The Document Components Ontology (DoCO)

Editor(s): ---
Solicited review(s): ---

Alexandru Constantin[a], Silvio Peroni[b,c,*], Steve Pettifer[d], David Shotton[e], and Fabio Vitali[b]

[a] *École Polytechnique Fédérale de Lausanne*
EPFL IC IIF LSIR, BC 159 (Bâtiment BC), Station 14, CH-1015 Lausanne, Switzerland
alex.constantin@epfl.ch

[b] *Department of Computer Science And Engineering, University of Bologna*
Mura Anteo Zamboni 7, 40127 Bologna (BO), Italy
silvio.peroni@unibo.it, fabio.vitali@unibo.it

[c] *Semantic Technology Laboratory, Institute of Cognitive Sciences and Technologies, National Research Council,*
Via Nomentana 56, 00161 Rome (RM), Italy

[d] *School of Computer Science, University of Manchester*
Kilburn Building, M13 9PL Manchester, United Kingdom
steve.pettifer@manchester.ac.uk

[e] *Oxford e-Research Centre, University of Oxford*
7 Keble Road, OX1 3QG Oxford, United Kingdom
david.shotton@oerc.ox.ac.uk

Abstract. The description of document layers, as well as of the document discourse (e.g. the scientific discourse in scholarly articles) in machine-readable forms is crucial in facilitating semantic publishing and overall comprehension of documents by both users and machines. In this paper we introduce *DoCO*, the *Document Components Ontology*, i.e., an OWL 2 DL ontology that provides a general-purpose structured vocabulary of document elements to describe document parts in RDF. In addition to the formal description of the ontology, its utility in practice is showcased through several in-house solutions and other works of the Semantic Publishing community that rely on DoCO to annotate and retrieve document components of scholarly articles.

Keywords: DEO, DoCO, PDFX, SPAR ontologies, Utopia Documents, document components, structural patterns

*Corresponding author. E-mail: silvio.peroni@unibo.it.

1. Introduction

One of the most important criteria for the evaluation of a scientific contribution is the coherent organisation of the textual narrative that describes it, most often published as a scientific article or book. In most academic disciplines, such writings have well-established models of organisation and rhetorical structure, to which all scholars and contributors generally abide. These expectations are shared by academic publishers, who ask for standardised models in the submissions they receive, constructed to efficiently describe the content's organisation over logical components. This in turn corresponds to publishers' quest for a structural model that can best express the expected structure, describe the article's parts correctly, and identify at a glance any omissions, redundancies or incorrect sequences. Unfortunately, the number of distinct vocabularies adopted by publishers to describe these requirements is quite large, and a need arises to integrate these different languages into a single, unifying framework that may be used for all content, regardless of provenance and scientific context. For instance, a recent report by Beck [3] explains the requirements for an XML vocabulary of scientific journals to be acceptable for inclusion in PubMed Central [superscripted link?].

Several studies exist that discuss models and theories for the description of structural, rhetorical and argumentative functions of texts. The description of documents' layers, as well as of their discourse in machine-readable form is crucial in facilitating their correct comprehension both by users and machines [9] [28] [7]. It is also a strict requirement of the complex process of semantic publishing [32] [33]. Being able to simplify and automate the time consuming process of annotating structural and rhetorical behaviours of document components (such as identifying front/body/back matters, related works, results, etc.), may be instrumental in providing a number of services to publishers, open archives, and even scientists themselves. For instance, the correct identification of structural patterns in academic documents could be used to automatically generate lists and summaries (e.g., tables of contents, lists of figures), to render the content in a web browser, or to provide full-scale converters between different component vocabularies, readily usable by delivery and publication platforms.

This paper describes *DoCO* – the *Document Components Ontology*, an OWL 2 DL ontology that provides a general-purpose structured vocabulary of document elements. Both the structural and the rhetorical foundations of the ontology are presented, along with hybrid structures that describe components in terms of their complementary structural and rhetorical behaviour. The utility of the ontology in practice is afterwards showcased through several in-house solutions that rely on DoCO to annotate and retrieve document components of scholarly articles. In addition, other works of the Semantic Publishing community are also introduced, that directly use or promote DoCO as one of the most comprehensive ontology to model document components in RDF.

The rest of this paper is organised as follows. In Section 2 we discuss some relevant work about models describing document components. In Section 3 we give an overview of DoCO, presenting its foundations and formal characterisation to describe the organisation of documents according to both structural patterns and rhetoric structures. In Section 4 we illustrate how DoCO is presently used for tasks of annotation and document component retrieval, of high value in processes of literature management and analysis. Finally, we conclude in Section 5 and present further development planned for the near future.

2. Related Works

2.1. Semantic Publishing and Referencing ontologies

In the past, several works have proposed (Semantic Web) models, such as RDFS vocabularies and OWL ontologies, for describing particular aspects of the publishing domain, even if they have mainly concerned the description of the metadata of bibliographic resources (e.g., DCTerms[1], PRISM[2] and BIBO[3]). One of the first attempts to address the description of the whole (or, at least, the main part of) publishing domain is the introduction of the Semantic Publishing and Referencing (SPAR) ontologies[4]. SPAR is a suite of orthogonal and complementary OWL 2 ontologies that enable all aspects of the publishing process to be described in machine-readable metadata statements, encoded using the Resource Description Framework (RDF).

The original set of SPAR ontologies is composed by eight different models. The following is a brief description of seven of these, while the last one, DoCO, is appropriately discussed in Section 3:

1. The *FRBR-aligned Bibliographic Ontology* (*FaBiO*)[5] [26] is an ontology for describing entities that are published or potentially publishable (e.g., journal articles, conference papers, books), and that contain or are referred to by bibliographic references;
2. The *Citation Typing Ontology* (*CiTO*)[6] [26] is an ontology that enables characterization of the nature or type of citations, both factually and rhetorically;
3. The *Bibliographic Reference Ontology* (*BiRO*)[7] [11] is an ontology meant to define bibliographic records, bibliographic references, and their compilation into bibliographic collections and bibliographic lists, respectively;
4. The *Citation Counting and Context Characterisation Ontology* (*C4O*)[8] [11] is an ontology that permits the number of in-text citations of a cited source to be recorded, along with the number of citations a cited entity has received globally on a particular date;
5. The *Publishing Roles Ontology* (*PRO*)[9] [27] is an ontology for the characterisation of the roles of agents – people, corporate bodies and computational agents in the publication process. These agents can be, e.g. authors, editors, reviewers, publishers or librarians;
6. The *Publishing Status Ontology* (*PSO*)[10] [27] is an ontology designed to characterize the publication status of documents at each stage of the publishing process (draft, submitted, under review, etc.);

1 DC Terms: http://purl.org/dc/terms.

2 PRISM: http://www.prismstandard.org/resources/mod_prism.html.

3 BIBO: http://purl.org/ontology/bibo/.

4 Semantic Publishing and Referencing ontologies: http://www.sparontologies.net.

5 FaBiO: http://purl.org/spar/fabio.

6 CiTO: http://purl.org/spar/cito.

7 BiRO: http://purl.org/spar/biro.

8 C4O: http://purl.org/spar/c4o.

9 PRO: http://purl.org/spar/pro.

10 PSO: http://purl.org/spar/pso.

7. The *Publishing Workflow Ontology* (*PWO*)[11] [16], is a simple ontology for describing the steps in the workflow associated with the publication of a document or other publication entity.

