:gruber95 a biro:BibliographicReference
  ; biro:references :principles-design-ontologies
  ; co:firstItem [ co:itemContent :author-name
  ; co:nextItem [ co:itemContent :publication-year
  ; co:nextItem [ co:itemContent
    :article-title ... ] ] ] .

:author-name a litre:Literal , foaf:name
  ; litre:isLiteralOf :gruber  # it is the URL
     identifying the person
  ; litre:hasLiteralValue "Gruber, T. R."^^xsd:string .

:publication-year a litre:Literal , fabio:
   hasPublicationYear
  ; litre:isLiteralOf [ frbr:realization :principles-
     design-ontologies ]
  ; litre:hasLiteralValue "1995"^^xsd:gYear .

:article-title a litre:Literal , dcterms:title
  ; litre:isLiteralOf [ frbr:realization :principles-
     design-ontologies ]
  ; litre:hasLiteralValue "Toward principles for the
     design of ontologies used for knowledge sharing"^^
     xsd:string .

...
```

As shown above, now the bibliographic reference under consideration is described as a list of literals, each of them having a particular semantic connotation.

5.3.1.3 EARMARK Ranges for Describing References

Another approach to deal with the semantic enhancement of bibliographic references would be to use LA-EARMARK ranges for associating appropriate semantic statements to textual fragments, as illustrated in Chap. 4. For instance, let us consider the article by Casanovas et al.'s "OPJK and DILIGENT: ontology modeling in a distributed environment" (Casanovas et al. 2007) implemented as an EARMARK document. In this case, I will have a particular docuverse containing the text of the reference used previously, for example:

```
:gruber95 a biro:BibliographicReference
  ; biro:references :principles-design-ontologies
  ; dcterms:bibliographicCitation "Gruber, T. R. (1995).
     Toward principles for the design of ontologies
     used for knowledge sharing, International Journal
     of Human Computer Studies, 43(5-6): 907-928." .
```

From this docuverse, I can define ranges for each string I want to use in order to describe the bibliographic reference according to BiRO. These ranges, that cover the same literal values used in the previous example can be defined as follows:

```
:gruber95 a biro:BibliographicReference
  ; biro:references :principles-design-ontologies
  ; co:firstItem [ co:itemContent :author-familyname
  ; co:nextItem [
    co:itemContent :author-firstlettersname
  ; co:nextItem [ co:itemContent :publication-year
  ; co:nextItem [ co:itemContent
    :article-title ... ] ] ] ] .

# the string "Gruber"
:first-author-surname a earmark:PointerRange
  ; earmark:refersTo :gruber95-reference
  ; earmark:begins "0"^^xsd:nonNegativeInteger
  ; earmark:ends "6"^^xsd:nonNegativeInteger .

# the string "T. R."
:author-firstlettersname a earmark:PointerRange
  ; earmark:refersTo :gruber95-reference
  ; earmark:begins "8"^^xsd:nonNegativeInteger
  ; earmark:ends "13"^^xsd:nonNegativeInteger .

# the string "1995"
:publication-year a earmark:PointerRange
  ; earmark:refersTo :gruber95-reference
  ; earmark:begins "15"^^xsd:nonNegativeInteger
  ; earmark:ends "19"^^xsd:nonNegativeInteger .

# the string "Toward principles for..."
:paper-title a earmark:PointerRange
  ; earmark:refersTo :gruber95-reference
  ; earmark:begins "22"^^xsd:nonNegativeInteger
  ; earmark:ends "95"^^xsd:nonNegativeInteger .

...
```

Furthermore, using the Linguistic Act ontology introduced in Chap. 4, it is possible to link EARMARK ranges to their formal meaning and to their particular references, i.e., literals. For instance, considering the range *:author-firstlettersname*, I can say that:

1. This range denotes a particular literal (e.g., “Thomas Robert”) that is owned by the first author.
2. This range express a particular meaning, e.g., the fact of having a given name.
3. This meaning is a conceptualisation of the literal hereby introduced.

Thus, according to LA-EARMARK, I will have:

```
# string "T. R."
:author-firstlettersname a la:Expression
  ; la:expresses foaf:givenName
  ; la:denotes :gruber-given-name .

:gruber-given-name a litre:Literal
  ; litre:hasLiteralValue "Thomas Robert"
  ; litre:isLiteralOf :gruber
  : la:hasConceptualization foaf:givenName .

[] a la:LinguisticAct
  ; sit:isSettingFor
    # myself, as author of this interpretation
      <http://www.essepuntato.it/me>
    # as the person having a certain name
    , :gruber
    # The letter identifying the name
    , :author-firstlettersname
    # The full version of the name
    , :gruber-given-name
    # The meaning associated to such a string
    , foaf:givenName .
```

5.3.2 C4O: How Much, Where and what Someone is Citing

Besides defining reference lists and bibliographic references in a machine-readable form, I will also focus on how these references are used in the citing paper. In particular, I would need entities that describe:

- in-text reference pointers within the citing paper;
- links to the bibliographic references denoted by in-text reference pointers;
- how much a particular document is locally cited by the citing document–i.e., the total amount of in-text reference pointers within the citing paper denoting the same bibliographic reference;
- how much an article is globally cited (according to particular bibliographic citation service);
- the contexts involved in a citation–i.e., the part P_{citing} of the citing article containing a particular in-text reference pointer and the part P_{cited} of the cited article that is relevant to P_{citing}.

The *Citation Counting and Context Characterization Ontology*[27] (*C4O*) has been developed to allow the description of the above entities. This ontology enables the characterisation of bibliographic citations in terms of their presence in an article by means of the following classes (shown in Fig. 5.6):

[27] C4O, the Citation Counting and Context Characterization Ontology: http://purl.org/spar/c4o. The prefix *c4o* refers to entities defined in it.

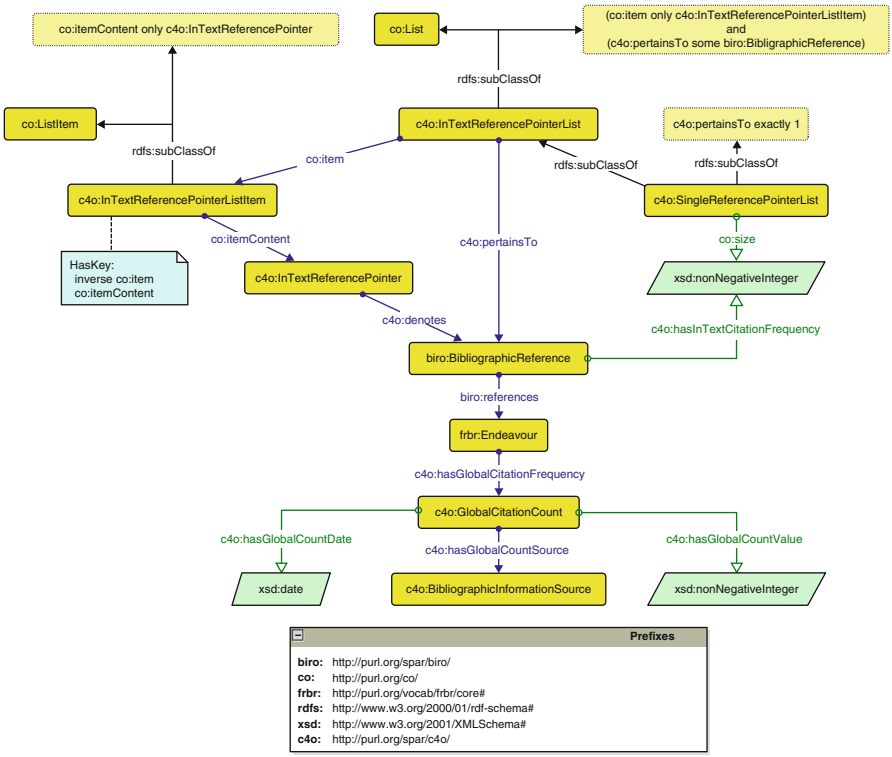

Fig. 5.6 Graffoo diagram summarising the C4O entities used for counting citations and references

- class *c4o:InTextReferencePointer*. An in-text reference pointer is a textual device denoting (property *c4o:denotes*) a single bibliographic reference that is embedded in the text of a document within the context of a particular sentence;
- class *c4o:InTextReferencePointerList*. A list containing (through the chain *co:item* and *co:itemContent*) only in-text reference pointers denoting the specific bibliographic references to which the list pertains (property *c4o:pertains*). Such a list cannot contain more than one item containing the same in-text reference pointer;
- class *c4o:SingleReferencePointerList*. Defined as subclass of the previous class, it is an in-text reference pointer list that pertains to exactly one bibliographic reference;
- class *c4o:GlobalCitationCount*. The number of times a work has been cited globally (property *c4o:hasGlobalCountValue*), as determined from a particular bibliographic information source (property *c4o:hasGlobalCountSource*) on a particular date (property *c4o:hasGlobalCountDate*).

C4O provides the ontological structures which allow one to record the number of in-text citations (property *c4o:hasInTextCitationFrequency*, i.e., the number of

in-text reference pointers to a single reference in the reference list of the citing article), and also the number of citations a cited entity has received globally (property *c4o:hasGlobalCitationFrequency*), as determined by a bibliographic information resource such as Google Scholar[28], Scopus[29] or Web of Knowledge[30] on a particular date.

Considering again the example in Sect. 5.3.1, I can write a set of assertions according to C4O that describe how many times a reference is used within the citing article and how much the cited article is globally cited (according to Google Scholar):

```
:gruber95 a biro:BibliographicReference
  ; biro:references :principles-design-ontologies
  ; c4o:hasInTextCitationFrequency
    "1"^^xsd:nonNegativeInteger .

:principles-design-ontologies
   c4o:hasGlobalCitationFrequency [
   a c4o:GlobalCitationCount
  ; c4o:hasGlobalCountDate "2013-06-09"^^xsd:date
  ; c4o:hasGlobalCountSource [
    a c4o:BibliographicInformationSource
      ; foaf:homepage <http://scholar.google.com> ]
  ; c4o:hasGlobalCountValue
    "6559"^^xsd:nonNegativeInteger ] .
```

Moreover, C4O enables ontological descriptions of the context where an in-text reference pointer appears in the citing document (modelled as shown in Fig. 5.7), and allows one to relate that context to relevant textual passages in the cited document.

Considering the previous bibliographic reference example, a possible C4O formalisation of the contexts involved by that citing act is:

[28] Google Scholar: http://scholar.google.it.

[29] Scopus: http://www.info.sciverse.com/scopus/.

[30] Web of Knowledge: http://apps.isiknowledge.com.

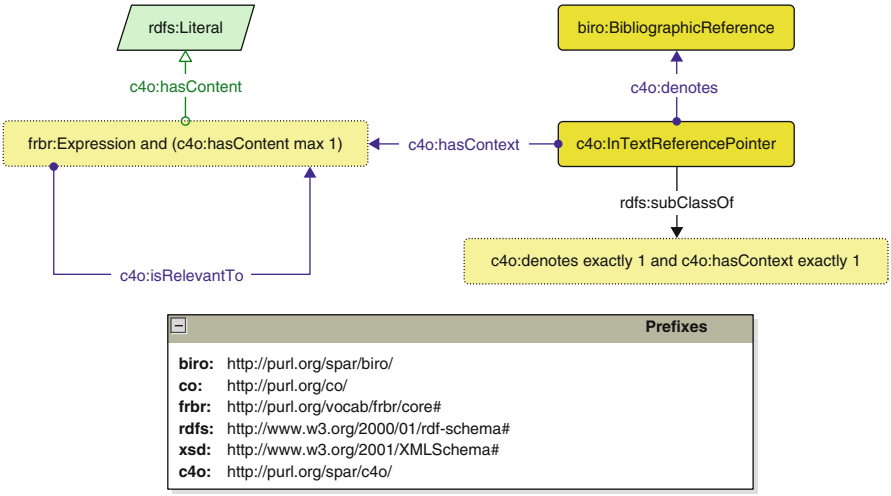

Fig. 5.7 Graffoo diagram summarising the C4O entities used for describing citation contexts

```
:version-of-record frbr:part :in-text-gruber95 .

:in-text-gruber95 a c4o:InTextReferencePointer
  ; c4o:denotes :gruber95
  ; c4o:hasContext :paragraph-in-version-of-record .

:paragraph-in-version-of-record a frbr:Expression
  ; c4o:hasContent "an ontology is defined as 'a shared
     specification of a conceptualization' (Gruber 1995)
     . Although 'shared' is an essential ..." .

