Data Science 1 (2017) 155–173
DOI 10.3233/DS-170012
IOS Press

Automating semantic publishing

Silvio Peroni

Digital And Semantic Publishing Laboratory, Department of Computer Science and Engineering, University of Bologna, Bologna, Italy
E-mail: silvio.peroni@unibo.it; ORCID: https://orcid.org/0000-0003-0530-4305

Editor: Michel Dumontier (https://orcid.org/0000-0003-4727-9435)
Solicited reviews: Tobias Kuhn (https://orcid.org/0000-0002-1267-0234); Karin Verspoor (https://orcid.org/0000-0002-8661-1544); Stephen Pettifer (https://orcid.org/0000-0002-1809-5621)

Received 31 May 2017
Accepted 23 August 2017

Abstract. Semantic Publishing involves the use of Web and Semantic Web technologies and standards for the semantic enhancement of a scholarly work so as to improve its discoverability, interactivity, openness and (re-)usability for both humans and machines. Recently, people have suggested that the semantic enhancements of a scholarly work should be undertaken by the authors of that scholarly work, and should be considered as integral parts of the contribution subjected to peer review. However, this requires that the authors should spend additional time and effort adding such semantic annotations, time that they usually do not have available. Thus, the most pragmatic way to facilitate this additional task is to use automated services that create the semantic annotation of authors' scholarly articles by parsing the content that they have already written, thus reducing the additional time required of the authors to that for checking and validating these semantic annotations. In this article, I propose a generic approach called *compositional and iterative semantic enhancement* (CISE) that enables the automatic enhancement of scholarly papers with additional semantic annotations in a way that is independent of the markup used for storing scholarly articles and the natural language used for writing their content.

Keywords: Semantic publishing, compositional and iterative semantic enhancement, CISE, principle of compositionality, downward causation, syntactic containment, structural patterns, structural semantics, rhetorical components

1. Print, digital, and semantic publishing

The scholarly communication domain has been involved in several revolutions concerning the way scientific knowledge has been shared in the past 300 years. Gutenberg's introduction of the print (around 1450) together with the creation of formally-defined groups of scholars (e.g. the Royal Society founded in 1660) have permitted research works to be shared according to a well-known medium, i.e. printed volumes of scholarly journals. Notable examples of this era are Philosophical Transactions published by the Royal Society (first issued in 1776, and still in publication) and The Lancet (first issued in 1823, and currently published by Elsevier). The basic form of the scientific paper remained unchanged until the introduction of the Internet (considering ARPANET as its first implementation around 1960), which enabled research results to be rapidly communicated by means of e-mails. However, it was the advent of the World Wide Web (WWW) in 1989 that made possible the explosion of the Digital Publishing, namely the use of digital formats for the routine publication and distribution of scholarly works. In the last twenty years, the availability of new Web technologies and the reduction of digital storage have resulted in an incredible growth in the availability of scholarly material online, and in an accompanying acceleration of the publishing workflow. However, only with the subsequent advent of one specific set of Web technologies, namely Semantic Web technologies [2], have we started to talk about *Semantic Publishing*.

Within the scholarly domain, Semantic Publishing concerns the use of Web and Semantic Web technologies and standards for enhancing a scholarly work semantically (by means of plain RDF statements [7], nano-publications [17], etc.) so as to improve its discoverability, interactivity, openness and (re-)usability for both humans and machines [35]. The assumptions of openness implicit in Semantic Publishing have been explicitly adopted for the publication of research data by the FAIR (Findable, Accessible, Interoperable, Re-usable) data principles [45]. Early examples of the semantic enrichment of scholarly works involved the use of manual (e.g. [36]) or (semi-)automatic post-publication processes (e.g. [1]) by people other than the original authors of such works.

The misalignment between those who authored the original work and those who added semantic annotations to it is the focal point for discussion by the editors-in-chief of this journal in this introductory issue. Their point is that, following the early experimentation with Semantic Publishing, we should now start to push for and support what they call *Genuine Semantic Publishing*. This *genuineness* basically refers to the fact that the semantic enhancement of a scholarly work should be undertaken by the authors of that scholarly work at the time of writing. According to this perspective, the semantic annotations included within the scholarly work includes should be considered as proper part of the contribution and treated as such, including being subjected to proper peer review.