The above seven ontologies, along with the Document Components Ontology (DoCO), form the original set of SPAR ontologies. This set has more recently been extended with four other complementary ontologies that extend the coverage of the possible description of the publishing domain. These are as follows:

- The *Scholarly Contributions and Roles Ontology* (*SCoRO*)[12] - an ontology based on PRO for describing the contributions that may be made, and the roles that may be held by a person with respect to a journal article or other publication (e.g. the role of article guarantor or illustrator);
- The *Funding, Research Administration and Projects Ontology* (*FRAPO*)[13] is an ontology for describing the administrative information of research projects, e.g., grant applications, funding bodies, project partners, etc.;
- The *DataCite Ontology*[14] is an ontology that enables the metadata properties of the *DataCite Metadata Schema Specification*[15] (i.e., list of metadata properties for the accurate and consistent identification of a resource for citation and retrieval purposes) to be described in RDF;
- The *Bibliometric Data Ontology* (*BiDO*)[16] [23], is a modular ontology that allows the description of numerical and categorial bibliometric data (e.g., journal impact factor, author h-index, categories describing research careers) in RDF.

Still actively maintained, the SPAR ontologies has drawn the attention of the Semantic Publishing community, as a reference point for standardising entity descriptions and fostering interoperability between services – as largely discussed in Section 4.

2.2. Existing models describing document components

To the best of our knowledge, the first concrete attempt at describing document components by means of Semantic Web technologies is the *Semantically Annotated LaTeX* (*SALT*) project[17] [18] [19]. SALT includes a set of ontologies for the description of the semantic organisation of documents according to three different layers: the structural layer (*Document Ontology*), describing sentences, paragraphs, figures, and the like; the rhetorical layer (*Rhetorical Ontology*), describing logical entities such as background knowledge, claims and evidence; and the annotation layer (*Annotation Ontology*) to link rhetorical characterisations with structural components.

Similar to the above, the *SWAN biomedical discourse ontology* [6] is a set of complementary OWL 2 DL ontologies that describe the discourse of scientific papers, with particular regard to the biomedical domain. The *Discourse elements ontology*[18] that forms part of SWAN allows one to characterise the parts of a text referring to claims, hypotheses, research questions and statements, while the relations among these and other document elements are defined in the *Discourse relationships ontology*[19] [5].

In [4], Ciccarese and Groza introduce the *Ontology of Rhetorical Blocks* (*ORB*)[20]. ORB is a model to describe large blocks of text (e.g., sections) in a rhetorical way, by capturing their logical roles within the whole scientific discourse of an article. In particular, the ontology defines seven different rhetorical blocks: one describing the front matter of the article (i.e., *orb:Head*), four blocks describing the major divisions of the body text (i.e., *orb:Introduction*, *orb:Methods*, *orb:Results*, and *orb:Discussion*), and two blocks referring to the back matter (i.e., *orb:Acknowledgements* and *orb:References*).

A detailed review and analysis of other RDF/OWL vocabularies and ontologies targeting the description of document components in terms of argumentative elements is presented by Schneider *et al*. in [31].

Other non-OWL proposals describing the possible structures that are used in documents also exist. An example is the *Medium-Grained structure* [10] devised by the W3C Scientific Discourse Task Force, which offers a medium-grained description (hypothesis, objects of study, direct representation of measurements, etc.) of the rhetorical components of a document.

From a more syntactical point of view, Tannier *et al*. [34] associate each (XML) element in a document to one of three different categories: *hard elements* - elements that are commonly used to structure the document content in different blocks and usually interrupt the linearity of a text, such as paragraphs and sections; *soft element*s - , elements that identify significant text fragments and are transparent while reading the text, such as emphasis and links; and *jump elements* - elements that are logically detached from the surrounding text, and that give access to related information, such as footnotes and comments.

Zou *et al*. [37] make Tannier *et al*.'s classification more extreme, defining only two categories of document elements: *inline* (those that do not introduce horizontal breaks) and *line-break* (those that do).

Finally, several XML vocabularies, which have been developed in the past years and that are currently used by scholarly publishers (e.g., the Elsevier Journal Article DTD[21], DocBook [36] and JATS [22]), define the most frequent structural components, such as sections, paragraphs, figures, tables, and the like. However, the same component is often expressed by different elements (e.g., a paragraph can be expressed using the elements *p*, *para*, or *par*) depending on the particular language in consideration.

---

11 PWO: http://purl.org/spar/pwo.

12 SCoRO: http://purl.org/spar/scoro.

13 FRAPO: http://purl.org/cerif/frapo.

14 DataCite Ontology: http://purl.org/spar/datacite.

15 DataCite schema: http://schema.datacite.org.

16 BiDO: http://purl.org/spar/bido.

17 Currently all the SALT ontologies seem not to be available at their original URLs. However, one can found the earliest versions of those ontologies at Linked Open Vocabularies (http://lov.okfn.org).

18 The SWAN Discourse Elements Ontology: http://purl.org/swan/2.0/discourse-elements/.

19 The SWAN Discourse Relationships Ontology: http://purl.org/swan/2.0/discourse-relationships/.

20 ORB – the Ontology of Rhetorical Blocks: http://purl.org/orb/.

21 Elsevier XML DTDs and transport schemas: http://www.elsevier.com/author-schemas/elsevier-xml-dtds-and-transport-schemas.