:sentence-in-principles-design-ontologies a frbr:Expression

  ; frbr:partOf :principles-design-ontologies
  ; c4o:hasContent "An ontology is an explicit
     specification of a conceptualization."
  ; c4o:isRelevantTo :paragraph-in-version-of-record .
```

5.4 Characterising Document Parts with DoCO

A large amount of literature exists about models and theories for the description of structural, rhetorical and argumentative functions of texts through the adoption of Semantic Web technologies, as summarised in Schneider et al. (2011). The description of these different document layers is crucial for Semantic Publishing. As stated in De Waard (2010a), to improve substantially the users' comprehension of a document, a formalisation of the document *discourse* (e.g., the *scientific discourse* in scholarly articles) should be explicitly represented in machine-readable form within the document itself.

The issues relating to the rhetorical and the argumentative layers in documents have been debated a long time, e.g., Kircz (1991) and Groza et al. (2011), even in communities outside Computer Science, such as Philosophy and Publishing. For example, in his influential work (Toulmin 1959), the British philosopher Stephen Toulmin introduces a model for the explanation of arguments (including scientific arguments). In this model, each argument is composed of statements belonging to one of the following six roles:

- **Claim.** A fact that must be established–"This is a scientific article".
- **Evidence.** Another fact that represents a foundation for the claim–"The article has been submitted to a scientific conference".
- **Warrant.** A statement bridging from the evidence to the claim–"An article submitted to a scientific conference is a scientific article".
- **Backing (optional).** Credentials that certify the warrant–the Call for Papers of the particular conference where the article was submitted.
- **Rebuttal (optional).** Restrictions that may be applied to the claim–"Unless the conference reviewers reject it, judging it altogether non-scientific".
- **Qualifier (optional).** It asserts the degree of certainty in relation to the claim via words or phrases such as "certainly", "possible", "probably", etc.

Similarly, in the field of Publishing there exist specific constraints that authors have to follow when writing a paper. For example, some scientific journals, such as the Journal of Web Semantics[31], impose that their articles to follow a particular rhetorical segmentation, in order to identify explicitly what the meaningful parts are from a scientific point of view–e.g., *introduction*, *background*, *evaluation*, *materials*, *methods*, *conclusion*, etc. Although these parts usually (but do not necessarily) correspond to the structural parts of the article, i.e. the sections, they carry a specific semantics that characterises all the text they contain. From this perspective, this text means more than "being within a section".

During the development of the SPAR ontologies, these aspects were analysed in detail. Particularly, I studied carefully the previous works that had tried to address the description of structural and rhetorical components of a document. With regards to the rhetorical aspect, I found three models that dealt with document segmentation: the *Ontology of Rhetorical Blocks* (*ORB*) (Ciccarese and Groza 2011), the *SALT Rhetorical Ontology* (Groza et al. 2007a, 2007b) and the *Medium-Grained structure* (De Waard 2010b). The first and the second models offer a coarse-grained description (header, introduction, methods, claims, etc.) and, the third one, a medium-grained description (hypothesis, objects of study, direct representation of measurements, etc.) of the rhetorical components of a document.

Although all those models are currently in use, they do not deal satisfactorily with all the compositional aspects of a document. As well as not allowing the user to express all the rhetorical functions that SPAR needs, those models do not enable

[31] Journal of Web Semantics, Guide for Authors: http://www.elsevier.com/wps/find/journaldescription.cws_home/671322/authorinstructions.

rich descriptions of the document structure. One of the requirements of publishers is to have a model that enables the description of the several sub-parts of a document according to its structural components and their rhetorical characterisations. To this end, I developed the *Document Components Ontology*[32] (*DoCO*), which provides a structured vocabulary of document components. These components are structural (e.g., block, inline, container), rhetorical (e.g., introduction, discussion, acknowledgements, reference list, figure, appendix) and mixed (e.g., paragraph, section, chapter), and the ontology allows them, and the documents which are composed by them, to be described in RDF.

In particular, DoCO imports the *Patterns Ontology* presented in Sect. 3.3.2 and the *Discourse Elements Ontology*[33] (*DEO*) to describe, respectively, structural and rhetorical components of documents. Moreover, the latter ontology uses seven rhetorical block elements (*background*, *conclusion*, *contribution*, *discussion*, *evaluation*, *motivation* and *scenario*) abstracted from the *SALT Rhetorical Ontology*[34]. In the following subsections I will analyse in detail the structural and rhetorical functions which can be expressed through DoCO entities.

5.4.1 Building Blocks for Structuring Documents

A brief introduction to the theory of structural patterns for documents was illustrated in Sect. 3.3.2 and in the cited works (Dattolo et al. 2007; Di Iorio et al. 2005, 2012). In this section I will list the instanceable patterns again and give more precise definitions, using HTML examples[35].

The first patterns I will introduce are *milestone* and *meta*. They are defined as empty elements that can have zero or more attributes. In addition:

- The distinctive characteristic of the pattern *milestone* is the *position* that it assumes in the document. This pattern typically describes elements that change the aspect of a document, depending on where they are inserted. Moreover, this pattern is usually followed by elements that are used to define the actual content of a document. In HTML, the element *img* is a perfect example of this pattern.

[32] DoCO, the Document Components Ontology: http://purl.org/spar/doco. The prefix *doco* refers to entities defined in it.

[33] DEO, the Discourse Elements Ontology: http://purl.org/spar/deo. The prefix *deo* refers to entities defined in it.

[34] SRO, the SALT Rhetorical Ontology: http://salt.semanticauthoring.org/ontologies/sro.rdfs.

[35] Note that it is possible to create valid HTML documents that are not compliant with the presented structural pattern theory hereby presented. For that reason, in the examples that follow I will use HTML elements and consider their (informal) semantics as a strong requirement to make a *correct* document. Indeed, there are other markup formats that fit the structural pattern theory better than HTML, such as Akoma Ntoso (Barabucci et al. 2009). However, I have chosen to use HTML because it is a well-known and easily understandable markup format.

- The main feature of the pattern *meta* is its *existence*, independently from the position that it takes within the document. All the elements following this pattern are commonly used to define metadata which relate to the document itself or part of it, independently of where they are. In HTML, the elements *meta* and *link* are good examples that comply with this pattern.

The patterns *atom* and *field* define elements that can contain text only. In HTML, the element *title* (inside the element *head*) is an example of the pattern *field*, while there is no particular representative for the pattern *atom*.

The following HTML code summarises the patterns introduced so far:

```
<html>
  <head>
    <title>S.'s home</title>
    <link href="layout.css"
      rel="stylesheet" type="text/css" />
    <meta http-equiv="Content-Type"
      content="text/html; charset=UTF-8" />
  </head>
  <body>
    <img src="profile/thumbnail.jpg" />
  </body>
</html>
```

The next two patterns I am going to illustrate, i.e., *inline* and *block*, are followed by elements that are commonly used for the specification of the document content. They can both contain text and have the same content model that enable the definition of hierarchical structures: they can contain other *inline*, *atom* and *milestone* elements and items that comply with the pattern *popup*–which I will introduce further below. Elements which are compliant with the patterns *inline* and *block* differ for two aspects:

- Although inline elements can contain other inlines, block elements cannot contains other blocks.
- Inline elements cannot be used as root element of documents, but they must always be contained by block elements.

In HTML, there are many elements that comply with these two patterns. For example, *p* and *h1* follow the pattern block, while *em* and *a* comply with the pattern inline.

The pattern *container* concerns the structural organisation of a document. All the elements following this pattern do not contain any non-empty text. However, they can contain elements compliant with the following patterns: meta, atom, block and all the subtypes of container, but excluding popup.

While the content model of container elements (e.g., HTML *body*) is allowed to contain all optional and repeatable elements, particular restrictions are applied to the subtypes of the pattern container in terms of element repeatability. In particular:

- The pattern *table* contains homogeneous and repeatable elements. In HTML, elements that comply with this pattern are *ul* and *table*.
- The pattern *record* contains no repeatable elements (e.g., HTML element *html*).

- The pattern *headed container* contains always an header at the beginning that must be formed by block elements only. In HTML, the element *section* (when containing always an *h1*as first child) is a good example of this pattern.

Finally, the pattern *popup* is a particular structure that does not contain text and that can be contained by block and inline elements only. It is often used for the inclusion of complex quotations or other complex structures. In HTML, the element *math* is a primal example of this pattern.

The following HTML code summarises the set of patterns described above:

```
<html>
  <head><title>The formula</title></head>
  <body>
    <section>
      <h1>The <em>magic</em> mathematical formula</h1>
      <p>In this section I would like to introduce two
         things:</p>
      <ul>
        <li><p>the magic mathematical formula;</p></li>
        <li><p>the <a href="http://dev.w3.org/html5">
           website</a> that inspired me.</p></li>
      </ul>
      <p>And now the <em>magic</em> mathematical formula
         , that is
        <math>
          <mi>x</mi>
          <mo>=</mo>
          <mfrac>
            <mrow>
              <moform="prefix">-</mo>
                        <mi>3</mi>
                        <mo>*</mo>
                        <msqrt>
                          <msup>
                            <mi>y</mi>
                            <mn>2</mn>
                          </msup>
                        </msqrt>
                      </mrow>
                      <mrow>
                        <mn>2</mn>
                      </mrow>
                    </mfrac>
                  </math>.
                Isn'titterrific?</p>
              </section>
            </body>
          </html>
```

The ontology introduced in Sect. 3.3.2 implements the whole theory I presented so far. As highlighted in that section, a document compliant with this theory results less ambiguous, more manageable and better-structured according to defined and shared principles of document engineering.

I have chosen to use this theory as one of the building blocks of DoCO (by importing the related ontology) for two reasons. On the one hand, it allows one to understand whether a document described in terms of DoCO entities is valid against the pattern theory, by simply checking whether the ABox which describes that document is consistent or not. On the other hand, by following specific and precise instructions, one can model a new document according to the pattern theory from scratch.

5.4.2 *Mixing Rhetorical Characterisation and Structural Components*

Documents such as a scientific research articles are characterised by precise rhetorical organisations, sometimes in a way that is in some measure independent from their structural components. As stated beforehand, there exist models—e.g., Ciccarese and Groza (2011), Groza et al. (2011), Groza et al. (2007a), Groza et al. (2007b), De Waard (2010b)–that try to describe rhetorical characterisations of documents from different perspectives. Although these ontologies can be used for the description of rhetorical aspects of documents, some of them do not link explicitly and correctly pure structural behaviours to rhetorical aspects. Probably, one of the principal causes of this deficiency should be attributed to the intrinsic complexity of defining some components as purely rhetorical or purely structural.

In order to clarify this point, let me consider as example a well-known document component: the *paragraph*. The structural behaviour of a document component can be described by the syntactic structures that it enables and not by relating it to its rhetoric nature. From this perspective, a paragraph cannot be considered a pure structural component–i.e., a component which carries only a syntactic function–since it carries *de facto* a meaning through its natural language sentences. Thus, paragraphs have more than a syntactic character. At the same time, the aforementioned models for the rhetorical characterisation of documents do not include the concept *paragraph* as part of them. Are thus paragraphs neither structural components nor rhetorical elements?

Of course, the truth must lie somewhere in between. Let me recall the two definitions that take an important part in this discussion. The definition of *rhetoric* as "the art of discourse, an art that aims to improve the facility of speakers or writers who attempt to inform, persuade, or motivate particular audiences in specific situations"[36], and the definition of *paragraph* as "a self-contained unit of a discourse in writing

[36] Wikipedia article "Rhetoric": http://en.wikipedia.org/wiki/Rhetoric.

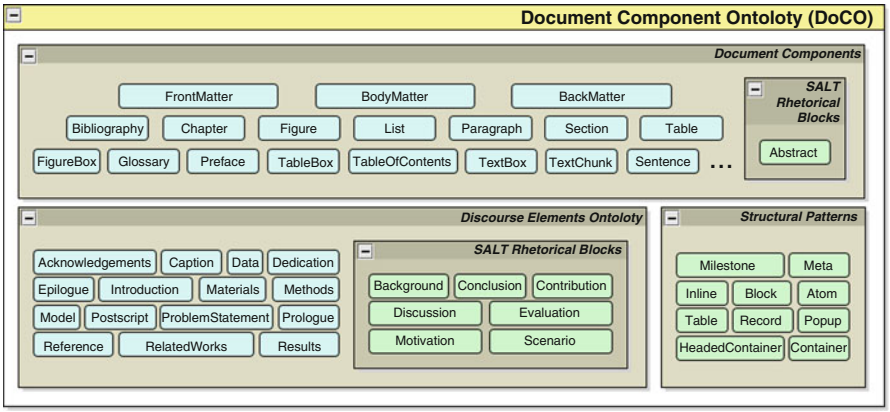

Fig. 5.8 Diagram describing the composition and the classes of the Document Components Ontology (*DoCO*)

dealing with a particular point or idea"[37]. From these definitions I can deduce that: the fact that a paragraph is a unit of *discourse* implies that it must have a rhetorical connotation, since the rhetoric is the art of *discourse*. Therefore, a textual fragment of a document is a paragraph when it is more than a mere syntactic structure: it should express some ideas and should carry some meanings.

On the other hand, document markup languages such as HTML and DocBook define a paragraph as a pure structural component, without any reference to its rhetoric function:

- "A paragraph is typically a run of phrasing content that forms a block of text with one or more sentences" (Berjon et al. 2013).
- "Paragraphs in DocBook may contain almost all inlines and most block elements" (Walsh 2010).[38]

Here the term "block of text" and the verb "contains" emphasise the structural connotation of the paragraph, which is amplified by our direct experience as readers. Our experience implicitly tells us that a particular textual fragment shown in a book or in an HTML page is a paragraph rather than a chapter or a table.

The *Document Components Ontology* (*DoCO*), shown in Fig. 5.8, has been developed to fill the gap between the pure structural characterisation of document elements and their the pure rhetorical connotation. Besides including the Pattern Ontology (describing structural components) and the Discourse Element Ontology (describing rhetorical components), DoCO defines also some other hybrid classes which describe elements that are structural and rhetorical at the same time. For instance:

[37] Wikipedia article "Paragraph": http://en.wikipedia.org/wiki/Paragraph.

[38] The words *inline* and *block* in these list items do not refer to the structural pattern theory introduced earlier, although some sort of overlapping exist.

- class *doco:Paragraph*. It is a discourse element based on the pattern block, and contains some sentences;
- class *doco:Sentence*. It is a discourse element based on the pattern inline;
- class *doco:Chapter*. It is a discourse element based on the pattern container, and it is part of the body-matter of a document;
- class *doco:BodyMatter*. It is a discourse element based on the pattern container, etc.

The following excerpt shows how to use DoCO to describe structural and rhetorical aspects of the text of the example in Sect. 5.3:

```
:version-of-record a pattern:Container
  ; pattern:contains
    :front-matter , :body-matter , :back-matter .