While the Genuine Semantic Publishing is, indeed, a desirable goal, it requires specific incentives to induce the authors to spend additional time creating the semantic enrichments to their articles, over and above that required to create the textual content. In recent experiments, my colleagues and I have undertaken in the context of the SAVE-SD workshops, described in [31], the clear trend is that, with the exception of the few individuals who *believe* in the Semantic Publishing as a public good, generally only a very low number of semantic statements are specified by the authors, even if we made available incentives or prizes for people who submit their scholarly papers in HTML+RDF format. The number of semantic statements in each of the papers presented during SAVE-SD workshops ranged from 24 to 903, with a median value of 46 (25th percentile 34, 75th percentile 175). The possible reasons for this behaviour, as identified by the study, included the lack of appropriate support, in the form of graphical user interfaces, that would facilitate the annotation of scholarly works with semantic data.

However, I do not think that this is the principal reason that prevents the majority of authors from enhancing their papers with semantic annotations. I firmly believe that the main bottleneck preventing the concrete adoption of Genuine Semantic Publishing principles is *the authors' lack of available time*. While interfaces may simplify the creation of semantic annotations, an author is still required to spend additional time and effort for this task, and often she does not have that time available. Thus, the most pragmatic method of encouraging authors to undertake this additional step is to provide services that do it for them in an *automatic fashion* by parsing the content that the authors have already written, thereby reducing the author's task to one of checking these automated proposals for semantic annotations and then adding them to the document with a few clicks.

In recent years, several tools have been developed for the automatic annotation of scholarly texts according to one or more particular dimensions, e.g. by considering documents that are available in specific document formats – e.g. the SPAR Extractor Suite developed for the RASH format [31] – or that are written in a particular language such as English – e.g. FRED[1] [16]. However, these tools are typically tied to certain requirements – in the aforementioned cases, the use of a specific *markup* for organising the document content and of a particular *language* for writing its text – that prevent their adoption in broader contexts. I wonder if we can propose an alternative approach that allows a machine to infer some of the semantic annotations for scholarly articles **without considering** the particular markup language used nor authoring language used.

This is a Document Engineering issue, rather than a pure Computational Linguistics one. Thus, an important aspect to investigate is whether one can use the syntactic organisation of the text (i.e. the way the various parts of the article are related to each other) to enable the meaningful automatic semantic annotation of the article. It is possible to abstract this view into a more generic research question: "Can the purely syntactic organisation of the various parts of a scholarly article convey something about its semantic and rhetorical representation, and to what extent?"

While I do not have a definite answer to that question, existing theories – such as the *principle of compositionality* [30] and the *downward causation* [3], which I discuss in more detail in Section 3 – seem to suggest that, under certain conditions, they can provide theoretical foundations for that question. Inspired by the way these theories have been used in several domains, I have developed a generic approach called *compositional and iterative semantic enhancement*, a.k.a. *CISE* (pronounced like *size*), that enables the automatic enhancement of scholarly articles solely by consideration of the containment between their components. This requires an iterative process in which each step provides additional semantic annotations to article components by applying specific rules to the enhancements generated by the previous steps.

In order to prove the applicability of CISE, and thus to provide a partial answer to the aforementioned research question, my colleagues and I have implemented and run some CISE-based algorithms, which enable us to annotate various components of scholarly articles with information about their syntactic and semantic structures and basic rhetorical purposes. To do this, we use a collection of ontologies that form a kind of hierarchy that can be used to annotate different aspects a publication – i.e. syntactic containment, syntactic structure, structural semantics, and rhetorical components – with semantic statements. The outcomes of using these CISE-based algorithms are encouraging, and seem to suggest the feasibility of the whole approach.

The rest of the paper is organised as follows. In Section 2 I introduce some of the most important research works on this topic. In Section 3 I introduce the foundational theories that have been used to derive the approach for automating the enhancement of scholarly articles. In Section 4 I introduce CISE by describing the main conditions needed for running it and by explaining the algorithm defining the approach. In Section 5 I briefly discuss the results of implementing CISE on a corpus of scholarly articles stored in XML formats. In Section 6 I present some limitations of the approach, as well as future directions for my research on this subject. Finally, in Section 7 I conclude the paper, reprising the research question mentioned above.