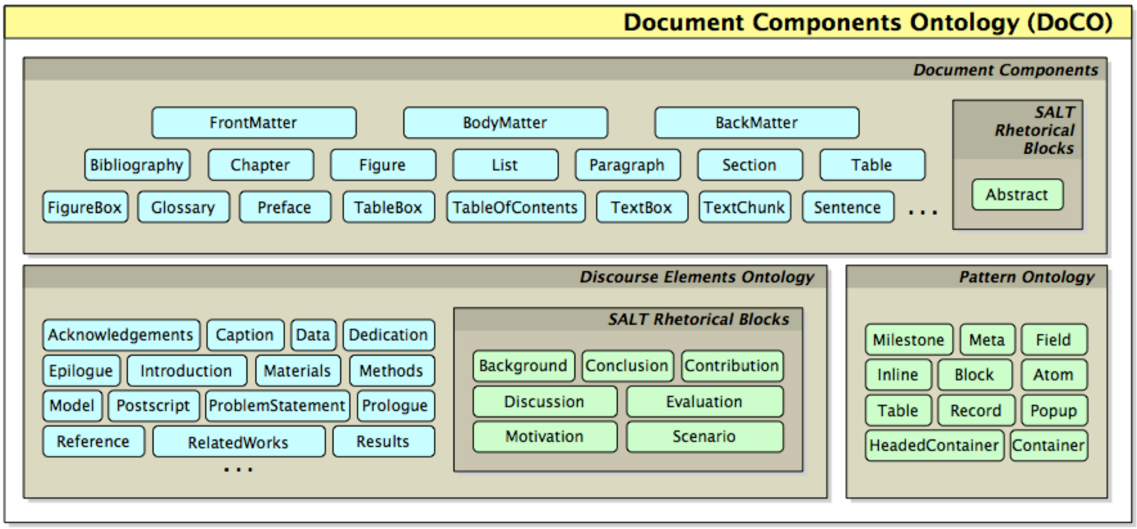

Fig. 1. Diagram describing the composition and the classes of the *Document Components Ontology* (*DoCO*).

Even if each of the aforementioned works proposes to model document components according to a particular perspective (e.g., structural vs. rhetorical, minimalistic vs. all-inclusive), a generic model harmonising all these aspects is still missing. DoCO is our tentative to cover the gap between all these different perspectives, since it is OWL model for describing all the extrinsic and intrinsic characterisations of document components.

3. Document Components

There is an intrinsic complexity in defining certain document components as purely rhetorical or purely structural. Even a well-known, easily identifiable component such at the paragraph cannot be considered as being strictly structural (i.e., carrying only a syntactic function), since it intrisically carries rhetoric as well, through its natural language sentences. Paragraphs therefore have more than a syntactic function.

However, document markup languages often define a paragraph as a pure structural component, without any reference to its rhetorical function:

- "A paragraph is typically a run of phrasing content that forms a block of text with one or more sentences" [20];
- "Paragraphs in DocBook may contain almost all inlines and most block elements" [36][22].

The above definitions emphasise the structural connotation of the paragraph, that "forms a block of text" or that "contains" other elements, and this connotation is amplified by our direct experience as readers. It is the structural aspect that readily stands out in a book or webpage and that helps us, as readers, to distinguish a paragraph from the surrounding text, yet it is insufficient for describing this element in its entirety.

The DoCO *Document Components Ontology* that we introduce below has been developed so as to bring together the purely structural characterisations of document elements and their purely rhetorical connotations.

The creation of DoCO was conducted by studying different corpora of documents (mainly scientific literature and web documents on different topics) and publishers' guidelines according to two different perspectives: the structural and the rhetorical, as also analysed by past works on document patterns [12] [13] [14]. DoCO imports the *Pattern Ontology* that describes structural patterns [13], and the *Discourse Element Ontology* that describes rhetorical components. Additionally, it also defines hybrid classes describing elements that are both structural and rhetorical at the same time, such as *paragraph*, *section* or *list*. A diagram describing the composition and the classes of DoCO is shown in Fig. 1. In the next subsections we briefly introduce our theory of structural patterns as described in [13], and the rhetorical components that usually appear in scholarly articles, which represent the theoretical underpinnings of DoCO. Then, we introduce some of the document components of DoCO relevant for the description of scientific articles. We provide their formal definitions using DL formulas.

3.1. Structural foundation: structural patterns

We have been investigating patterns of textual documents to understand how their structure can be segmented into atomic components that can be addressed independently and manipulated for different purposes. Instead of defining a large number of complex and diversified structures, in [12] we proposed a small number of structural *patterns* that are sufficient to express what most users need, characterised by two main aspects:

- *orthogonality* – each pattern needs to have a unique and specific purpose, fitting a specific context;
- *specificity* – each pattern can be used only in specific locations (e.g., within other patterns).

---

22 The words *inline* and *block* in these list items do not refer to the structural pattern theory introduced in the following section, although some sort of overlapping exist.

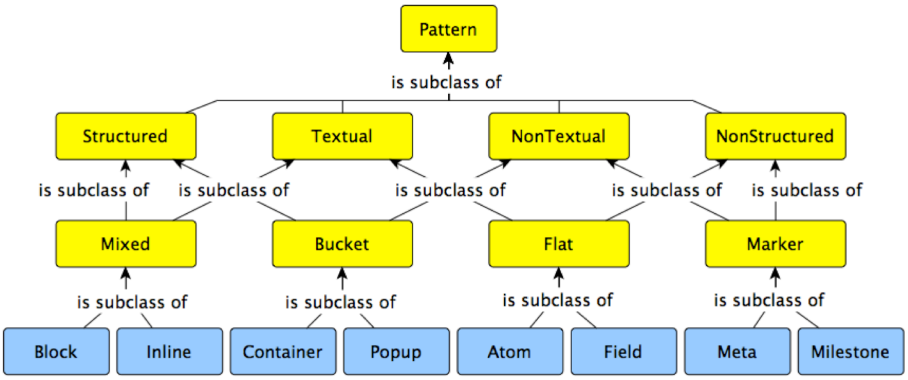

Fig. 2. A Graffoo diagram [15] describing the eight concrete patterns for document structures (bottom classes, in blue) described as particular kinds of high-level and abstract patterns (top classes, in yellow).

These patterns for textual documents were fully described in [13] and modelled as an OWL ontology called The Pattern Ontology[23], which is summarised in Fig. 2. All the patterns are defined in terms of two main kinds of entities, themselves characterised by two different properties[24]: the possibility of containing text (*po:Textual*) or not (*po:NonTextual*, disjoint with the previous one), and the possibility of being organised in substructures (*po:Structured*) or not (*po:NonStructured*, disjoint with the previous one). These basic properties are thus combined in order to obtain four different disjoint classes describing entities that (A) contain both text and substructures (*po:Mixed*), (B) contain substructures but do not contain text (*po:Bucket*), (C) contain text but do not contain substructures (*po:Flat*), (D) do not contain text, nor substructures (*po:Marker*). Each of these four classes is a superclass to two other disjoint subclasses that collectively define the eight concrete patterns that can be used to characterise structures in text. A special case is that of the pattern *po:Container*, which is further split into three more specialised subunits.

These patterns are briefly introduced in Table 1. They facilitate the creation of unambiguous, manageable and well-structured documents. The regularity of pattern-based documents (defined by means of markup languages such as DocBook or LaTeX) then makes it possible to perform complex operations easily, even when knowing very little about the documents' markup vocabulary. This in turn enables designers to implement more reliable and efficient tools [13], make hypotheses regarding the meanings of document fragments [14], identify special cases, and study global properties of sets of documents [12].