...

:body-matter a doco:BodyMatter
  ; pattern:contains :introduction , :related-works ,
     ...

:related-works a doco:Section , deo:RelatedWork
  ; pattern:contains :first-paragraph , ...
    , :paragraph-in-version-of-record , ...

:paragraph-in-version-of-record a doco:Paragraph
  ; pattern:contains :sentence1 , :sentence2, ...

:sentence1 a doco:Sentence
  ; pattern:contains :in-text-renear02 .

:in-text-renear02 a deo:Reference , pattern:Inline .

...

:back-matter a doco:BackMatter
  ; pattern:contains [ a doco:Section
    ; pattern:contains :reference-list ] .

:reference-list a doco:ListOfReferences
  ; pattern:contains
    :barwise83 ,:black37 ... , :renear02 , ...

:renear02 a
  deo:BibliographicReference , pattern:Inline .

...
```

Moreover, as shown for BiRO (Sect. 5.3.1.3) and as illustrated in Sect. 4.3, DoCO can be used in combination with LA-EARMARK to enhance the document markup with axioms related to its structural and rhetorical aspects.

5.5 In the Past you were it, Now you are not it

When modelling a domain, we often need to describe scenarios in which an *entity* has some *value* only within a specific temporal interval and/or contextual (e.g., social, cultural, physical) environment, and a different one (or none at all) otherwise. For instance, in the publishing domain, we may want to describe the *status* of a document at any given moment (e.g., *draft*, *under review*, *accepted*, *published*), the institution of which the author is a member, or the *role* held by different people in the publishing process. All these scenarios involve an *entity*, a *value* and, in particular, a *time* and a *context* within which the entity is associated to the value.

Most ontologies are unable to model such scenarios effectively, for different reasons. Three techniques in particular have been used in attempting to address this modelling issue–*class subsumptions*, *property links* and *inter-linked classes*, but each fell short in some aspect[39].

5.5.1 Using Class Subsumptions

To clarify this design technique and the issues that arise from it, let me consider the agent/role relations as described in the Portal Ontology[40] of the AKT Reference Ontology. This ontology defines the class *Student* as a person (class *portal:Person*) who studies at (property *portal:studies-at*) some institution (class *portal:EducationalOrganization*), as follows (in Turtle (Prud'hommeaux and Carothers 2013)):

```
portal:Student a owl:Class
  ; rdfs:subClassOf [ owl:intersectionOf (
    portal:Person
    [ a owl:Restriction
      ; owl:onProperty portal:studies-at
      ; owl:someValuesFrom
        portal:EducationalOrganization ] ) ] .
```

The fact of being a person is *time-independent*–Silvio Peroni is a living person while Kurt Vonnegut is a dead person, but we still describe both as persons. On the contrary, the fact of being a student is strictly *time-dependent*–Silvio Peroni was recently a graduate student, but he is one no longer.

[39] Contrarily to the paper used as examplar in the previous sections, i.e., Casanovas et al. (2007), here I have decided to use other two papers, i.e., Peroni et al. (2008) , which I co-authored. This is totally necessary because in Sects. 5.5 and 5.6 I will describe in detail how bibliographic entities and related resources are perceived over time. Thus, it is preferable to use a context I am well versed in rather than inventing possible (or even fake) scenarios related to Casanovas et al. (2007).

[40] Portal Ontology: http://www.aktors.org/ontology/portal. The prefix *portal* refers to entities defined in it.

Thus the subsumption model shows a clear design problem: a class having time-dependent characteristics, namely *portal:Student*, has been placed in the same *is-a* hierarchy (i.e., defined through a *rdfs:subClassOf* relation) as a class having time-independent characteristics, namely *portal:Person*. As suggested in Guarino and Welty (2002), which calls them respectively *anti-rigid* and *rigid* classes, I believe they should be part of two separated hierarchies, and I conclude that descriptions of time-dependant entities cannot be satisfactorily achieved using plain subsumption.

5.5.2 Using Property Links

A solution that takes into proper account anti-rigid characteristics is the use of a specific property for defining each time-dependent *value* that an entity has (e.g., the role of a person), while continuing to express the *entity* itself as an individual of a particular class (e.g., a document). Many ontologies for describing bibliographic resources, such as DCTerms (Dublin Core Metadata Initiative 2012) and BIBO (D'Arcus and Giasson 2009), use object properties to model this, linking the document to the persons who are its authors by the use of a specific property (namely, *dcterms:creator*and *bibo:authorList*, respectively), as shown here:

```
# Using DCTerms
:paper1 a dcterms:BibliographicResource
  ; dcterms:creator :peroni , :vitali .
:paper2 a dcterms:BibliographicResource
  ; dcterms:creator
    :peroni , :motta , :daquin .
# Using BIBO
:paper1 a bibo:Article
  ; bibo:authorList ( :peroni , :vitali ) .
:paper2 a bibo:Article
  ; bibo:authorList
    ( :peroni , :motta , :daquin ) .
```

This approach presents at least two problems. The first is that we need as many properties as there are roles. Thus if the requirements are not fully known in advance or change over time, the TBox of the ontology will require extensions to include new properties on a case-by-case basis, and will require continuous maintenance, with the increasing risk of inconsistencies. For instance, the number of roles in the publishing domain (author, editor, publisher, etc.) has been increasing in recent times (see, for example, the list of MARC relators[41] dated 7th December 2010), and the current technological and cultural evolution will surely lead to the creation of new ones (for instance, very soon now, that of *linked-data manager*).

Alternatively, one could use data properties rather than object properties. For instance, the W3C specification of the ontology for describing vCard objects in

[41] MARC Code List for Relators: http://www.loc.gov/marc/relators/relaterm.html.

RDF (Iannella 2013) prescribes the use of a (very general) data property, *vcard:role*, for an individual's roles. But while this allows easy extensions to the ontology by adding arbitrary literals to represent new roles, it also lacks a clear and well-defined vocabulary for existing ones, causing potential ambiguities (e.g., with the literals "Graduate student" and "Ph.D. student" being used to refer to the same role but being formally different within the model).

A second problem, that affects the scenario regardless of whether we use object properties or data properties, is the difficulty of discerning the context in which an *entity-value* association holds. For instance, consider an author having different institutional affiliations in the context of different publications (e.g., because he/she moved from one to the other). Using the *Semantic Web Conference Ontology*[42] (Moller et al. 2009), we can define affiliations for *:peroni* as follows:

```
:peroni swrc:affiliation :cs-unibo . # Paper 1
:cs-unibo a foaf:Organization
  ; dcterms:description
    "CS Dept., University of Bologna" .
:peroni swrc:affiliation :kmi . # Paper 2
:kmi a foaf:Organization
  ; dcterms:description
    "KMi, Open University" .
```

This specification, although straightforward, does not differentiate between associations. Namely, in OWL author *:peroni*, is associated indifferently to both *:cs-unibo* and *:kmi*, and it is not possible to determine the affiliation of an author within the context of a particular paper–e.g., "give me the institutional affiliation of the person *:peroni* as author of paper 2" (although in Masolo et al. (2004), Masolo et al. propose an approach to deal with this issue by using *qua-individuals*). Finally, the approach proposed in CIDOC CRM allows one to use the meta-property *P14.1 in the role of* Crofts et al. (2011) (a sub-property of property *P14 carried out by*) in order to specify the role that an agent has in the context of a particular event (such as being affiliated to an institution) through an instance of the class *E55 Type*. However, the official RDFS ontology of CIDOC CRM[43] does not implement any meta-property, and in practice RDF lacks the expressive power needed to define meta-properties.

5.5.3 Using Inter-Linked Classes

A different way to address the time-dependent association of entities to values is to consider both as classes (with no declared or inferable subsumption), and to link

[42] The Semantic Web Conference Ontology: http://data.semanticweb.org/ns/swc/ontology. The prefixes *swc* and *swrc* refer to entities defined in it.

[43] RDFS ontology of CIDOC CRM: http://www.cidoc-crm.org/rdfs/cidoc-crm-english-label.

them through object properties. For instance, the *Semantic Web Conference Ontology* (Moller et al. 2009) implements this method through two classes, *foaf:Person* and *swc:Role*, and the property *swc:holdsRole* linking them.

The extensibility of the ontology is thus guaranteed, reducing the possibility of undesirable inferential side effects. In fact, adding new roles simply involves adding new individuals to the class *swc:Role,* requiring no modification to the TBox. However, as before, this method is still unable to describe the context or time frame in which the person holds the particular role. For instance, Silvio Peroni was an undergraduate student at the University of Bologna between 2005 and 2008, a graduate student at the same university between 2009 and 2012, and also an intern at the University of Oxford in 2010

The SWC ontology, used together with FOAF (Brickley and Miller 2010), gives only a partial description of this, as shown by the following excerpt:

```
:undergraduateStudent a swc:Role .
:graduateStudent a swc:Role .
:intern a swc:Role .
:peroni a foaf:Person
  ; swc:holdsRole :undergraduateStudent
    , :graduateStudent , :intern .
```

This description cannot answer the question "Was *:peroni* a graduate student in 2008?", because it lacks information about time. By adding a specific model, such as the Time ontology[44] (Hobbs and Pan 2006), we manage to capture this information as follows:

```
:atTime a owl:ObjectPropery
  ; rdfs:domain swc:Role
  ; rdfs:range time:TemporalEntity .
:undergraduateStudent :atTime [
a time:TemporalEntity
; time:hasBeginning [ a time:Instant
  ; time:inDateTime [
  ; time:year "2005" ] ]
; time:hasEnd [ a time:Instant
  ; time:inDateTime [
  ; time:year "2008" ] ] ] . ...
```

The problem here is that the time-related information is associated to the roles, rather than to the person holding them. Being an undergraduate student is associated to the 2005–2008 time interval. This of course creates problems once we add another person with the same role, since it will become impossible to reuse the same role unless he/she also happens to have been an undergraduate in the same period. Things becomes even more complicated if we need also to describe the social or cultural *context* within which the agent-role relation holds, for example, by specifying in

[44] The Time Ontology: http://www.w3.org/2006/time. The prefix *time* refers to entities defined in it.

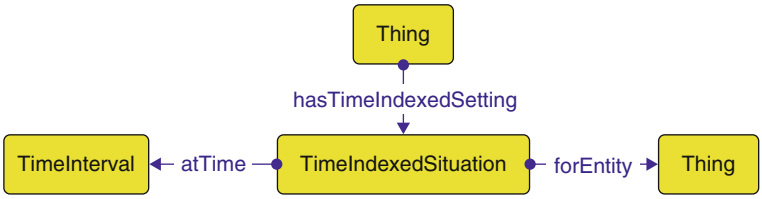

Fig. 5.9 A graphical representation of the *time-indexed situation* ontological pattern

which institution *:peroni* was an intern on a given date. It would require one to multiply the instances of role by the number of contexts and time intervals for which the different roles are relevant.

5.5.4 Using N-ary Class Modelling

Some ontological patterns that partially address these issues have been developed (Presutti and Gangemi 2008; Hayes and Welty 2006; Aranguren et al. 2008). For example, through the *time-indexed situation* pattern[45], shown in Fig. 5.9[46], it becomes possible to link a subject to a time-dependant description of a situation[47].

Using this pattern, the scenario presented in Sect. 5.5.3 can be defined as follows:

```
# University of Bologna
:unibo a foaf:Organization .
# University of Oxford
:oxac a foaf:Organization .
```

[45] Time-indexed situation pattern: http://ontologydesignpatterns.org/cp/owl/timeindexedsituation.owl. The prefixes *tisit*, *sit* and *ti* refer to entities defined in it.

[46] This and all the following graphical representations of ontologies are drawn using Graffoo, the Graphical Framework for OWL Ontologies, available at http://www.essepuntato.it/graffoo. *Yellow rectangles* represent classes (*solid border*) and restrictions (*dotted border*), *green parallelograms* represent datatypes, *arrows* starting out of a *filled circle* refer to object property definitions, *arrows* starting out of an *open circle* refer to data property definitions, while other arrows represent assertions between resources.

[47] In this context, a *situation* is defined as a view on a set of entities. It can be seen as a "relational context", reifying a relation.