[1] Even if the official website of FRED claims it is able to "parse natural language text in 48 different languages and transform it to linked data", empirical tests indicate that it has been trained appropriately only for English text. In fact, the other languages are not addressed directly, but rather they are handled first by translating non-English text into English (via the Microsoft Translation API) and then by applying the usual FRED workflow for transforming the English translation into Linked Data [16].

2. Related works

Past Document Engineering works that have proposed algorithms for the characterization and identification of particular structural behaviours of various parts of text documents. For instance, in [39], Tannier et al. present an algorithm based on Natural Language Processing (NLP) tools for assigning each element of an XML document to one of three categories. These categories are *hard tag* (i.e. those elements that interrupt the linearity of a text, such as paragraphs and sections), *soft tag* (i.e. the elements that identify significant text fragments that do not break the text flow, such as emphasis and links), and *jump tag* (i.e. those elements that are detached from the surrounding text, such as footnotes and comments).

In another work [46], Zou et al. propose a categorization of HTML elements based on two classes: *inline* (i.e. those that do not provide horizontal breaks in the visualisation of an HTML document) and *line-break* tags (i.e. the opposite of the inline class). They also have developed an algorithm that uses this categorization and a Hidden Markov Model for identifying the structural roles (title, author, affiliation, abstract, etc.) of textual fragments – using a corpus of medical journal articles stored in HTML to provide examples.

The approach proposed by Koh et al. [22] is to identify junk structures in HTML documents, such as navigation menus, advertisements, and footers. In particular, their algorithm recognises recurring hierarchies of nested elements and allows one to exclude all the HTML markup that does not concern the actual content of the document.

Other approaches based on the application of Optical Character Recognition (OCR) techniques to a corpus of PDF documents have also been proposed, with the goal of reconstructing the organisation of a document by marking up its most meaningful parts. For instance, in [21], Kim et al. propose an approach based on the OCR recognition of article zones so as to label them with particular categories, such as affiliations, abstract, sections, titles and authors. Similarly, in [38], Taghva et al. use an OCR technique for reconstructing the logical structure of technical documents starting from the information about fonts and geometry of a scanned document.

Several other works have introduced theories and algorithms for the identification of various characterizations of scholarly articles, such as entities cited in articles (e.g. [13]), rhetorical structures (e.g. [24]), arguments (e.g. [34]), and citation functions (e.g. [8,40] and [19]). In addition, models and ontologies have been proposed for creating and associating annotations to documents and their parts, e.g. [4,28,33], and [25].

These papers thus propose algorithms based on NLP tools, Machine Learning approaches, and OCR frameworks for annotating the various component parts of a document with specific categories. In contrast, the work I present in this article uses none of these techniques. Rather, it involves a pure Document Engineering analysis of the containment relations between the various document parts, without consideration of any aspect related to the language used to write the document or the markup language used to store it. CISE is thus compatible with and complementary to the aforementioned tools, and in principle each could be used to refine the results of the other. However, exploring these kinds of interactions is beyond the scope of this article.

3. A pathway from syntax to semantics

The *principle of compositionality* "states that the meaning of an expression is a function of, and only of, the meanings of its parts together with the method by which those parts are combined" [30]. This definition is quite broad. In fact, it does not precisely define several aspects, such as what is a *meaning*, what is a *function*, and what is a *part* of an expression. Despite its vagueness, this principle has been used in works of several disciplines, such as:

- Linguistics, e.g. Richard Montague, in one of his seminal works [27], proposes an approach that allows the definition of a precise syntax of a natural language (such as English) by means of a set of syntactic categories that are mapped to their possible semantic representations expressed as mathematical formulas by means of explicit conversion rules. In particular, the way the syntactic categories are combined within the syntactic structure of a sentence is used to derive its meaning;
- Computer Science, e.g. the Curry–Howard isomorphism [18], which states that the proof system and the model of computation (such as the lambda calculus) are actually the same mathematical tool presented from two different perspectives. In this light, mathematical proofs can be written as runnable computer programs, and vice versa;
- Molecular Biology, e.g. the reductionist approach used by Crick [6], among others, which claims that the behaviour of high-level functions of a larger biological system (e.g. an organism) can be explained by looking at the ways in which its low-level components (e.g. its genes) actually work.