3.2. Rhetorical foundation: discourse elements

The pure rhetorical characterisation of document components is not necessarily linked to the structural organisation that a scholarly article may have. For example, some scientific journals such as the Journal of Web Semantics[25] impose that their articles follow a particular rhetorical segmentation, in order to identify explicitly what the meaningful parts are from a scientific point of view – i.e., introduction, background, evaluation, materials, methods and conclusion. These parts usually, but not necessarily, correspond to the coarse structural parts of the article – its sections. Whilst the background is usually weaved together with the introduction, it may be also presented as a separate section, or indeed substitute the introduction entirely.

The characterisations of these purely rhetorical components, which are not always linked explicitly to a particular structure, are defined in the *Discourse Element Ontology* (*DEO*)[26]. DEO provides a structured vocabulary for rhetorical elements within documents, enabling these to be described in RDF. The main class of this ontology is *deo:DiscourseElement*, which describes all those elements of a document that carry out a rhetorical function. All the remaining rhetorical behaviours are modelled as subclasses of this class. DEO reuses some of the rhetorical blocks from the SALT Rhetorical Ontology (as shown in Fig. 1), and extends them by introducing additional classes, notably:

- *deo:Reference*, which specifies a connection either to a specific part of the document or to another publication. In written text, numbered superscripts standing for footnotes, items in a table of contents, and items describing entities in a reference section, can be modelled as individuals of this class;
- *deo:BibliographicReference*, a subclass of the *deo:Reference* that describes references to other publications, such as journal articles, books, book chapters or websites; such references are often contained in a footnote or a bibliographic reference list;
- *deo:Caption, that* defines the text accompanying another item (e.g., a picture);
- *deo:Introduction*, the initial description that states the purpose and goals of the subsequent text;
- *deo:Material*, that documents the specific materials used in the described work;
- *deo:Methods*, that documents the methods used in the work (may be combined with a description of the materials used);
- *deo:Result*, that describes a report of the specific findings of an investigation;
- *deo:RelatedWork*, that describes a critical review of current knowledge by specific

23 Pattern Ontology: http://www.essepuntato.it/2008/12/pattern.

24 All prefixes are declared in http://www.essepuntato.it/2014/doco/prefixes.

25 Journal of Web Semantics Guide for Authors: http://www.elsevier.com/journals/journal-of-web-semantics/1570-8268/guide-for-authors.

26 Discourse Elements Ontology: http://purl.org/spar/deo.

Table 1. Eight (plus three) structural patterns for descriptive documents.

Pattern	Description	Example
po:Atom	Any simple box of text, without internal substructures, that is allowed in a mixed content structure but not in a container.	The various parts composing a free-text bibliographic reference of an article (title, source, etc.)
po:Block	Any container of text and other substructures except for (even recursively) other block elements.	A paragraph, a cell in a table
po:Container	Any container of a sequence of other substructures that does not directly contain text.	The body part of the article, a floating box containing a figure
po:Field	Any simple box of text, without internal substructures that is allowed in a container but not in a mixed content structure.	An e-mail of an author specified in the front matter of an article
po:Inline	Any entity containing text and other substructures, including (even recursively) other inline elements.	An emphasis, an hyper-textual link
po:Meta	Any content-less structure (but data could be specified in attributes) that is allowed in a container but not in a mixed content structure.	A marker identifying the corresponding author of an article
po:Milestone	Any content-less structure (but data could be specified in attributes) that is allowed in a mixed content structure but not in a container.	A picture inserted in the body of the article
po:Popup	Any structure that, while still not allowing text content inside itself, is nonetheless found in a mixed content context and interrupts but does not break the main flow of the text.	A footnote, a comment
po:HeadedContainer (subtype of po:Container)	Any container starting with a head of one or more block elements. The pattern is usually employed to represent nested hierarchical elements as well as their headings.	A section or subsection of the article with its heading
po:Record (subtype of po:Container)	Any container that does not allow substructures to repeat themselves internally. The pattern is meant to represent database records with their variety of (non-repeatable) fields.	The set containing the metadata concerning the authors of the article (first name, family name, address, affiliation list, email, etc.)
po:Table (subtype of po:Container)	Any container that allows a repetition of homogeneous substructures. The pattern is meant to represent a table of a database with its content of multiple similarly structured records.	A table (as a sequence of ordered rows) or a list (as a sequence of ordered items) inserted in the body of the article

reference to other relevant works, both in terms of substantive findings and theoretical and methodological contributions within a domain of study;

- *deo:FutureWork*, a proposal for new investigations to be undertaken in order to continue and advance the work described in the publication.

Note that it is still possible to apply two different rhetorical characterisations to the same block of text. For instance, in journal articles it is common to have a section entitled "Materials and Methods", which can be characterised rhetorically by using both the classes *deo:Methods* and *deo:Materials*.

3.3. Hybrid structures in DoCO

In this subsection, we introduce those classes of DoCO that bring together both the purely structural behaviour (i.e., the structural patterns introduced in Section 3.1) and the generic rhetorical characterisation (i.e., the rhetorical components recounted in Section 3.2). We focus particularly on the structures that usually define the main components of scientific papers[27].

The class *Sentence* describes all those expressions in natural language forming single grammatical units. Usually, in written text, a sentence is terminated by major punctuation, such as a full stop, a colon, a semi-colon, etc. It is defined in DoCO as follows:

```
Sentence ⊑ deo:DiscourseElement ⊓ po:Inline
```

27 DoCO actually counts more classes than those described herein, covering also other kinds of bibliographic entities, such as books and poems.

A *paragraph* is a self-contained unit of discourse that deals with a particular point or idea, structured in one or more sentences. In written text, the start of a paragraph is indicated by beginning on a new line, which may be indented or separated by a small vertical space from the preceding paragraph. In DoCO, the class *Paragraph* is disjoint with *Sentence* and is modelled as follows[28]:

```
Paragraph ⊑
  deo:DiscourseElement ⊓ po:Block ⊓
  ∃po:contains.Sentence
```

A *footnote* is a particular structure that permits the author to make a comment or to cite another publication in support of the text, or both. A footnote is normally flagged by a superscript marker (e.g., a number) immediately following the portion of text to which it relates. For convenience of reading, the text of the footnote is usually printed at the bottom of the page or at the end of a text. The DoCO class *Footnote* is disjoint with the previous classes and is defined as follows[29]:

28 In this and the following description logic excerpts, we use some properties that are defined in imported ontologies. In particular, *po:contains*, and its inverse *po:isContainedBy*, are object properties defined in the Pattern Ontology that allows us to specify explicitly containment relations among pattern-based elements (in particular, those having type *po:Structured*). In DoCO, these two properties are defined as sub-properties of *dcterms:hasPart* and *dcterms:isPartOf* respectively. Note that even if it is not explicitly stated in DoCO, we consider these DC Terms object properties to be transitive.