```
:peroni tisit:hasTimeIndexedSetting
  :peroniAsGraduateStudentInUnibo
  , :peroniAsIndernInOxAc .
:peroniAsGraduateStudentInUnibo
  a tisit:TimeIndexedSituation
  ; tisit:atTime [ a ti:TimeInterval
    ; ti:hasIntervalStartDate
      "2009"^^xsd:gYear ]
  ; tisit:forEntity :unibo , :graduateStudent .
:peroniAsInternInOxAc a
  tisit:TimeIndexedSituation
  ; tisit:atTime [ a ti:TimeInterval
    ; ti:hasIntervalStartDate
      "2010-06"^^xsd:gYearMonth
    ; ti:hasIntervalEndDate
      "2010-12"^^xsd:gYearMonth ]
  ; tisit:forEntity :oxac , :intern .
```

Although this pattern correctly describes our scenario, it is still too abstract, both as a model and in terms of its terminology. In particular, the *tisit:forEntity* object property provides little or no guidance as to the interpretation of the relation with the entity. In fact, for instance in *:peroniAsInternInOxAc*, the way the various entities were involved in the situation is not clear, and we do not know what are the relations linking *:peroni*, *:oxac* and *:intern*. Given that, we could come to different conclusions just permuting the entities involved, e.g., a) the person *:peroni* was associated to the institution *:oxac* in a particular period, during which the latter had the role of *:intern* (definitely incorrect), or b) that the person *:peroni* worked for a particular period with somebody having the role of *:intern* within the institution *:oxac* (still incorrect), or c) that the person *:peroni* has the role *:intern* within the institution *:oxac* during a particular period (as intended, finally). Thus, using this pattern we would need to apply additional (unwarranted and "smart") steps to infer the correct interpretation of the situation.

To resolve this ambiguity, I decided to extend this pattern so as to specify the relations held by individuals in a situation involving a time-indexed value in a context.

5.5.5 A General Pattern for Roles and Statuses

What emerges from the preceding argument is the need for a model to describe time-dependant and contextualised entities. In particular, I identified four different items involved in these types of scenarios:

1. the *entity* having some value, e.g., a person or a document possessing a role or a status;
2. the *value* had by someone, e.g., a role or a status;
3. the *time period* during which the entity *has* that value, e.g., from April 2008 to September 2008;
4. the particular *context* that characterises the act of *having that value*, e.g., being a member of an institution or the editor of a particular journal.

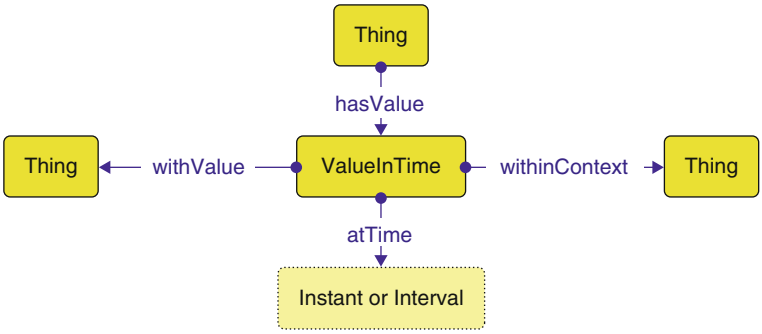

Fig. 5.10 The Graffoo diagram of the *time-indexed value in context* ontological pattern

In Sect. 5.5.4 I introduced a pattern that is able to describe this scenario at an abstract level but that it lacks a mechanism to describe the reciprocal relations of the entities involved. Using that as a starting point, we now wish to define a new ontological pattern called *time-indexed value in context* (*TVC*), summarised in Fig. 5.10 and available as an OWL ontology[48].

This pattern is composed by of different classes and four object properties:

- The class *ValueInTime* is a particular kind of time-indexed situation (i.e., a subclass of *tisit:TimeIndexedSituation*) as shown in Sect. 5.5.4 that represents a hub linking the *entity* having a particular value, the *value* itself and the *temporal and contextual extents* on which the entity-value relationship depends.
- The object property *hasValue* (sub-property of *tisit:hasTimeIndexedSetting*) links an entity (e.g., a Person) to a particular *ValueInTime* situation.
- The object property *withValue* (sub-property of *tisit:forEntity*) gives the value held by the entity taking part inthe situation.
- The classes *Instant* and *Interval* are used, respectively, to specify the particular temporal instant or time period in which the situation takes place. This is done through the object property *atTime*, which is not defined as a sub-property of *tisit:atTime* since it can be used to describe instants as well as intervals.
- The object property *withinContext* (sub-property of *tisit:forEntity*) links to the specific social, cultural or physical context within which the fact of the entityhaving the valueis relevant.

Using the TVC ontology, the *:peroniAsInternInOxAc*excerpt introduced in the previous section can be re-written as follows:

[48] *Time-indexed value in context* pattern: http://www.essepuntato.it/2012/04/tvc. The prefix *tvc* refers to entities defined in it.

```
:peroni tvc:hasValue :peroniAsInternInOxAC .
:peroniAsInternInOxAc
  a tvc:ValueInTime
  ; tvc:atTime [ a ti:TimeInterval
    ; ti:hasIntervalStartDate
      "2010-06"^^xsd:gYearMonth
    ; ti:hasIntervalEndDate
      "2010-12"^^xsd:gYearMonth ]
  ; tvc:withinContext :oxac
  ; tvc:withValue :intern .
```

where *:intern* is the value (i.e., the role) held by *:peroni* during that particular time period, and *:oxac* is the context. In the following sections I will expand these concepts, introducing use cases and explaining the benefits of using TVC.

5.5.5.1 Querying a TVC-Based Model via SPARQL

In principle, the TVC pattern allows a large number of SPARQL 1.1 queries (Garlik and Seaborne 2013) to return intuitively correct answers. In this section, I will discuss as examples three queries of increasing difficulty.

For instance, we can ask for all the values assigned to a person (e.g., the roles held by *:peroni*):

```
SELECT DISTINCT ?value WHERE {
  :peroni a foaf:Person
    ; tvc:hasValue/tvc:withValue ?value }
```

This query can be refined to consider, for instance, only those values that are defined in a particular context, e.g., the University of Bologna (entity *:unibo*):

```
SELECT DISTINCT ?value WHERE {
  :peroni a foaf:Person
    ; tvc:hasValue [ a tvc:ValueInTime
      ; tvc:withValue ?value
      ; tvc:withinContext :unibo ] }
```

This will return both the undergraduate and the graduate student roles of *:peroni*. We can further filter the previous results to return just those roles that are applicable at a particular date such as 24 August 2010:

```
SELECT DISTINCT ?value WHERE {
  :peroni a foaf:Person
    ; tvc:hasValue [ a tvc:ValueInTime
      ; tvc:withValue ?value
      ; tvc:withinContext :unibo
      ; tvc:atTime [ a ti:TimeInterval
        ; ti:hasIntervalStartDate ?start
        ; ti:hasIntervalEndDate ?end ] ]
  FILTER(
    xsd:dateTime(?start)
      <= "2010-08-24T00:00:00Z" &&
    xsd:dateTime(?end) > "2010-08-25T00:00:00Z"
  ) }
```

This will return just the role of graduate student. If the condition *tvc:withinContext:unibo* was omitted, the query would return both *:peroni*'s role as a graduate student at the University of Bologna, and his concurrent role on that date as an intern at the University of Oxford. More complicated and domain-specific queries are introduced in Sect. 5.5.6.

5.5.5.2 Reusing External Classes as Values

It is possible, by means of the meta-modelling features of OWL 2 (i.e., OWL punning), to define classes of external ontologies as objects of *tvc:withValue* assertions. In this way, we can use them interchangeably either as instances, when we want to associate them directly with some entity, or as classes when we want to understand hierarchical relationships between them. In addition to opening up the TVC ontology for reuse, this method may be very useful for inferring new data for specific categories, even when, in a query, we use their more abstract generalisations (i.e., superclasses).

Consider for example the following dataset defined according to TVC and including entities from the Portal Ontology:

```
:person1 tvc:hasValue [ a tvc:ValueInTime
  ; tvc:withValue portal:Affiliated-Person ] .
:person2 tvc:hasValue [ a tvc:ValueInTime
  ; tvc:withValue portal:Student ] .
:person3 tvc:hasValue [ a tvc:ValueInTime
  ; tvc:withValue portal: PhD-Student ] .
# Statements defined in the Portal Ontology
portal:Student rdfs:subClassOf
  portal:Affiliated-Person .
portal:PhD-Student rdfs:subClassOf
  portal:Student .
```

In this way, it is possible to query the dataset through SPARQL, asking for all the people affiliated with the University of Bologna (the entity *:unibo*), independently

from the roles they may hold as a student, a Ph.D. student, or other subclass of *portal:Affiliated-Person*:

```
SELECT DISTINCT ?person WHERE {
  ?person tvc:hasValue [ a tvc:ValueInTime
    ; tvc:withValue ?aff
    ; tvc:withinContext :unibo ] .
  { SELECT ?aff WHERE {
    { ?aff a owl:Class .
      FILTER(?aff = portal:Affiliated-Person) }
            UNION
            { ?aff rdfs:subClassOf+
              portal:Affiliated-Person } } } }
```

TVC makes it possible and useful to reuse specific parts of other ontologies which describe categories in the form of classes, thus taking advantages of the OWL 2 punning.

5.5.5.3 Constructing Second-Order Inferences

Of course, it is sometimes desirable to reuse ontologies that specify categories (e.g., roles) through properties rather than classes, as introduced in Sect. 5.5.2. Consider, for example, BIBO (D'Arcus and Giasson 2009), that associates agent roles with documents through particular sub-properties (e.g., *bibo:translator*, *bibo:director*, *bibo:editor*) of the general property *dcterms:contributor*. Using the BIBO ontology with TVC, these object properties can be used as objects of *tvc:withValue* assertions, by means of OWL 2 punning. Moreover, it is possible to construct second-order inferences using the objects of *tvc:withValue* assertions as properties:

```
CONSTRUCT { ?doc ?property ?person } WHERE {
  ?person a foaf:Person
    ; tvc:hasValue [ a tvc:ValueInTime
      ; tvc:withValue ?property
      ; tvc:withinContext ?doc ]
  { SELECT ?property WHERE {
    { ?property a owl:ObjectProperty .
     FILTER(?aff = dcterms:contributor) } UNION
    { ?property rdfs:subPropertyOf+
      dcterms:contributor } } } }
```

Through a model that combines TVC and an ontology which defines categories as property links, such as BIBO, it becomes feasible to infer second-order logical statements. More generally, TVC can be used as an intermediate model for the conversion of entity-value relationships from one ontology into another, independent of the particular design technique used by each ontology (e.g., class subsumptions, property links, inter-linked classes or n-ary relationships).

5.5.6 *Identifying a Person's Roles with PRO*

The ability to define publishing roles in SPAR was crucial for the completeness of this suite of ontologies. The problems associated with the adoption of external ontologies to handle this particular requirement has been discussed above. None of those ontologies were able to satisfy the modelling requirements imposed by SPAR in full, in particular the need for an ease of *extendibility* and for the simultaneous representation of *time periods* and *contexts*.

Using TVC as the basis, I implemented *PRO*, the *Publishing Roles Ontology*[49]. This ontology, shown in Fig. 5.11, permits the characterisation of the roles of agents–people, corporate bodies and computational agents–in the publication process. Furthermore, it allows one to specify the role an agent has in relation to a particular bibliographic entity (as author, editor, reviewer, etc.) or to a specific institution (as publisher, librarian, etc.), and the period during which each role is held.

Using PRO and its TVC-compliant structure as illustrated in Table 5.2 on page 173, it is possible to describe all the scenarios discussed in Sects. 5.5.2 and 5.5.3, as follows:

[49] PRO, the Publishing Roles Ontology: http://purl.org/spar/pro. The prefix *pro* refers to entities defined in it.

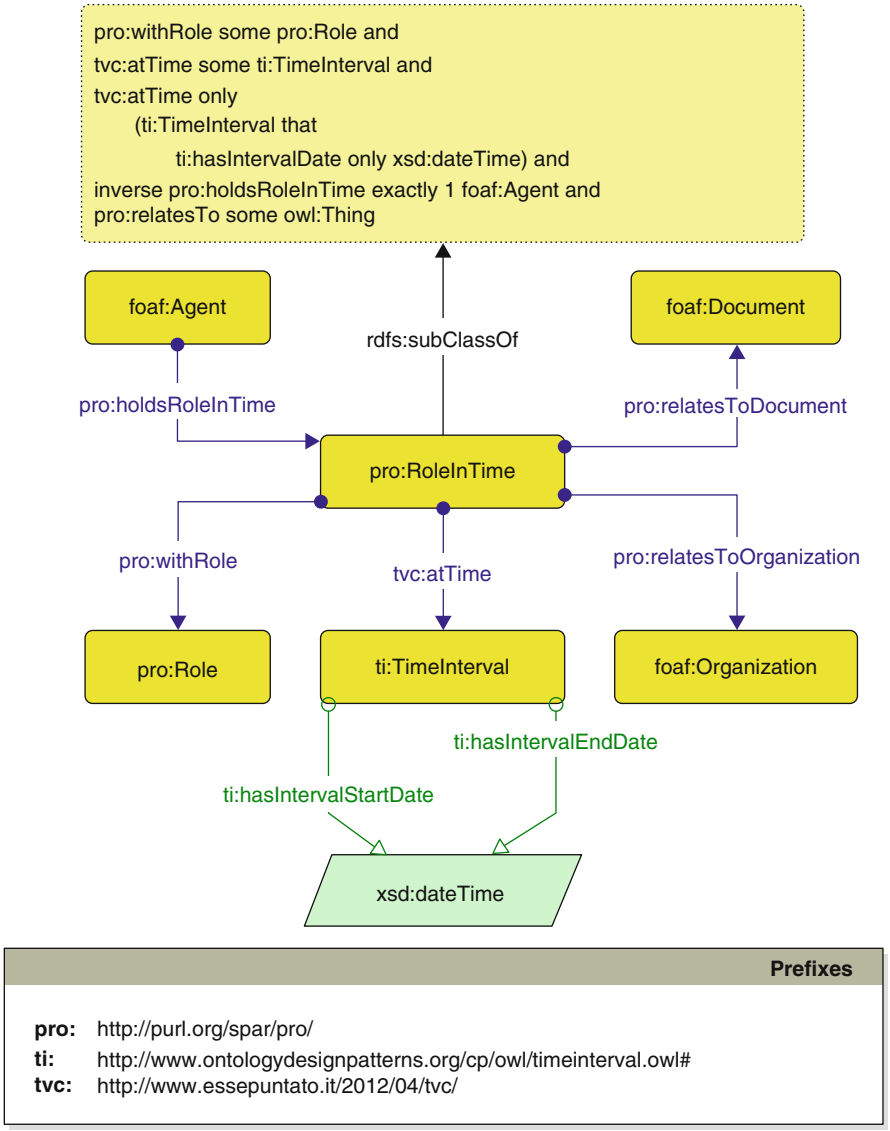

Fig. 5.11 Graffoo representation of the Publishing Roles Ontology (*PRO*)

Table 5.2 Alignments between TVC and PRO

TVC entity	PRO entity	Description
ValueInTime	RoleInTime	The class of the particular situation that describes the role an agent has within a particular time interval
hasValue	holdsRoleInTime	The object property linking any *foaf:Agent* (e.g., a person, a group, an organisation or a software agent), to a *pro:RoleInTime* situation
withValue	withRole	The object property linking the situation to the role the agent has. Currently, 31 roles are defined in the PRO ontology as individuals of the class *pro:Role*
Instant or Interval	TimeInterval	Two (starting and ending) points in time that define a particular period related to (object property *tvc:atTime*) a *pro:RoleInTime* situation
withinContext	relatesToDocument and relatesToOrganization	Object properties linking any kind of bibliographic work (*foaf:Document*) or publishing organisation (*foaf:Organization*) taking part in a *pro:RoleInTime* as contextual extent