I propose that the same principle of compositionality can be used to infer high-level semantics from the low-level structural organisation of a scholarly article. The idea is to annotate the various parts of a scholarly article progressively according to diverse layers of annotations, from the lower syntactic layers to higher semantic layers.[2] For instance, a possible stratification of such layers (from the most syntactic ones to the most semantic ones) is illustrated as follows:

1. syntactic containment, i.e. the dominance and containment relations [37] that exist between the various parts of scholarly articles;
2. syntactic structures, i.e. the particular structural pattern (inline, block, etc.) of each part;
3. structural semantics, i.e. the typical article structural types within articles, such as sections, paragraphs, tables, figures;
4. rhetorical components, i.e. the functions that characterize each section, such as introduction, methods, material, data, results, conclusions;
5. citation functions, i.e. the characterization of all inline citations (i.e. in-text reference pointers) with the inferred reason why the authors have made each citation [40];
6. argumentative organisation, i.e. the relations among the various parts of scholarly articles according to particular argumentative models such as Toulmin's [42];
7. article categorization, i.e. the type, either in terms of publication venue (journal article, conference paper, etc.) or content (research paper, review, opinion, etc.), of each scholarly article;
8. discipline clustering, i.e. the identification of the discipline(s) to which each article belongs to.

[2] In the past, my colleagues and I have proposed a separation of the aspects characterizing any unit of scholarly communication (such as an article) into eight different containers we called *semantic lenses* [12]. Each lens is able to describe a particular semantic specification of an article, and it can concern either the description of the article content from different angles (e.g. structure, rhetoric, argumentation of such article), or contextual elements relating to the creation of a paper (e.g. research project, people contributions, publication venue).

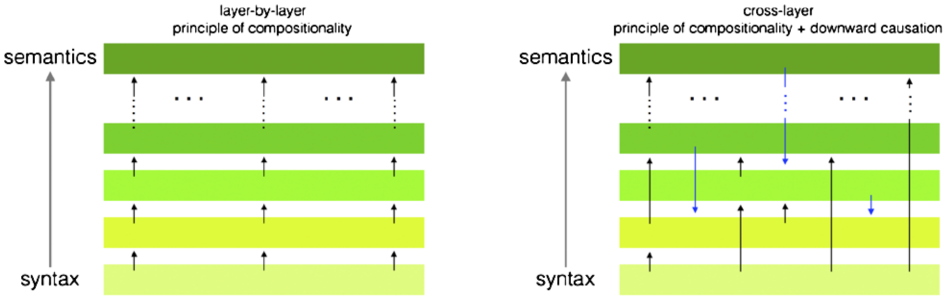

layer-by-layer
principle of compositionality
semantics
syntax
cross-layer
principle of compositionality + downward causation
semantics
syntax

Fig. 1. Two graphs depicting possible uses of the principle of compositionality and of downward causation in the context of scholarly articles. The graph on the left depicts a pure application of the sole principle of compositionality (described by the bottom-to-top black arrows in the figure), where the information in a particular layer is totally derived from the information available in the previous one. In the graph on the right, the information in each layer can be derived by means of the information made available by one or more of the lower layers (principle of compositionality) or by one or more of the higher layers (downward causation, described by the top-to-bottom blue arrows in the figure).