29 Potentially there exist two different ways of organising footnotes, since their structural semantics can depend on the particular (markup) language we use to express it, as discussed in [14]. For instance, a container-based behaviour is adopted by JATS [22], that allows one to specify footnotes (through the element *fn*) by using an element that is totally separated from the main text where it is referred to (usually through XML attributes). The popup-based behaviour, instead, is typical in LaTeX (by using the marker *\footnote{}*), where a paragraph can be abruptly interrupted by other paragraphs specified in a footnote.

```
Footnote ⊑
  deo:DiscourseElement ⊓ (po:Container ⊔ po:Popup)
```

A *table* is a set of data arranged in cells within rows and columns. From a pure structural pattern perspective, the element identifying the whole structure is organised according to the pattern *po:Table,* while those elements identifying the rows are always containers. The DoCO class *Table* is disjoint with the previous classes and is defined as follows:

```
Table ⊑
  deo:DiscourseElement ⊓ po:Table ⊓
  ∃po:contains.po:Container
```

A *figure* is a communication object comprising one or more graphics, drawings, images, or other visual representations. In DoCO, it is disjoint with the previous classes is modelled as a flat element without textual content, as introduced in the following definition:

```
Figure ⊑
  deo:DiscourseElement ⊓ (po:Milestone ⊔ po:Meta)
```

Commonly, in scientific publications, figures and tables are placed in captioned boxes (i.e., a *po:Container* containing a caption). The class *CaptionedBox* is disjoint with the previous classes and is defined as follows:

```
CaptionedBox ⊑
  deo:DiscourseElement ⊓ po:Container ⊓
  ∃dcterms:hasPart.deo:Caption
```

Captioned boxes can be used to define a space within a document that contains either a figure (i.e., *FigureBox*) or a table (i.e., *TableBox*) and its caption. These two classes are mutually disjoint and are defined respectively as follows:

```
FigureBox ⊑
  CaptionedBox ⊓ ∃dcterms:hasPart.Figure
TableBox ⊑
  CaptionedBox ⊓ ∃po:contains.Table
```

A *list* is an enumeration of items, which may be paragraphs, author names, bibliographic references, etc., delimited by distinct graphical symbols, either inline with the article text, or following a uniform spatial alignment. In DoCO, the class *List* is disjoint with the previous classes and is defined as follows:

```
List ⊑
  deo:DiscourseElement ⊓ po:Table ⊓
  ∃po:contains.po:Pattern ⊓
  ∀po:contains.((po:Container ⊓ ¬ (po:Table ⊔
    po:HeadedContainer)) ⊔ po:Field ⊔ po:Block)
```

This class is particularly useful to describe other, more specific kinds of lists: table of contents, list of figures, list of tables, etc. In particular, the class *BibliographicReferenceList* describes a list, usually within a bibliography, of all the references within the citing document that refer to articles, books, chapters, websites or similar publications. It is defined in DoCO as follows:

```
BibliographicReferenceList ≡
  List ⊓ ∀po:contains.deo:BibliographicReference
```

All above textual or graphical constructs are usually contained in broader elements that aim to describe the overall organisation of the document structure. First, we have the *front matter*, i.e., the initial principal part of a document, usually containing self-referential metadata. Although in a book it can be quite extensive, in a journal article the front matter is normally restricted to the title, authors and the authors' affiliation details, although the latter may alternatively be included in a footnote or the back matter. The DoCO class *FrontMatter* is disjoint with the previous classes and is defined as follows:

```
FrontMatter ⊑
  deo:DiscourseElement ⊓ po:Container ⊓
  ∀po:isContainedBy.(¬ (BodyMatter ⊔ BackMatter))
```

Following the front matter, the *body matter* describes the central principal part of a document, that contains the core discourse of the work. The class *BodyMatter* is disjoint with the previous classes and is defined as follows:

```
BodyMatter ⊑
  deo:DiscourseElement ⊓ po:Container ⊓
  ∀po:isContainedBy.(¬ (FrontMatter ⊔ BackMatter))
```

The *back matter* is the final principal part of a document, usually comprising the bibliography, index, appendices, etc. Disjoint to both the previous classes, it is defined as follows:

```
BackMatter ⊑
  deo:DiscourseElement ⊓ po:Container ⊓
  ∀po:isContainedBy.(¬ (FrontMatter ⊔ BodyMatter))
```

The aforementioned elements are composed of other textual structures used for a coarse-grained and hierarchical organisation of text, such as *chapters* and *sections*. Both the classes *Chapter* and *Section* describe entities used for logically dividing the text, organised in paragraphs and possibly other (sub)sections, numbered and/or titled. While chapters and sections may contain (sub)sections, they cannot contain any other chapter. They are mutually disjoint and also disjoint with the previous classes, and are defined in DoCO as follows:

```
Chapter ⊑
  deo:DiscourseElement ⊓ po:HeadedContainer ⊓
  ∃po:contains.(Paragraph ⊔ Section) ⊓
  ∀po:contains.(¬ Chapter)
Section ⊑
  deo:DiscourseElement ⊓ po:HeadedContainer ⊓
  ∃po:contains.(Paragraph ⊔ Section) ⊓
  ∀po:contains.(¬ Chapter)
```

Articles normally have particular kinds of sections (and even chapters, sometimes) that have a particular structural and rhetorical function, such as the *bibliography* or the *abstract*. The former contains a list of bibliographic references, and the related DoCO class *Bibliography* is defined as follows:

```
Bibliography ⊑
  (Section ⊔ Chapter) ⊓
  ∃dcterms:hasPart.BibliographicReference
```

The latter kind of section/chapter, defined by the class *sro:Abstract* imported from the SALT Rhetorical Ontology, describes a brief summary of a bibliographic entity, the purpose of which is to help the reader quickly ascertain the publication's purpose and points of focus. In DoCO, it is disjoint with *Bibliography* and defined as follows:

```
sro:Abstract ⊑
  (Section ⊔ Chapter) ⊓
  ∃dcterms:isPartOf.(FrontMatter ⊔ BodyMatter)
```

Sections and other high-level constructs such as chapters, captioned boxes or the document itself, can be introduced by a *title*. The DoCO class *Title* was introduced to describe a word, phrase or sentence that precedes and indicates the subject of a document or a document component. It is disjoint with the previous classes and is defined as follows:

```
Title ⊑
  deo:DiscourseElement ⊓ (po:Block ⊔ po:Field) ⊓
  ∃po:isContainedByAsHeader.po:HeadedContainer
```

Starting from the above definition, it is then easy to describe particular kinds of titles, such as *section titles* or *chapter titles* modelled as the title being part of a particular section/chapter:

```
SectionTitle ⊑
  Title ⊓ ∃po:isContainedByAsHeader.Section
ChapterTitle ⊑
  Title ⊓ ∃po:isContainedByAsHeader.Chapter
```

A (partial) RDF description of this paper according to DoCO is available online[30].