```
:peroni pro:holdsRoleInTime
  # as author of two different papers
  [ a pro:RoleInTime
    ; pro:withRole pro:author
    ; pro:relatesToDocument
      :earmark-paper , :kce-paper ]
  # as affiliate of UniBo CS Dept
  , [ a pro:RoleInTime
    ; pro:withRole pro:affiliate
    ; pro:relatesToDocument :earmark-paper
    ; pro:relatesToOrganization :cs-unibo ]
  # as affiliate of OU KMi
  , [ a pro:RoleInTime
    ; pro:withRole pro:affiliate
    ; pro:relatesToDocument :kce-paper
    ; pro:relatesToOrganization :kmi ] .
```

As we can see, through PRO one can model very rich scenarios, and thus answer complex queries, such as the previously introduced "give me the institutional affiliation of the person *:peroni* as author of the paper *:earmark-paper*":

```
SELECT ?aff WHERE { :peroni pro:holdsRoleInTime
  [ a pro:RoleInTime ; pro:withRole pro:author
    ; pro:relatesToDocument :earmark-paper ]
    , [ a pro:RoleInTime
      ; pro:withRole pro:affiliate
      ; pro:relatesToDocument :earmark-paper
      ; pro:relatesToOrganization ?aff ] }
```

5.5.7 Specifying Document Statuses with PSO

The *status* of a document is the second subdomain of publishing handled in SPAR and based on the TVC pattern. In this case, the entity is a document holding a particular status at a certain time as a direct consequence of a particular event. For instance, a document is under review until all reviewers send in their comments and the editor decides whether to accept or reject the paper. After the acceptance/rejection decision is made, the status "under review" is no longer valid: this should be formally describable using an appropriate ontology. Moreover, it is sometimes useful to link documents to the decisions or events that cause the acquisition or loss of a particular status.

Preexisting ontologies describing the status of documents (e.g., BIBO (D'Arcus and Giasson 2009), the *Project Documents Ontology*[50] (Varma 2010) and the *Document Status Ontology*[51]), rely for this on specific property links. As discussed in Sect. 5.5.2, this approach prevents a proper description of scenarios that require a temporal duration for each status. With the exception of the Document Status Ontology, which describes status changes as events, the other ontologies do not allow time-dependent data, or can do so only partially.

In order to address these issues in a more satisfactory manner, I developed *PSO*, the *Publishing Status Ontology*[52]. This ontology (shown in Fig. 5.12) characterises the publication status of a document or any other publication entity at each of the different stages in the publishing process (e.g., draft, submitted, under review, rejected for publication, accepted for publication, version of record, peer reviewed, open access, etc.). As with PRO, PSO was developed following the TVC pattern, as shown in Table 5.3 on page 176. Using PSO, it is possible to describe the statuses of a document and how they change over time. For instance, consider the following description:

> The paper *:earmark-paper* was submitted to DocEng 2009 on 24 April 2009 at 13:18. At noon on 26 April, when the authors received acknowledgement of safe receipt of the paper from the conference editorial committee, the paper was considered "under review" until 27 May at 17:38.

PSO can be used to represent this description, as follows:

[50] Project Documents Ontology:http://ontologies.smile.deri.ie/pdo#.

[51] Document Status Ontology: http://ontologi.es/status#.

[52] PSO, the Publishing Status Ontology: http://purl.org/spar/pso. The prefix *pso* and *part* refer to entities defined in it.

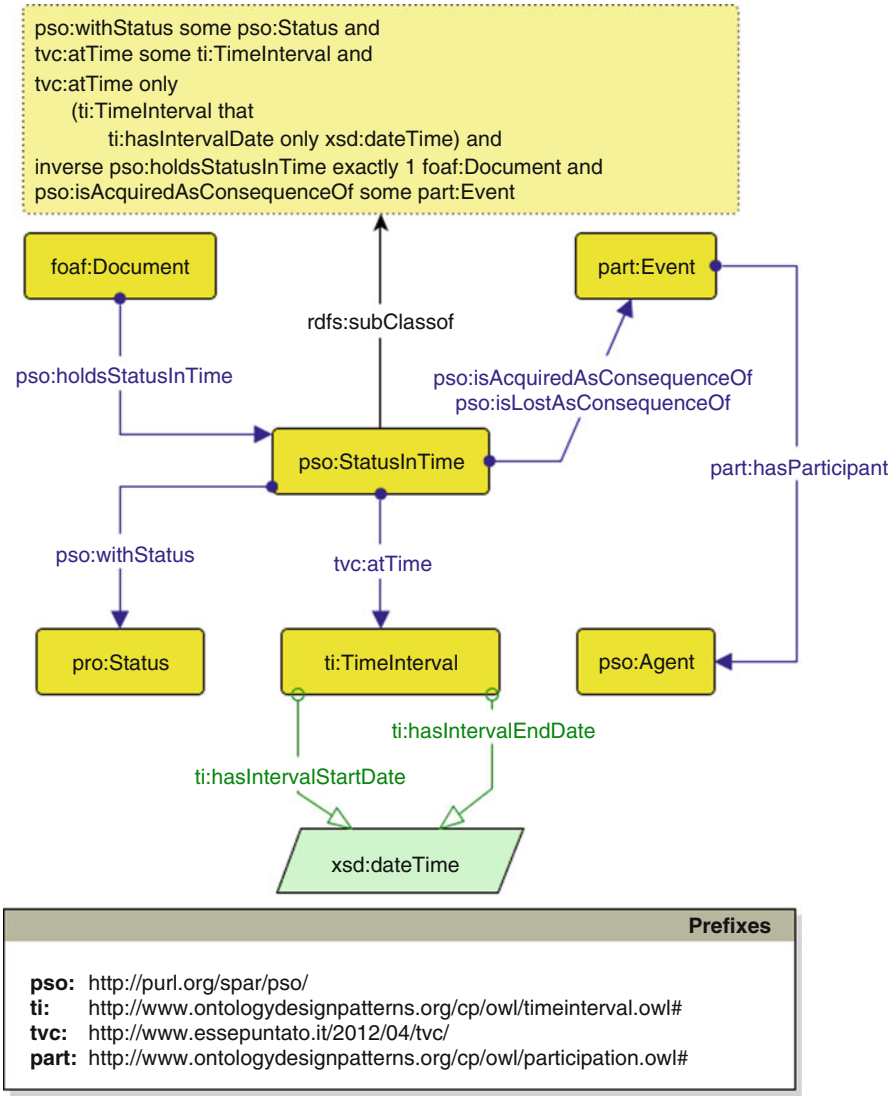

Fig. 5.12 Graffoo representation of the Publishing Status Ontology (*PSO*)

```
:earmark-paper
  pso:holdsStatusInTime [ a pso:StatusInTime
    ; pso:withStatus pso:submitted
    ; tisit:atTime [ a ti:TimeInterval
      ; ti:hasIntervalStartDate
        "2009-04-24T13:18:21Z"^^xsd:dateTime ]
    ; pso:isAcquiredAsConsequenceOf [
      a part:Event
      ; dcterms:description "An author
      submitted the paper through the online
      conference submission system."] ]
```

Table 5.3 Alignments between TVC and PSO

TVC entity	PSO entity	Description
ValueInTime	StatusInTime	The class of the particular situation of the status a document has at a particular time as consequence of one or more events.
hasValue	hasStatusInTime	The object property linking a *foaf:Document* (i.e., any bibliographic work) to a *pro:StatusInTime*.
withValue	withStatus	The object property linking the situation to the status a document has. Currently, 26 statuses are defined in the PSO ontology as individuals of the class *pro:Status*.
Instant or Interval	TimeInterval	Two (starting and ending) points in time that define a particular period related to (object property *tisit:atTime*) a *pso:StatusInTime* situation.
withinContext	isAcquiredAsConsequenceOf and isLostAsConsequenceOf	Object properties linking a situation to the events (*part:Event*) of any publishing process that changes the status of the document (e.g., writing a draft, submitting a preprint for publication, or publishing the final paper).

```
, [ a pso:StatusInTime
  ; pso:withStatus pso:under-review
  ; tisit:atTime [ a ti:TimeInterval
    ; ti:hasIntervalStartDate
      "2009-04-26T12:00:00Z"^^xsd:dateTime
  ; ti:hasIntervalEndDate
    "2009-05-27T17:38:01Z"^^xsd:dateTime ]
; pso:isAcquiredAsConsequenceOf [
  a part:Event
  ; dcterms:description "The editorial
  committee sent the paper to reviewers for
  consideration." ]
; pso:isLostAsConsequenceOf [ a part:Event
; dcterms:description "The reviewers
  completed their reviews of the paper." ].
```

5.6 Describing Publishing Workflows with PWO

Keeping track of the publication processes is a crucial task for publishers. This activity allows them to produce statistics on their goods (e.g., books, authors, editors) and to understand whether and how their production changes over time. Organisers of particular events, such as academic conference, have similar needs. Tracking the number of submissions in the current edition of the conference, the number of accepted papers, the review process, and etc., are important statistics that can be used to improve the review process in future edition of the conference itself.

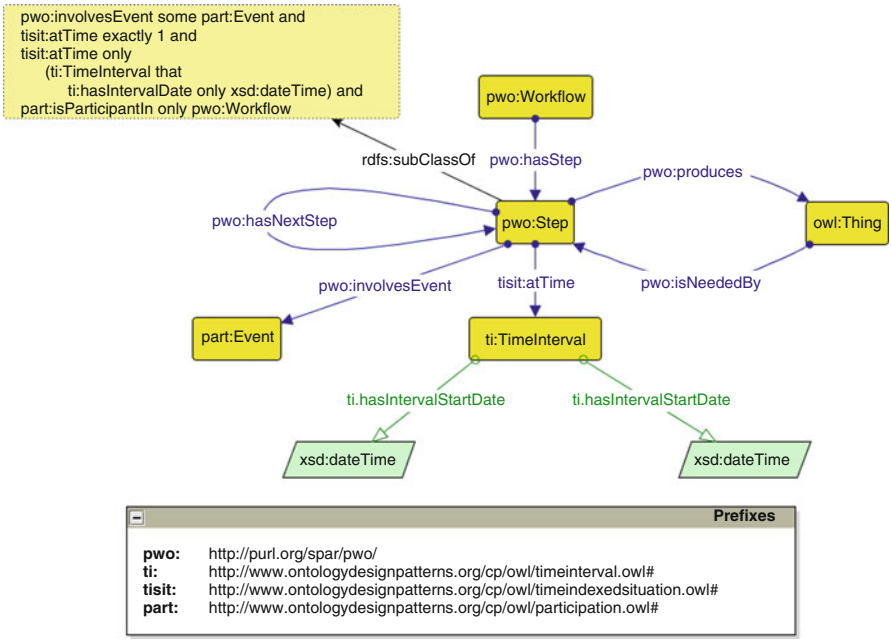

Fig. 5.13 Graffoo representation of the Publishing Workflow Ontology (*PWO*)

Some communities have started to publish[53] data which describes those events as RDF statements in the Linked Data, in order to allow software agents and applications to check and reason on them and to infer new information. However, the description of processes, for instance the peer-review process or the publishing process, is something that is not currently handled—although sources of related raw data exist. Furthermore, having these types of data publicly available would increase the transparency of the aforementioned processes and allow their use for statistical analysis. Of course, a model for describing these data is needed. Moreover, the model should be easy to integrate and adapt according to the needs and constraints of different domains (publishing, academic conferences, research funding, etc.).

In order to accommodate these requirements, I developed the *Publishing Workflow Ontology*[54] (*PWO*). This ontology allows one to describe the logical steps in a workflow, as for example the process of publication of a document. Each step may involve one or more events that take place to a particular phase of the workflow (e.g., authors are writing the article, the article is under review, reviewer suggests to revise the article, the article is in printing, the article has been published, etc.).

As shown in Fig. 5.13, PWO is based on two main classes, which are:

[53] Semantic Web Dog Food: http://data.semanticweb.org.

[54] PWO, the Publishing Workflow Ontology: http://purl.org/spar/pwo. The prefix *pwo* refers to entities defined in it.

- class *pwo:Workflow*. It represents a sequence of connected tasks (i.e., steps) undertaken by the agents. A workflow may be seen as an abstract model of a real-life work;
- class *pwo:Step*. It is an atomic unit of a workflow; it is characterised by a starting time and an ending time, and it is associated with one or more events. A workflow step usually involves some input information, material or energy needed to complete the step, and some output information, material or energy produced by that step. In the case of a publishing workflow, a step typically results in the creation of a publication entity, usually by the modification of another pre-existing publication entity, e.g., the creation of an edited paper from a rough draft, or of an HTML representation from an XML document.

In the following sections I introduce two example of application of PWO for the description of workflow processes in two different domain. First, in Sect. 5.6.1, I show how to describe the process of publication of a scholarly article and I also introduce the main features and ontological component that PWO includes for modelling these kinds of scenario. Then, in Sect. 5.6.2, I show how to use PWO to describe workflow scenarios that are not directly connected to the scholarly publishing domain, such as the description of the process of codification of statutes according to the United States legislation.

5.6.1 An Example of Workflow in Scholarly Publishing

The following excerpt presents the PWO description of the workflow which describes the publication of an article (i.e., the resource *:earmark-paper*) introduced in the example in Sect. 5.5:

```
:workflow a pwo:Workflow
  ; pwo:hasFirstStep :stepOne
  ; pwo:hasStep :stepTwo , :stepThree , :stepFour .