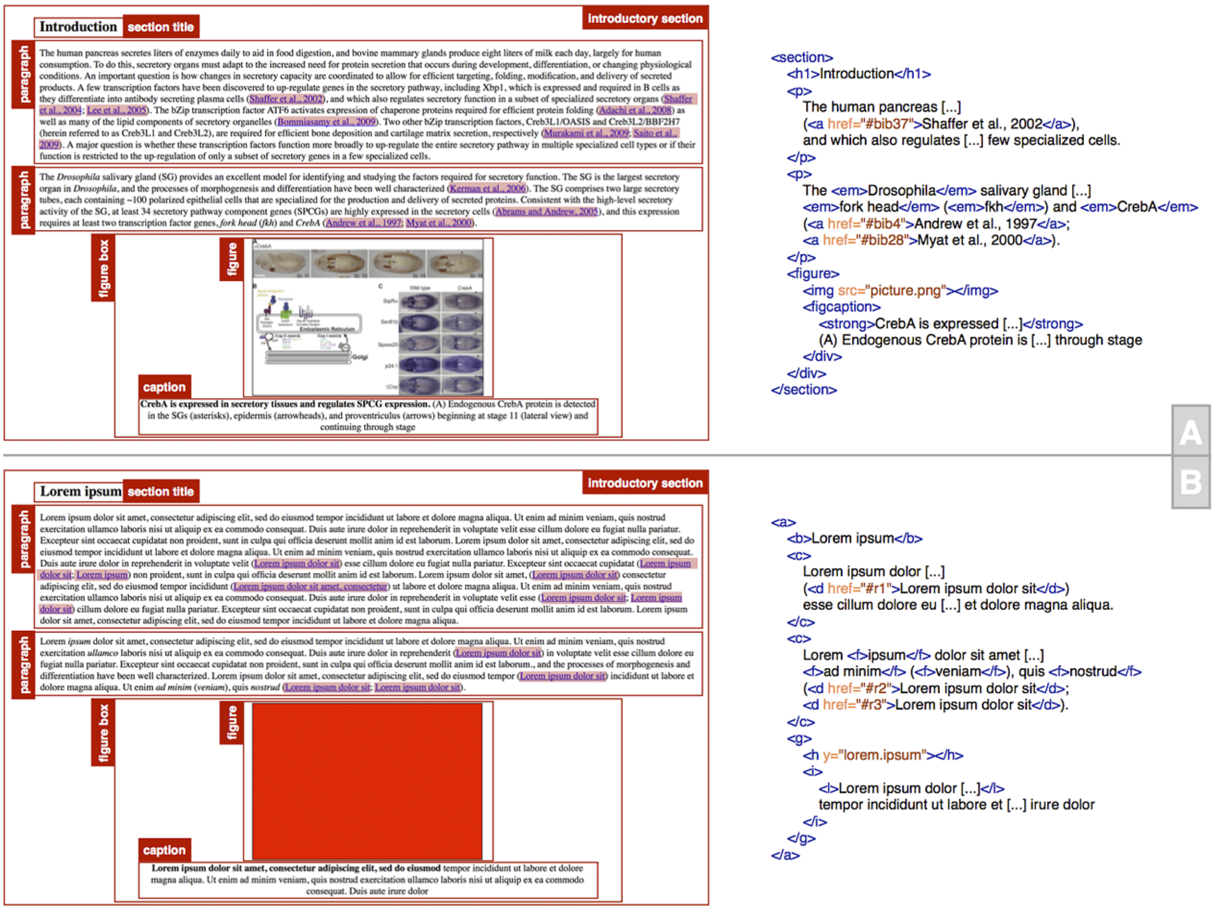
The graph on the left in Fig. 1 shows a strict application of the principle of compositionality. While it may seem valid from an intuitive perspective, past studies (e.g. [29][3]) have proved that the sole application of such principle could not guarantee successful recognition of all the possible interactions existing in a layered system, such as the one that defines the various kinds of meanings possessed by each part of a scholarly article. For instance, it is possible that some semantic annotations of a higher layer can cause the specification of new meanings to a lower layer.

This causal relationship is called *downward causation* [3], and it has been one of the main objections to a purely reductionist approach for the description of the composition and behaviour of a biological system. Downward causation can be defined as a converse of the principle of compositionality: higher layers of a biological system can cause changes to its lower layers. Thus, in the context of the aforementioned eight layers of scholarly articles, it would be possible, for instance, to infer new meanings for the elements annotated in layer 3 by applying the downward causation from layer 4 – e.g. by explicitly marking a section as bibliography of an article (layer 3) if all the child elements that such a section contains (except its title) have been referenced somewhere within the article (layer 4).