4. Adoption and uses of DoCO

This section represents an evaluation of the uses of DoCO, made by listing its adoption in different application scenarios involving the works of different research groups. In particular, we discuss some relevant applications of DoCO in tools and algorithms for the annotation and processing of scholarly articles developed by our two research groups, one at the University of Bologna, and another at the University of Manchester in the past years. In addition, at the end of this section, we briefly list other external works that concretely use DoCO for different purposes within the Semantic Publishing community.

4.1. Processing scholarly articles: PDFX

PDFX[31] [7] [8] is a rule-based system for analysing scientific publications in PDF form and recovering their fine-grained logical and rhetorical structures. Its analysis result is stored in an XML format that describes the document's organisation over logical units, and also links it to geometrical typesetting markers in the original PDF, such as column or page breaks. As of version 1.9, PDFX can differentiate 19 different element types. These types, given in Table 2, cover the principal parts of a typical research article.

Table 2. The rhetorical element types that PDFX can differentiate.

Front matter	Body matter	Back matter/others
Title	Body text	Bibliographic item
Author	(Sub)section	URI
Abstract	(Sub)section heading	Email
Author Footnote	Image	Side note
	Table	Header/Footer
	Caption	Page number
	Figure/Table reference	
	Bibliographic reference (in-text citation)	
	Labelled formula	

The identified elements are ultimately stored in an XML file with a tag hierarchy that closely follows the ANSI/NISO Journal Article Tag Suite standard (JATS) [22]. The semi-structured nature of the XML serves as a convenient, quick access route to any of the articles components.

A "class" attribute has been added to each XML element in order to facilitate interoperability with other services. This attribute is derived from the tag given to an element in the identification stage and is set in accordance with DoCO. This procedure facilitates aligning the structure recognition output of PDFX to the inputs that other text processing pipelines expect, and adds a valuable metadata layer to the original publication. A multitude of different-purpose workflows can treat the PDF-to-DoCO-compliant-XML conversion as a pre-processing step, to greatly widen their application domain in terms of accepted input.

4.2. Enhancing scholarly articles: Utopia Documents

Utopia Documents[32] [1] is a PDF-reader designed to improve the user's experience of reading scholarly papers (particularly in the domain of the Life Sciences) by linking the article and its contents to online resources.

DoCO is a disciplined way for PDFX and Utopia Documents to interoperate. In particular, Utopia Documents uses PDFX to reconstruct the structure of a PDF. DoCO is used as a mechanism for tagging the output of PDFX and other Utopia Documents plugins in an interchangeable way; thus if plugins want to exchange tables/figures and references, they use DoCO annotations. Additionally, third party plugins that are used for text mining can use the tagged structure to tune their behaviour as they pass through the document (e.g., some algorithms may want to include/exclude certain sections, or to become more or less sensitive, or to include/exclude captions or references during processing). For example the mention of a particular gene or protein in the introduction or discussion sections of a paper is likely to have a very different meaning to the mention of it in the "materials and methods" section (where it is likely to be an "ingredient").

Utopia Documents works as follows. When a user opens an article, Utopia Documents uses PDFX to analyse the document's structure. DoCO *FrontMatter* features are used to identify the article in various online databases and tools, allowing Utopia Documents to display data such as Article Level or Alternative metrics, and to find entries in databases that cite the article as a whole. In the article's body, regions identified by PDFX and tagged as instances of *Image* or *Table* are converted into interactive objects allowing the user to browse the article by figures, or to export the data from tables. In the back matter, bibliographic references (i.e., *BibliographicReference* objects) are identified and linked to their in-text citation positions in the PDF, enabling users to see which articles are being cited at a particular location without the need to scroll to the reference section.

4.3. Retrieving structures from XML sources

Although the most frequent structural components are expressed in most XML vocabularies used by scholarly publishers – e.g., the Elsevier Journal Article DTD, DocBook and JATS – they are often expressed by different elements. For instance, the element *para* in DocBook and the element *p* in JATS refer to the same concept of one of a set of vertically-organized containers of text often called *paragraph*. Starting from these bases, the services previously mentioned, such as table of contents generation or in-browser rendering, should be developed according to the peculiarities of each individual markup language. DoCO represents a generic model according to which the semantics of any structural XML tag could be retrieved automatically, circumventing the need to write bespoke parsers for each encountered format.

In making steps towards addressing this issue, we have recently used DoCO as a theoretical base for the development of an ontology-aware algorithm to retrieve the meaning of markup structures in XML article sources [14] without looking at the particular markup language used, or the actual content of the document. The algorithm was developed starting from the actual specification of DoCO classes, and,then tuned according to other statistical and topological principles (e.g. the frequency of markup elements, their

30 Example of use of DoCO (in Turtle): http://www.essepuntato.it/2014/doco/example.

31 The PDFX web service: http://pdfx.cs.man.ac.uk/.

32 Utopia Documents - http://getutopia.com.

position within the document, etc.)[33]. The final goal of the algorithm was to associate a particular DoCO class to each markup element used in these documents.

We performed a preliminary test (fully described in [14]) on a dataset consisting of 117 scientific papers encoded in DocBook and published between 2008 and 2011 in the Balisage Series Conferences[34]. The documents vary a lot in their internal structure and size: from 3 Kbytes to 160 Kbytes, with an average size of about 60 Kbytes. We compared the outcomes of the algorithm with an hand-crafted gold standard created by studying the XML vocabulary originally used to mark up the documents, and by associating each of its elements with one or more DoCO structures[35]. The overall results of this test were encouraging, since the overall values of precision and recall were quite high (0.887 and 0.890, respectively).

We are currently extending the algorithm in order to try to recognise additional DoCO components through the algorithm such as *Introduction*, *RelatedWork*, *Methods*, *Evaluation*, and *Conclusion*. We are collecting a more comprehensive document test set of XML sources that will include articles coming from the PubMed Central Open Access Subset[36] and from Elsevier's Science Direct[37].

4.4. Community uptake

In addition to our works described in the previous sections, here we list some of the most important works within the Semantic Publishing community that work with or reference DoCO, according to a bipartite classification: works that use DoCO for internal project goals, and works that discuss its use for modelling document components.

4.4.1. Adoptions of DoCO as part of existing works

Biotea. The Biotea project [17] aims to convert scholarly documents into self-describing machine-readable formats on the basis of several ontologies developed for the publishing domain. As a first step, the authors processed all the XML sources contained in the PubMed Central Open Access Subset and converted them into RDF. DoCO was used to represent textual portions of the paper such as sections, paragraphs, figures and tables, and to link these portions to cited material.