:stepOne a pwo:Step # Authors write the paper
  ; pwo:involvesEvent [ a part:Event
    ; dcterms:description "Authors write the paper" ]
  ; tisit:atTime [ a ti:TimeInterval
    ; ti:hasIntervalStartDate
      "2009-02-14T00:00:00Z"^^xsd:dateTime
    ; ti:hasIntervalEndDate
      "2009-03-25T00:00:00Z"^^xsd:dateTime ]
  ; pwo:produces :earmark-paper
  ; pwo:hasNextStep :stepTwo .

:stepTwo a pwo:Step # Paper submitted
  ; pwo:involvesEvent :author-submits-paper
  ; tisit:atTime [ a ti:TimeInterval
    ; ti:hasIntervalStartDate
      "2009-04-24T13:18:21Z"^^xsd:dateTime
    ; ti:hasIntervalEndDate
      "2009-04-24T13:18:21Z"^^xsd:dateTime ]
  ; pwo:needs :earmark-paper
  # New status in time for the paper
  ; pwo:produces :submitted
  ; pwo:hasNextStep :stepThree .

:stepThree a pwo:Step # Paper reviewed
  ; pwo:involvesEvent
    :reviewers-working , :reviewers-finish
  ; tisit:atTime [ a ti:TimeInterval
    ; ti:hasIntervalStartDate
      "2009-04-26T12:00:00Z"^^xsd:dateTime
    ; ti:hasIntervalEndDate
      "2009-05-26T12:00:00Z"^^xsd:dateTime ]
  ; pwo:needs :earmark-paper
  ; pwo:produces :review1 , :review2 , review3 .
  ; pwo:hasNextStep :stepFour .

:review1 a fabio:Comment # Review 1
  ; frbr:realizationOf [ a fabio:Review ]
  ; cito:reviews :earmark-paper
  ; pro:isDocumentContextFor [ a pro:RoleInTime
    ; pro:withRole pro:author
    # First anonymous reviewer
    ; pro:isRoleHeldBy [ a foaf:Person
      ; pro:hasRoleInTime [ a pro:RoleInTime
        ; pro:withRole pro:reviewer
        ; pro:relatesToDocument :earmark-paper ] ] .
```

```
:review2 a fabio:Comment ...

:stepFour a pwo:Step # Notification of acceptance
  ; pwo:involvesEvent
    :committee-accepts , :committee-notifies-to-authors
  ; tisit:atTime [ a ti:TimeInterval
    ; ti:hasIntervalStartDate
      "2009-04-26T12:00:00Z"^^xsd:dateTime
    ; ti:hasIntervalEndDate
      "2009-05-27T17:38:01Z"^^xsd:dateTime ]
  ; pwo:needs :review1 , :review2 , :review3
  ; pwo:produces
     :reviewed-and-accepted-for-publication
    , :acceptance-notification .

# The e-mail notifying the acceptance
:acceptance-notification a fabio:Email
  ; frbr:realizationOf [ a fabio:Opinion ]
  ; pro:isDocumentContextFor [ a pro:RoleInTime
    ; pro:withRole pro:author
    # The committee
    ; pro:isRoleHeldBy [ a foaf:Group ] .
```

PWO was implemented according to three particular ontology patterns:

- The *time-indexed situation pattern* (Gangemi 2010d) to describe workflow steps as entities that involve a duration and that are characterised by events and objects (needed for and produced by the step).
- The *sequence pattern*[55] (Gangemi 2010c) to define the order in which steps appear within a workflow.
- The *participation pattern*[56] (Gangemi 2010a) to describe events (and eventually agents involved) taking part in the steps.

In order to be consistent, PWO implements strong constraints on the steps by means of a particular model: the *Error Ontology*[57]. This ontology is a unit test that produces an inconsistent model if a particular (and incorrect) situation happens. It works by means of a data property, *error:hasError*, that denies its usage for any resource, as shown as below (in Manchester Syntax (Horridge and Patel-Schneider 2012)):

[55] The sequence pattern: http://www.ontologydesignpatterns.org/cp/owl/sequence.owl. The prefix *seq* refers to entities defined in it.

[56] The participation pattern: http://www.ontologydesignpatterns.org/cp/owl/participation.owl. The prefix *part*refers to entities defined in it.

[57] The Error Ontology: http://www.essepuntato.it/2009/10/error. The prefix *error* refers to entities defined in it.

```
DataProperty: error:hasError
  Domain: error:hasError exactly 0
  Range: xsd:string
```

A resource that asserts to have an error makes the ontology inconsistent, since its domain is defined as "all those resources that do not have any *error:hasError* assertion".

By means of the Error Ontology, I can generate an inconsistency every time the steps of a workflow are not arranged in a correct temporal order. In particular, an error is raised when a step requires (property *pwo:needs*) to use a particular object that will be produced (property *pwo:produces*) as consequence of another sequent step. The following excerpt shows the implementation of this constraint through a SWRL rule (Horrocks et al. 2004):

```
step(?step1) , step(?step2) ,
needs(?step1,?resource) ,
produces(?step2,?resource) ,
sequence:precedes(?step1,?step2)
  -> error:hasError(?step1,"A step cannot need a
       resource that will be produced by a following
       step"^^xsd:string)
```

5.6.2 An Example of Workflow in the Legislative Domain

Although PWO had been thought in principle to describe workflows related with the publishing domain, it has been developed on purpose as an ontology for the description of generic workflows. To this end, in this section I show how to use PWO to describe a workflow that concerns the process of codification of a particular law of the United States legislation, i.e. the codification of Title 51 of the United States Code, as described in an introductory webpage of the Office of the Law Revision Counsel[58].

The first step of such a workflow was the introduction of the codification bill "H.R. 3237" by the Office of the Law Revision Counsel on July 16, 2009. The result of the first step was the production of a first version of such bill, which was accompanied by an explicative document, the *bill explanation*, which documented the modifications that should be done on the Title 51 of the Code. The step is rendered through PWO as follows[59]:

[58] Positive law codification of title 51 of the United States Code: http://uscode.house.gov/codification/t51/index.html.

[59] The RDF representation of the agents involved in this and in the following examples are taken from DBpedia, where the prefix *dbpedia* stands for http://dbpedia.org/resource/.

```
:workflow a pwo:Workflow
  ; pwo:hasFirstStep :step-one .

# introduction of the codification bill
:step-one a pwo:Step
  ; pwo:involvesEvent [ a part:Event
    ; dcterms:description "Drafting the codification
       bill H.R. 3237"
    ; part:hasParticipant
       dbpedia:Office_of_the_Law_Revision_Counsel ]
  ; tisit:atTime [ a ti:TimeInterval
    ; ti:hasIntervalEndDate
      "2009-06-16T00:00:00Z"^^xsd:dateTime ]
  ; pwo:produces
    :hr-3237-bill-first-version , :hr-3237-explanation .

:hr-3237-bill a fabio:Work
  ; frbr:realization :hr-3237-bill-first-version .

:hr-3237-bill-first-version a fabio:Expression
  ; frbr:realizer dbpedia:
     Office_of_the_Law_Revision_Counsel .

:hr-3237-explanation a fabio:Expression
  ; frbr:realizer
     dbpedia:Office_of_the_Law_Revision_Counsel
  ; frbr:supplementOf :hr-3237-bill-first-version .
```

The second step of the process involved the Committee on the Judiciary of the House of Representatives, that referred the codification bill. In particular, the Committee considered the bill in full committee markup on October 21, 2009, and ordered the bill to be reported. The (amended version of the) bill was then reported by the Committee on November 2, 2009. The amended version of the bill was also accompanied by a written report based on the bill explanation produced in the previous step. The step is rendered through PWO as follows:

```
:workflow pwo:hasStep :step-two .

:step-one pwo:hasNextStep :step-two .

# the bill was reported by the C.J.H.
:step-two a pwo:Step
  ; pwo:involvesEvent :referring-bill-committee-judiciary-house
  ; tisit:atTime [ a ti:TimeInterval
    ; ti:hasIntervalStartDate
      "2009-10-21T00:00:00Z"^^xsd:dateTime
    ; ti:hasIntervalEndDate
      "2009-11-02T00:00:00Z"^^xsd:dateTime ]
  ; pwo:needs
    :hr-3237-bill-first-version , :hr-3237-explanation
  ; pwo:produces
    :hr-3237-bill-amended , :hr-3237-report-document ,
    :hr-3237-bill-reported-by-committee-judiciary-house .

:referring-bill-committee-judiciary-house a part:Event
  ; dcterms:description
    "Referring the codification bill H.R. 3237"
  ; part:hasParticipant
    dbpedia:
       United_States_House_Committee_on_the_Judiciary .

:hr-3237-bill-amended a fabio:Expression
  ; frbr:realizationOf :hr-3237-bill
  ; frbr:realizer
    dbpedia:
       United_States_House_Committee_on_the_Judiciary
  ; frbr:revisionOf :hr-3237-bill-first-version
  ; pso:holdsStatusInTime
    :hr-3237-bill-reported-by-committee-judiciary-house .

:hr-3237-report-document a fabio:ReportDocument
  ; frbr:realizer dbpedia:
     United_States_House_Committee_on_the_Judiciary
  ; frbr:adaptionOf :hr-3237-explanation
  ; frbr:supplementOf :hr-3237-bill-amended .

:hr-3237-bill-reported-by-committee-judiciary-house
    a pso:StatusInTime
  ; pwo:withStatus :reported
  ; tvc:atTime [ a ti:TimeInterval
    ; ti:hasIntervalStartDate
      "2009-11-02T00:00:00Z"^^xsd:dateTime ]
  ; pso:isAcquiredAsConsequenceOf
    :referring-bill-committee-judiciary-house .

:reported a pso:Status .
```

In third step of the process, the bill was passed by the House of Representatives on January 13, 2010. On January 20, 2010, the bill was received by the Senate and referred to the Committee on the Judiciary of the Senate. The Committee considered the bill in full committee markup on May 6, 2010, and ordered the bill to be reported. The bill was reported by the Committee on May 10, 2010. The step is rendered through PWO as follows:

```
:workflow pwo:hasStep :step-three .

:step-two pwo:hasNextStep :step-three .

# the bill was reported by the C.J.S.
:step-three a pwo:Step
  ; pwo:involvesEvent
    [ a part:Event
      ; dcterms:description "Passing the codification
         bill H.R. 3237 by the House of Representatives"
      ; part:hasParticipant
        dbpedia:United_States_House_of_Representatives ] ,
    [ a part:Event
      ; dcterms:description "Receiving the codification
         bill H.R. 3237"
      ; part:hasParticipant dbpedia:United_States_Senate ] ,
    :referring-bill-committee-judiciary-senate
  ; tisit:atTime [ a ti:TimeInterval
    ; ti:hasIntervalStartDate
      "2010-01-13T00:00:00Z"^^xsd:dateTime
    ; ti:hasIntervalEndDate
      "2010-05-10T00:00:00Z"^^xsd:dateTime ]
  ; pwo:needs
    :hr-3237-bill-amended , :hr-3237-report ,
    :hr-3237-status-reported-by-committee-judiciary-
       house
  ; pwo:produces
    :hr-3237-status-reported-by-committee-judiciary-
       senate .

:referring-bill-committee-judiciary-senate a part:Event
  ; dcterms:description "Referring the codification bill
      H.R. 3237"
  ; part:hasParticipant
    dbpedia:
       United_States_Senate_Committee_on_the_Judiciary .