Alighieri's Convivio. Trying to develop mechanisms to represent the knowledge in the notes of Dante Alighieri's essay named *Convivio*, Bartalesi *et al*. [2] described a preliminary study to convert such notes (expressed in XML format) into RDF. Along the same lines as the Biotea project, the authors chose to use several ontologies to model the various aspects involved in the conversion, choosing DoCO to represent portions of the *Convivio*'s structure.

SLOR. In [24] [25], the authors introduce a tool that allows any researcher to create an open repository of research-relevant objects by adding semantic linkages among them according to specific RDF vocabularies and OWL ontologies. This repository, called *Semantic Linkages Open Repository* (*SLOR*), uses DoCO as one of the main ontologies for the description of possible structural and taxonomical relationships between scholarly works.

4.4.2. On the use of DoCO for modelling documents

Reviewing ontologies for scholarly documents. In their work [30], Ruiz-Iniesta and Corcho review several ontologies according to three different contexts: document structure, scientific discourse and citations. As an outcome of such and analysis, they suggest to use DoCO for describing document structures and one of its imported ontologies, DEO, for describing the majority of rhetorical elements.

HuCit. *HuCit* is a light-weight ontology for the description of citation data (with a particular focus on the Humanities). In [29], its authors introduce the classes *BibliographicReferenceList* and *deo:BibliographicReference* as one of the first RDF-based models to describe bibliographic references in scholarly articles.

Mathematical knowledge. In his review article [21], Lange analyses which ontologies could be used to represent mathematical knowledge in form of RDF data. He includes a description of DoCO as a comprehensive way to represent structures and rhetoric of document components of mathematical literature and publications.

ParlBench. *ParlBench* [35] is an RDF benchmark that models digitally-published parliamentary proceedings and related actors, e.g., parliament members and political parties, from the Dutch legislation. DoCO is cited as one of the vocabularies that can be used to describe generic components of parliamentary documents.

5. Conclusions

In this paper we introduced *DoCO*, the *Document Components Ontology*. DoCO is currently one of the most used ontologies for the description of document components and allows one to query, for example, all the bibliographic references cited in "Materials" sections, or to retrieve the sentences containing citations. Its viability as well as its usefulness have been demonstrated through its adoption by different research groups, some of which have been recounted throughout Section 4.

Technically speaking, DoCO is a model that provides a general structured vocabulary of document components on the basis of our previous work on document patterns [13] and other existing works on the rhetorical characterisation of documents, such as [18] [19]. In particular, in this article we formally described the DoCO components that most commonly appear within scientific articles, such as paragraphs, figure, tables, sections, and the like. In addition, we showed tools and methods that use DoCO for different purposes, such as annotating PDF documents or retrieving the intended semantics the document components of scholarly articles.

As future work, starting from the encouraging results we obtained from our tests described in Section 4.3, we plan to refine the heuristics we used in the algorithm so as to increase the precision and recall for each element relative to the gold standard. We plan to extend the set of DoCO structures handled, to enable automated identification of other significant document components such as mathematical formulas, block quotes and front matter metadata (authors, affiliations, e-mail addresses for corresponding authors, etc.).

33 The algorithm (fully-introduced in [14]) is neither an intelligent nor adaptive algorithm, rather a prescriptive one that uses the logical characterisations of DoCO components as a basis to identify them through an iterative process.

34 Balisage Conference Series: http://www.balisage.net – all the data gathered during the test are available at http://www.essepuntato.it/2013/doco/test.

35 We note that this analysis was subjective and solely based on our understanding of the semantics of the element, its definition schema and its documentation.

36 PubMed Central Open Access Subset: http://www.ncbi.nlm.nih.gov/pmc/tools/openftlist/.

37 Science Direct: http://www.sciencedirect.com.

An initial mapping of DoCO with DocBook is already described in [14]. We plan to also extend this mapping and to add additional ones,such as JATS in the near future.

In addition, we are working on extending the current implementation of PDFX in order to identify other document components, including those purely rhetorical (e.g., methods, materials, experiment, data, result, evaluation, discussion), all of which will have adequate DoCO annotations in the XML conversion outputs.

6. Acknowledgements

We would like to thank Angelo Di Iorio and Francesco Poggi for their support and contribution for the development of the structural pattern theory summarised in Section 3.1 and for the algorithm for retrieving the structural characterisation of document components introduced in Section 4.3.

References

[1] Attwood, T. K., Kell, D. B., McDermott, P., Marsh, J., Pettifer, S. R., & Thorne, D. (2010). Utopia documents: linking scholarly literature with research data. Bioinformatics, 26(18): i568–i574. DOI: 10.1093/bioinformatics/btq383

[2] Bartalesi, V., Locuratolo, E., Versienti, L., Meghini, C. (2013). A preliminary study on the semantic representation of the notes to Dante Alighieri's Convivio. In Proceedings of the 2013 Workshop on Collaborative Annotations in Shared Environments: metadata, vocabularies and techniques in the Digital Humanities (DH-CASE 2013). DOI: 10.1145/2517978.2517983

[3] Beck, J. (2010). Report from the Field: PubMed Central, an XML-based Archive of Life Sciences Journal Articles. In Proceedings of the International Symposium on XML for the Long Haul: Issues in the Long-term Preservation of XML. DOI: 10.4242/BalisageVol6.Beck01

[4] Ciccarese, P., Groza, T. (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/ (last visited May 7, 2014)

[5] Ciccarese, P., Shotton, D., Peroni, S., Clark, T. (2014). CiTO+SWAN: The Web Semantics of Bibliographic References, Citations, Evidence and Discourse Relationships. Semantic Web – Interoperability, Usability, Applicability, 5(4): 295–311. DOI: 10.3233/SW-130098

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

[7] Constantin, A. (2014). Automatic Structure and Keyphrase Analysis of Scientific Publications. PhD Thesis. The University of Manchester, UK, 2014.

[8] Constantin, A., Pettifer, S., Voronkov, A. (2013). PDFX: fully-automated PDF-to-XML conversion of scientific literature. In Proceedings of the 2013 ACM symposium on Document Engineering: 177–180. DOI: 10.1145/2494266.2494271

[9] De Waard, A. (2010). From Proteins to Fairytales: Directions in Semantic Publishing. In IEEE Intelligent Systems, 25 (2): 83-88. DOI: 10.1109/MIS.2010.49.