:hr-3237-bill-reported-by-committee-judiciary-senate
    a pso:StatusInTime
  ; pwo:withStatus :reported
  ; tvc:atTime [ a ti:TimeInterval
    ; ti:hasIntervalStartDate
      "2010-05-10T00:00:00Z"^^xsd:dateTime ]
  ; pso:isAcquiredAsConsequenceOf
    :referring-bill-committee-judiciary-senate .
```

Finally, in the latest step of the process, the bill was passed by the Senate on December 3, 2010. On December 18, 2010, the bill became Public Law 111–314, which was codified in Title 51 of the United States Code.

```
:workflow pwo:hasStep :step-four .

:step-three pwo:hasNextStep :step-four .

# the bill became Public Law 111-314
:step-four a pwo:Step
  ; pwo:involvesEvent
    [ a part:Event
      ; dcterms:description "Passing the codification
       bill H.R. 3237 by the Senate"
    ; part:hasParticipant dbpedia:United_States_Senate ]
  ; tisit:atTime [ a ti:TimeInterval
    ; ti:hasIntervalStartDate
      "2010-12-03T00:00:00Z"^^xsd:dateTime
    ; ti:hasIntervalEndDate
      "2010-12-18T00:00:00Z"^^xsd:dateTime ]
  ; pwo:needs
    :hr-3237-bill-amended ,
    :hr-3237-status-reported-by-committee-judiciary-
       senate
  ; pwo:produces
    :public-law-111-314 ,
    :title-51-enacted-by-public-law-111-314 .

:public-law-111-314 a fabio:Work ;
  frbr:adaptationOf :hr-3237-bill ;
  frbr:realization [ a fabio:Expression
    ; cito:providesExcerptFor
      :title-51-enacted-by-public-law-111-314 ] .

:title-51 a fabio:Work ;
  frbr:realization
    :title-51-negative-law ,
    :title-51-enacted-by-public-law-111-314 .