[10] De Waard, A. (2010). Medium-Grained Document Structure. http://www.w3.org/wiki/HCLSIG/SWANSIOC/Actions/RhetoricalStructure/models/medium (last visited May 7, 2014)

[11] Di Iorio, A., Nuzzolese, A. G., Peroni, S., Shotton, D., & Vitali, F. (2014). Describing bibliographic references in RDF. In Proceedings of 4th Workshop on Semantic Publishing, CEUR Workshop Proceedings 1155. http://ceur-ws.org/Vol-1155/paper-05.pdf

[12] Di Iorio, A., Peroni, S., Poggi, F., Vitali, F. (2012). A first approach to the automatic recognition of structural patterns in XML documents. In Proceedings of the 2012 ACM symposium on Document Engineering: 85-94. DOI: 10.1145/2361354.2361374

[13] Di Iorio, A., Peroni, S., Poggi, F., Vitali, F. (2014). Dealing with structural patterns of XML documents. Journal of the American Society for Information Science and Technology, 65(9): 1884–1900. DOI: 10.1002/asi.23088

[14] Di Iorio, A., Peroni, S., Poggi, F., Vitali, F., Shotton, D. (2013). Recognising document components in XML-based academic articles. In Proceedings of the 2013 ACM symposium on Document Engineering: 181–184. DOI: 10.1145/2494266.2494319

[15] Falco, R., Gangemi, A., Peroni, S., & Vitali, F. (2014, in press). Modelling OWL ontologies with Graffoo. To appear in ESWC 2014 Satellite Events - Revised Selected Papers, Lecture Notes in Computer Science. Berlin, Germany: Springer.

[16] Gangemi, A., Peroni, S., Shotton, D., & Vitali, F. (2014, in press). A pattern-based ontology for describing publishing workflows. To appear in Proceedings of the 5th International Workshop on Ontology and Semantic Web Patterns, CEUR Workshop Proceedings.

[17] Garcia Castro, L., McLaughlin, C., Garcia, A. (2013). Biotea: RDFizing PubMed Central in support for the paper as an interface to the Web of Data. Journal of Biomedical Semantics, 4(Suppl 1): S5. DOI: 10.1186/2041-1480-4-S1-S5

[18] Groza, T., Handschuh, S., Moller, K., Decker, S. (2007). SALT - Semantically Annotated LaTeX for Scientific Publications. In Proceedings of the 4th European Semantic Web Conference: 518–532. DOI: 10.1007/978-3-540-72667-8_37

[19] Groza, T., Moller, K., Handschuh, S., Trif, D., Decker, S. (2007). SALT: Weaving the Claim Web. In Proceedings of the 6th International Semantic Web Conference: 197–210. DOI: 10.1007/978-3-540-76298-0_15

[20] Hickson, I. (2011). HTML5: A vocabulary and associated APIs for HTML and XHTML. W3C Candidate Recommendation 29 April 2014. World Wide Web Consortium. http://www.w3.org/TR/html5/ (last visited May 7, 2014)

[21] Lange, C. (2013). Ontologies and languages for representing mathematical knowledge on the Semantic Web. Semantic Web – Interoperability, Usability, Applicability, 4(2): 119–158. DOI: 10.3233/SW-2012-0059

[22] National Information Standards Organization. (2012). JATS: Journal Article Tag Suite. American National

Standard No. ANSI/NISO Z39.96-2012, 9 August 2012. http://www.niso.org/apps/group_public/download.php/10591/z39.96-2012.pdf (last visited May 7, 2014)

[23] Osborne, F., Peroni, S., & Motta, E. (2014, in press). Clustering Citation Distributions for Semantic Categorization and Citation Prediction. To appear in Proceedings of the 4th Workshop on Linked Science (LISC 2014), CEUR Workshop Proceedings.

[24] Parinov, S. (2012). Open Repository of Semantic Linkages. In Proceedings of the 11th International Conference on Current Research Information Systems: 33–42.

[25] Parinov, S., Kogalovsky, M. (2014). Semantic linkages in research information systems as a new data source for scientometric studies. Scientometrics, 98(2): 927–943. DOI: 10.1007/s11192-013-1108-3

[26] Peroni, S., & Shotton, D. (2012). FaBiO and CiTO: Ontologies for describing bibliographic resources and citations. Web Semantics: Science, Services and Agents on the World Wide Web, 17: 33–43. DOI: 10.1016/j.websem.2012.08.001

[27] Peroni, S., Shotton, D., & Vitali, F. (2012). Scholarly publishing and linked data: describing roles, statuses, temporal and contextual extents. In Proceedings of the 8th International Conference on Semantic Systems: 9–16. DOI: 10.1145/2362499.2362502

[28] Pettifer, S., McDermott, P., Marsh, J., Thorne, D., Villeger, A., Attwood, T. K. (2011). Ceci n'est pas un hamburger: modelling and representing the scholarly article. Learned Publishing, 24(3): 207–220. DOI: 10.1087/20110309

[29] Romanello, M., Pasin, M. (2013). Citations and annotations in classics: old problems and new perspectives. In Proceedings of the 2013 Workshop on Collaborative Annotations in Shared Environments: metadata, vocabularies and techniques in the Digital Humanities. DOI: 10.1145/2517978.2517981

[30] Ruiz-Iniesta, A., & Corcho, O. (2014). A review of ontologies for describing scholarly and scientific documents. In Proceedings of 4th Workshop on Semantic Publishing (SePublica 2014), CEUR Workshop Proceedings. Aachen, Germany: CEUR-WS.org. Retrieved from http://ceur-ws.org/Vol-1155/paper-07.pdf

[31] Schneider, J., Groza, T., Passant, A. (2013). A review of argumentation for theSocial Semantic Web. Semantic Web – Interoperability, Usability, Applicability, 4(2): 159–218. DOI: 10.3233/SW-2012-0073

[32] Shotton, D. (2009). Semantic Publishing: the coming revolution in scientific journal publishing. Learned Publishing, 22 (2): 85-94. DOI: 10.1087/2009202.

[33] Shotton, D., Portwin, K., Klyne, G., Miles, A. (2009). Adventures in Semantic Publishing: Exemplar Semantic Enhancements of a Research Article. PLoS Computational Biology, 5 (4): e1000361. DOI: 10.1371/journal.pcbi.1000361.

[34] Tannier, X., Girardot, J., Mathieu, M. (2005). Classifying XML tags through "reading contexts". In Proceedings of the 2005 ACM symposium on Document Engineering: 143–145. DOI: 10.1145/1096601.1096638

[35] Tarasova, T., Marx, M. (2013). ParlBench: A SPARQL Benchmark for Electronic Publishing Applications. In ESWC 2013 Satellite Events - Revised Selected Papers: 5–21. DOI: 10.1007/978-3-642-41242-4 2

[36] Walsh, N. (2010). DocBook 5: The Definitive Guide. Sebastopol, CA, USA: O'Really Media. Version 1.0.3. ISBN: 0596805029

[37] Zou, J., Le, D., & Thoma, G. R. (2007). Structure and Content Analysisfor HTML Medical Articles: A Hidden Markov Model Approach. In Proceedings of the 2007 ACM symposium on Document Engineering: 199–201. DOI: 10.1145/1284420.1284468