:title-51-enacted-by-public-law-111-314
  a fabio:Expression
  ; frbr:revisionOf :title-51-negative-law .
```

The aim of this section was twofold. On the one hand, the above excerpts have provided a running example of the use of PWO to describe processes of codification of laws of US legislation. On the other hand, as promised in Sect. 2.3.5, the excerpts have also showed how FRBR can be used in the context of the legislative process of the federal legislation of the United States.

5.7 How Communities Uptake SPAR

The SPAR ontologies are now being used or are under consideration in a variety of academic and publishing environments. The adoption of these models by different communities can be ascribed, at least in part, to the fact that we have adopted the following strategies during the development of the ontologies:

- Frequent and ongoing interactions between the authors of SPAR, and publishers, service developers and other end-users, that have allowed us to understand their various needs and interests.
- The minimisation of the constraints applied to the ontological entities, so that the ontologies can be applied in a wide variety of situations.

The following sections will briefly describe how SPAR ontologies are now being used in various communities.

5.7.1 SWAN Ontology

The most recent version (v2.0) of the *SWAN ontology ecosystem* (Ciccarese et al. 2008), introduced above in Sect. 2.3.7, has recently been harmonised to include FaBiO and it now works seamlessly with CiTO (Ciccarese et al. 2011). David Shotton and I undertook this harmonisation collaboratively, while developing version 1.6 of CiTO (Shotton 2010) into CiTO v2.0 and FaBiO v1.0, and by Paolo Ciccarese and Tim Clark (Harvard University), authors of the SWAN Ontologies. The resulting CiTO/FaBiO + SWAN model is specified in OWL 2 DL, it is fully modular, and it inherently supports agent-based searching and mash-ups.

The principles adopted for its activity, which resulted in the harmonisation described in Fig. 5.14, involved:

- The renaming of classes (concepts) or properties (relationships) in one or both set of ontologies to avoid any apparent overlap.
- The re-definition of classes or properties to resolve actual overlap between concepts.
- The deprecation of elements of individual ontologies, or even whole ontologies, in favour of others that would more effectively serve the domain of knowledge under consideration, having greater granularity or a more effective structure.

In summary, the SWAN Citations ontology module was deprecated in favour of FaBiO, certain classes in the SWAN Discourse Relationship were renamed and redefined, the property *discourse-relationships:cites* in that module was deprecated, and CiTO was linked to that module by making *cito:cites* a sub-property of *discourse-relationships:refersTo.*Full details are given in Ciccarese et al. (2011).

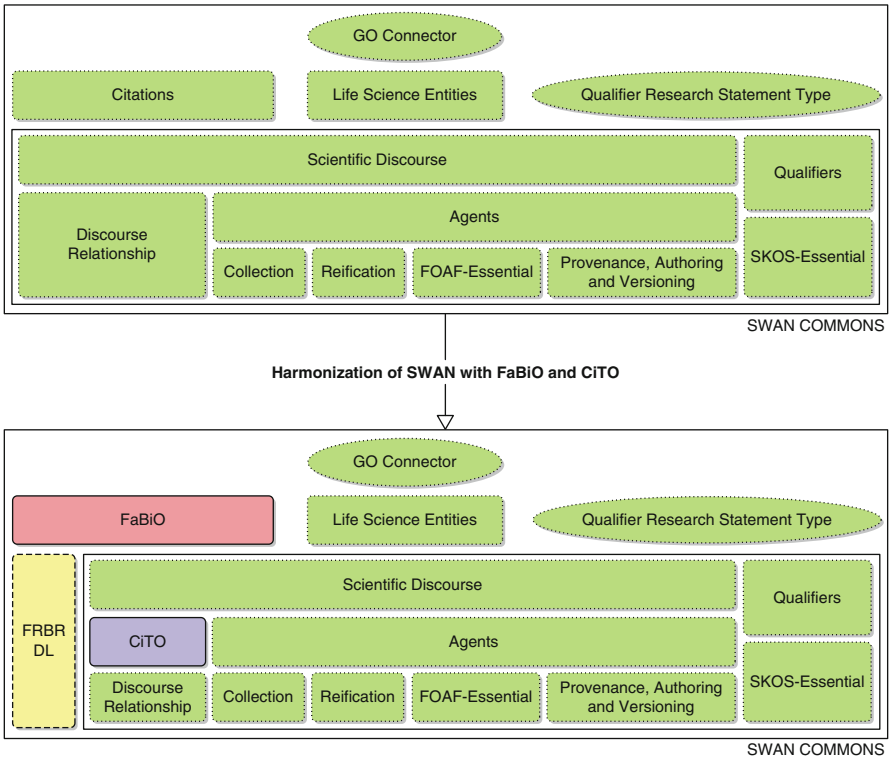

Fig. 5.14 The SWAN ontology ecosystem before (*above*) and after (*below*) the harmonisation activity that resulted in the inclusion of FaBiO and CiTO in the SWAN Commons set of ontologies

5.7.2 CiteULike

Egon Willighagen of Uppsala University has pioneered the use of CiTO[60] to characterise bibliographic citations within *CiteULike*[61], the free service for managing and discovering scholarly references. A user can add a CiTO relationship between articles via the CiteULike interface, provided that both the citing and the cited articles are in the user's library.

[60] http://chem-bla-ics.blogspot.com/2010/10/citeulike-cito-use-case-1-wordles.html.

[61] CiteULike: http://www.citeulike.org/.

5.7.3 WordPress

In a blog post[62], Martin Fenner describes a plug-in for *WordPress* called *Link-to-Link*[63], that makes it easy to add citation typing into references within a blog post, using a sub-set of the most commonly used CiTO relationships presented in a convenient drop-down menu.

5.7.4 Linked Education

Few months ago, the open platform *Linked Education*[64], which aims at sharing and promoting the use of Linked Data for educational purposes, added CiTO to its listing and, recently, all the other SPAR ontologies[65] of RDF schemas and vocabularies suitable for use in educational contexts, for example to describe educational resources, were added.

5.7.5 Virtual Observatory

In a recent paper (Accomazzi and Dave 2011), Accomazzi and Dave report the adoption of the FaBiO and CiTO ontologies as part of their efforts to create a semantic knowledge base allowing easier integration and linking of the body of heterogeneous astronomical resources into what they term a Virtual Observatory.

5.7.6 Open Citations Corpus

The Open Citations Corpus[66] is a database of approximately 6.3 million biomedical literature citations, harvested from the reference lists of all open access articles in PubMed Central. These contain references to approx. 3.4 million papers, which represent $\sim$20 % of all PubMed-listed papers published between 1950 and 2010, including all of the most highly cited papers in every biomedical field. The Open

[62] Blog post by Martin Fenner entitled "How to use citation typing ontology (CiTO) in your blog post": http://blogs.plos.org/mfenner/2011/02/14/how-to-use-citation-typing-ontology-cito-in-your-blog-posts/.

[63] Link to link: http://wordpress.org/extend/plugins/link-to-link/.

[64] Linked Education: http://linkededucation.org/.

[65] Linked Education–Schemas and vocabularies: http://linkededucation.wordpress.com/data-models/schemas/.

[66] The Open Citations Corpus: http://opencitations.net/.

Citations Corpus website allows one to browse these bibliographic records and citations, to select an individual article, and to visualise its citation network in a variety of displays. Details of each selected reference, and the data and diagrams for its citation network, may be downloaded in a variety of formats, while the entire Open Citations Corpus can be downloaded in several formats including RDF and BibJSON. SPAR ontologies have been used to encode this information in RDF. Further information is given on the Open Citations Blog[67].

5.7.7 WebTracks

WebTracks[68] is an open source project funded by the JISC Managing Research Data Programme[69] that is developing a peer-to-peer protocol to enable web-scale link tracking.

Established techniques such as OAI-PMH and the emerging Linked Web of Data provide tools to publish data for linking. WebTracks focuses on actually making these connections, particularly between research datasets and related publications. It provides a mechanism for informing the target of a hyperlink that a link has been made to that target, so that it can reciprocally link back–for example, by including the correct DOI of a published paper in the metadata of a previously published dataset to which the paper refers. WebTracks creates semantically annotated links between data resources using CiTO, yielding a graph of citation and provenance to enable web-scaled data management by exposing links between related objects.

5.7.8 Società editrice il Mulino

The Italian scholarly publishing house *Società editrice il Mulino*[70] is collaborating with the Department of Computer Science of the University of Bologna, to explore how best to benefit from Semantic Web technologies can be used for the digital publication and sharing of bibliographic objects such as books and articles, and their related metadata.

This has led to the recent prototyping of an application called *Folksauro* (the name comes from the concatenation of the words *folksonomy* and *thesaurus*). Using FaBiO and DoCO as its main ontologies, and one or more discipline-specific thesauri developed in SKOS, Folksauro allows a user to associate terms from the thesauri and/or

[67] The Open Citations Blog: http://opencitations.wordpress.com/.

[68] WebTracks: http://webtracks.jiscinvolve.org/wp/about/.

[69] JISC Managing Research Data Programme: http://www.jisc.ac.uk/whatwedo/programmes/mrd.aspx.

[70] Il Mulino: http://www.mulino.it.

free-text keywords with the whole document, and/or with its sub-parts (chapters, sections, paragraphs, etc.), by means of an intuitive interface that hides the complexity of models and languages used.

5.7.9 Utopia

Utopia Documents[71] (Attwood et al. 2010) is a novel PDF reader that semantically integrates visualisation and data-analysis tools with published research articles. It brings PDF documents to life by linking to live resources on the web and by turning static data into live interactive content, and is now being regularly used by the editors of the Biochemical Journal[72] to transform static document features into objects that can be linked, annotated, visualised and analysed interactively.

Utopia has a mechanism that deconstructs a PDF document into its constituent parts, which are then annotated using DoCO. This is useful for a number of reasons: generating bibliometric metadata; improving "mouse selection" in multi-column documents; and identifying the correct flow of text in the document, by allowing intruding text such as running headers, footers and captions to be excluded. This in turn is useful for text and data mining algorithms, which can now be targeted, for example, at "all the main text excluding intruders" or "just the text in the figure captions". Recently, the Utopia team has released a free web service, called PDFX[73], that takes a PDF document, deconstructs it, and returns DoCO-annotated XML.

In addition, the Utopia team is currently developing an Open Citations plugin that pulls bibliographic citation data live from the Open Citations Corpus and uses it to display the citation network for the paper manifested by the PDF, or for any of the articles references in the paper's reference list.

References

Accomazzi, A., and R. Dave. 2011. Semantic interlinking of resources in the virtual observatory era. ArXiv:1103.5958. http://arxiv.org/pdf/1103.5958. Accessed 30 July 2013.

Aranguren, M. E., E. Antezana, M. Kuiper, and R. Stevens. 2008. Ontology design patterns for bio-ontologies: A case study on the cell cycle ontology. *BMC Bioinformatics* 9 (5): S1. (London, United Kingdom: BioMed Central). doi:10.1186/1471-2105-9-S5-S1.

Attwood, T. K., D. B. Kell, P. McDermott, J. Marsh, S. R. Pettifer, and D. Thorne. 2010. Utopia documents: Linking scholarly literature with research data. *Bioinformatics* 26 (18): 568–574. doi:10.1093/bioinformatics/btq383.

Barabucci, G., L. Cervone, M. Palmirani, S. Peroni, and F. Vitali. 2009. Multi-layer markup and ontological structures in Akoma Ntoso. In Proceeding of the international workshop on

[71] Utopia Documents: http://getutopia.com.

[72] http://www.biochemj.org/bj/424/3/.

[73] PDFX: http://pdfx.cs.man.ac.uk.

AI Approaches to the Complexity of Legal Systems II (AICOL-II), lecture notes in computer science 6237, ed. P. Casanovas, U. Pagallo, G. Sartor, and G. Ajani, 133–149. Berlin: Springer. doi:10.1007/978-3-642-16524-5_9.

Berjon, R., T. Leithead, E. D. Navara, E. O'Connor, and S. Pfeiffer. 2013. HTML5: A vocabulary and associated APIs for HTML and XHTML. W3C candidate recommendation 6 August 2013. World Wide Web Consortium. http://www.w3.org/TR/html5/. Accessed 30 July 2013.

Bojars, U., and J. G. Breslin. 2010. SIOC core ontology specification. 25 March 2010. http://rdfs.org/sioc/spec/. Accessed 30 July 2013.

Brickley, D., and L. Miller. 2010. FOAF vocabulary specification 0.98. Namespace document, 9 August 2010-Marco Polo Edition. http://xmlns.com/foaf/spec/. Accessed 30 July 2013.

Casanovas, P., N. Casellas, C. Tempich, D. Vrandecic, and R. Benjamins. 2007. OPJK and DILIGENT: Ontology modeling in a distributed environment. *Artificial Intelligence and Law* 15 (2): 171–186. doi:10.1007/s10506-007-9036-2.

Ciccarese, P., and T. Groza. 2011. Ontology of Rhetorical Blocks (ORB). Editor's draft, 5 June 2011. World Wide Web Consortium. http://www.w3.org/2001/sw/hcls/notes/orb/. Accessed 30 July 2013.

Ciccarese, P., and S. Peroni. 2013. The collections ontology: Creating and handling collections in OWL 2 DL frameworks. To appear in *Semantic Web–Interoperability, Usability, Applicability*. doi:10.3233/SW-130121.

Ciccarese, P., E. Wu, J. Kinoshita, G. Wong, M. Ocana, A. Ruttenberg, and T. Clark. 2008. The SWAN biomedical discourse ontology. *Journal of Biomedical Informatics* 41 (5): 739–751. doi:10.1016/j.jbi.2008.04.010.

Ciccarese, P., D. Shotton, S. Peroni, and T. Clark. 2011. CiTO + SWAN: The web semantics of bibliographic records, citations, evidence and discourse relationships. To appear in *Semantic Web–Interoperability, Usability, Applicability*. doi:10.3233/SW-130098.

Cimiano, P., and J. Volker. 2005. Text2Onto—A framework for ontology learning and data-driven change discovery. In Proceedings of the 10th international conference on applications of natural language to information systems (NLDB05), lecture notes in computer science 3513, ed. A. Montoyo, R. Munoz, and E. Metais, 227–238. Berlin: Springer. doi:10.1007/11428817_21.

Crofts, N., M. Doerr, T. Gill, S. Stead, and M. Stiff. 2011. Definition of the CIDOC conceptual reference model. Version 5.0.4, November 2011. ICOM/CIDOC CRM special interest group. http://www.cidoc-crm.org/docs/cidoc_crm_version_5.0.4.pdf. Accessed 30 July 2013.

D'Arcus, B., and F. Giasson. 2009. Bibliographic ontology specification. Specification document, 4 November 2009. http://bibliontology.com/specification. Accessed 30 July 2013.

Dattolo, A., A. Di Iorio, S. Duca, A. A. Feliziani, and F. Vitali. 2007. Structural patterns for descriptive documents. In Proceedings of the 7th international conference on web engineering 2007 (ICWE 2007), lecture notes in computer science 4607, ed. L. Baresi, P. Fraternali, and G. Houben, 421–426. Berlin: Springer. doi:10.1007/978-3-540-73597-7_35.

De Waard, A. 2010a. From proteins to fairytales: Directions in semantic publishing. *IEEE Intelligent Systems* 25 (2): 83–88. doi:10.1109/MIS.2010.49.

De Waard, A. 2010b. Medium-grained document structure. http://www.w3.org/wiki/HCLSIG/SWANSIOC/Actions/RhetoricalStructure/models/medium. Accessed 30 July 2013.

Di Iorio, A., D. Gubellini, and F. Vitali. 2005. Design patterns for document substructures. Proceedings of the extreme markup languages 2005. Rockville: Mulberry Technologies, Inc. http://conferences.idealliance.org/extreme/html/2005/Vitali01/ EML2005Vitali01.html. Accessed 30 July 2013.

Di Iorio, A., S. Peroni, F. Poggi, and F. Vitali. 2012. A first approach to the automatic recognition of structural patterns in XML documents. Proceedings of the 2012 ACM symposium on Document Engineering (DocEng 2012), 85–94. New York: ACM. doi:10.1145/2361354.2361374.

Dublin Core Metadata Initiative. 2012. DCMI metadata terms. DCMI recommendation. http://dublincore.org/documents/dcmi-terms/. Accessed 30 July 2013.

Gangemi, A. 2010a. Submission: Participation. http://ontologydesignpatterns.org/wiki/Submissions:Participation. Accessed 30 July 2013.

Gangemi, A. 2010b. Submission: Region. http://ontologydesignpatterns.org/wiki/Submissions:Region. Accessed 30 July 2013.

Gangemi, A. 2010c. Submission: Sequence. http://ontologydesignpatterns.org/wiki/Submissions:Sequence. Accessed 30 July 2013.

Gangemi, A. 2010d. Submission: TimeIndexedSituation. http://ontologydesignpatterns.org/wiki/Submissions:TimeIndexedSituation. Accessed 30 July 2013.

Gangemi, A., S. Peroni, and F. Vitali. 2010. Literal reification. Proceedings of the Workshop on Ontology Pattern 2010 (WOP 2010), CEUR workshop proceedings 671, 65–66. Aachen: CEUR-WS.org. http://CEUR-WS.org/Vol-671/pat04.pdf. Accessed 30 July 2013.

Garlik, S. H., and A. Seaborne. 2013. SPARQL 1.1 query language. W3C recommendation 21 March 2013. World Wide Web Consortium. http://www.w3.org/TR/sparql11-query/. Accessed 30 July 2013.

Groza, T., K. Möller, S. Handschuh, D. Trif, and S. Decker. 2007. SALT: Weaving the claim web. In Proceedings of 6th International Semantic Web Conference and of the 2nd Asian Semantic Web Conference (ISWC 2007 + ASWC 2007), lecture notes in computer science 4825, ed. K. Aberer, K. Choi, N. F. Noy, D. Allemang, K. Lee, L. J. B. Nixon, J. Golbeck, P. Mika, D. Maynard, R. Mizoguchi, G. Schreiber, and P. Cudré-Mauroux, 197–210. Berlin: Springer. doi:10.1007/978-3-540-76298-0_15.

Groza, T., S. Handschuh, K. Möller, and S. Decker. 2007. SALT–semantically annotated LaTeX for scientific publications. In Proceedings of the fourth European Semantic Web Conference (ESWC 2007), lecture notes in computer science 4519, ed. E. Franconi, M. Kifer, and W. May, 518–532. Berlin: Springer. doi:10.1007/978-3-540-72667-8_37.

Groza, T., S. Handschuh, and S. Decker. 2011. Capturing rhetoric and argumentation aspects within scientific publications. *Journal on Data Semantics* 15: 1–36. doi:10.1007/978-3-642-22630-4_1.

Gruber, T. R. 1995. Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human Computer Studies* 43 (5–6): 907–928. doi:10.1006/ijhc.1995.1081.

Guarino, N., and C. Welty. 2002. Evaluating ontological decisions with OntoClean. *Communications of the ACM* 45 (2): 61–65. doi:10.1145/503124.503150.

Hammond, T. 2008. RDF site summary 1.0 modules: PRISM. http://nurture.nature.com/rss/modules/mod_prism.html. Accessed 30 July 2013.

Hayes, P., and C. Welty. 2006. Defining N-ary relations on the semantic web. W3C working group note 12 April 2006. World Wide Web Consortium. http://www.w3.org/TR/swbp-n-aryRelations/. Accessed 30 July 2013.

Hobbs, J. R., and F. Pan. 2006. Time ontology in OWL. W3C working draft, 27 September 2006. World Wide Web Consortium. http://www.w3.org/TR/owl-time/. Accessed 30 July 2013.

Horridge, M., and P. Patel-Schneider. 2012. OWL 2 web ontology language Manchester syntax. 2nd ed. W3C working group note 11 December 2012. World Wide Web Consortium. http://www.w3.org/TR/owl2-manchester-syntax/. Accessed 30 July 2013.

Horrocks, I., P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean. 2004. SWRL: A semantic web rule language combining OWL and RuleML. W3C Member Submission 21 May 2004. World Wide Web Consortium. http://www.w3.org/Submission/SWRL/. Accessed 30 July 2013.

Iannella, R. 2013. vCard ontology: For describing people and organisations. W3C working draft 24 September 2013. World Wide Web Consortium. http://www.w3.org/TR/vcard-rdf/. Accessed 30 July 2013.

International Federation of Library Associations and Institutions Study Group on the Functional Requirements for Bibliographic Records. 2009. Functional requirements for bibliographic records final report. International federation of library associations and institutions. http://www.ifla.org/files/cataloguing/frbr/frbr_2008.pdf. Accessed 30 July 2013.

Kircz, J. G. 1991. Rhetorical structure of scientific articles: The case for argumentational analysis in information retrieval. *Journal of Documentation* 47 (4): 354–372. doi:10.1108/eb026884.

Masolo, C., L. Vieu, E. Bottazzi, C. Catenacci, R. Ferrario, A. Gangemi, and N. Guarino. 2004. Social roles and their descriptions. Proceedings of the 9th international conference on the principles of knowledge representation and reasoning (KR2004), 267–277. https://www.aaai.org/Papers/KR/2004/KR04-029.pdf. Accessed 30 July 2013.

Miles, A., and S. Bechhofer. 2009. SKOS simple knowledge organization system reference. W3C recommendation 18 August 2009. World Wide Web Consortium. http://www.w3.org/TR/skos-reference/. Accessed 30 July 2013.

Moller, K., S. Bechhofer, and T. Heath. 2009. Semantic web conference ontology. http://data.semanticweb.org/ns/swc/swc_2009-05-09.html. Accessed 30 July 2013.

Peroni, S., and D. Shotton. 2012. FaBiO and CiTO: Ontologies for describing bibliographic resources and citations. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web* 17:33–43. doi:10.1016/j.websem.2012.08.001.

Peroni, S., E. Motta, and M. d'Aquin. 2008. Identifying key concepts in an ontology, through the integration of cognitive principles with statistical and topological measures. In Proceedings of the 3rd Asian Semantic Web Conference (ASWC 2008), ed. J. Domingue and C. Anutariya. Berlin: Springer.

Picca, D., A. Gliozzo, and A. Gangemi. 2008. LMM: An OWL-DL MetaModel to represent heterogeneous lexical knowledge. Proceedings of the 6th Language Resource and Evaluation Conference (LREC 2008). Luxembourg: European Language Resources Association. http://www.lrec-conf.org/proceedings/lrec2008/pdf/608_paper.pdf. Accessed 30 July 2013.

Presutti, V., and A. Gangemi. 2008. Content ontology design patterns as practical building blocks for web ontologies. In Proceedings of the 27th international conference on conceptual modeling (ER 2008), lecture notes in computer science 5231, ed. Q. Li, S. Spaccapietra, E. S. K. Yu, and A. Olivé, 128–141. Berlin: Springer. doi:10.1007/978-3-540-87877-3_11.

Prud'hommeaux, E., and G. Carothers. 2013. Turtle, Terse RDF triple language. W3C candidate recommendation 19 February 2013. World Wide Web Consortium. http://www.w3.org/TR/turtle/. Accessed 30 July 2013.

Rector, A. 2003. Modularisation of domain ontologies implemented in description logics and related formalisms including OWL. In Proceedings of the 2nd international conference on Knowledge Capture (K-CAP 2003), ed. J. H. Gennari, B. W. Porter, and Y. Gil. New York: ACM.

Schneider, J., T. Groza, and A. Passant. 2011. A review of argumentation for the social semantic web. *Semantic Web–Interoperability, Usability, Applicability* 4 (2): 159–218. doi:10.3233/SW-2012-0073.

Shotton, D. 2009. Semantic publishing: The coming revolution in scientific journal publishing. *Learned Publishing* 22 (2): 85–94. doi:10.1087/2009202.

Shotton, D. 2010. CiTO, the citation typing ontology. *Journal of Biomedical Semantics* 1 (1): S6. doi:10.1186/2041-1480-1-S1-S6.

Shotton, D., C. Caton, and G. Klyne. 2010. Ontologies for sharing, ontologies for use. http://ontogenesis.knowledgeblog.org/2010/01/22/ontologies-for-sharing/. Accessed 12 March 2012.

Toulmin, S. 1959. *The uses of argument*. Cambridge: Cambridge University Press. (ISBN 0521827485).

Varma, P. 2010. Project documents ontology. http://vocab.deri.ie/pdo. Accessed 30 July 2013.

Walsh, N. (2010). DocBook 5: *The definitive guide*. Sebastopol: O'Really Media. Version 1.0.3. (ISBN: 0596805029).

Wan, S., C. Paris, and R. Dale. 2010. Supporting browsing-specific information needs: Introducing the citation-sensitive in-browser summariser. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web* 8 (2–3): 196–202. doi:10.1016/j.websem.2010.03.002.