The graph on the right in Fig. 1 illustrates the simultaneous use of the principle of compositionality and that of downward causation for inferring the various meanings (at different layers) associated with the parts of a scholarly article. It admits the possibility that a layer can convey meaningful information to any of the higher *or* lower layers. Of course, in order to use the principles of compositionality and downward causation for inferring the characterization of the various parts of a scholarly article, it is important to clarify what are their main components – i.e. the *parts* of an expression, their *meanings*, and the *functions* that enable either the compositionality or downward causation between layers. In the context of scholarly articles, I define these three aspects as follows:

- a *part* of a scholarly article is any content enclosed by a particular marker – for instance, if we consider a scholarly article stored with an XML-like markup language, each markup element defines a particular part of that scholarly article. In the example shown in Fig. 2, the empty boxes with red

[3] In his book [29], Noble claims that, if we regard complexity of a biological system as layered, the more complex layers show emergent properties that cannot be fully explained by the reductionist approach of looking at the simpler layers (e.g. [6]). For example, while the message of DNA (higher layer) is encoded in the four types of nucleotide base (lower layer) that comprise it, it is not determined by them.



Introduction

The human pancreas secretes liters of enzymes daily to aid in food digestion, and bovine mammary glands produce eight liters of milk each day, largely for human consumption. To do this, secretory organs must adapt to the increased need for protein secretion that occurs during development, differentiation, or changing physiological conditions. An important question is how changes in secretory capacity are coordinated to allow for efficient targeting, folding, modification, and delivery of secreted products. A few transcription factors have been discovered to up-regulate genes in the secretory pathway, including Xbp1, which is expressed and required in B cells as they differentiate into antibody secreting plasma cells (Shaffer et al., 2002), and which also regulates secretory function in a subset of specialized secretory organs (Shaffer et al., 2004; Lee et al., 2005). The bZip transcription factor ATF6 activates expression of chaperone proteins required for efficient protein folding (Adachi et al., 2008) as well as many of the lipid components of secretory organelles (Bommiasamy et al., 2009). Two other bZip transcription factors, Creb3L1/OASIS and Creb3L2/BBF2H7 (herein referred to as Creb3L1 and Creb3L2), are required for efficient bone deposition and cartilage matrix secretion, respectively (Murakami et al., 2009; Saito et al., 2009). A major question is whether these transcription factors function more broadly to up-regulate the entire secretory pathway in multiple specialized cell types or if their function is restricted to the up-regulation of only a subset of secretory genes in a few specialized cells.

The *Drosophila* salivary gland (SG) provides an excellent model for identifying and studying the factors required for secretory function. The SG is the largest secretory organ in *Drosophila*, and the processes of morphogenesis and differentiation have been well characterized (Kerman et al., 2006). The SG comprises two large secretory tubes, each containing ~100 polarized epithelial cells that are specialized for the production and delivery of secreted proteins. Consistent with the high-level secretory activity of the SG, at least 34 secretory pathway component genes (SPCGs) are highly expressed in the secretory cells (Abrams and Andrew, 2005), and this expression requires at least two transcription factor genes, *fork head* (*fkh*) and *CrebA* (Andrew et al., 1997; Myat et al., 2000).

CrebA is expressed in secretory tissues and regulates SPCG expression. (A) Endogenous CrebA protein is detected in the SGs (asterisks), epidermis (arrowheads), and proventriculus (arrows) beginning at stage 11 (lateral view) and continuing through stage

```
<section>
   <h1>Introduction</h1>
   <p>
      The human pancreas [...]
      (<a href="#bib37">Shaffer et al., 2002</a>),
      and which also regulates [...] few specialized cells.
   </p>
   <p>
      The <em>Drosophila</em> salivary gland [...]
      <em>fork head</em> (<em>fkh</em>) and <em>CrebA</em>
      (<a href="#bib4">Andrew et al., 1997</a>;
      <a href="#bib28">Myat et al., 2000</a>).
   </p>
   <figure>
      <img src="picture.png"></img>
      <figcaption>
         <strong>CrebA is expressed [...]</strong>
         (A) Endogenous CrebA protein is [...] through stage
      </div>
   </div>
</section>
```

Lorem ipsum

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit (Lorem ipsum dolor sit) esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat (Lorem ipsum dolor sit; Lorem ipsum) non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Lorem ipsum dolor sit amet, (Lorem ipsum dolor sit) consectetur adipiscing elit, sed do eiusmod tempor incididunt (Lorem ipsum dolor sit amet, consectetur) ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse (Lorem ipsum dolor sit; Lorem ipsum dolor sit) cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Lorem *ipsum* dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation *ullamco* laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit (Lorem ipsum dolor sit) in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum., and the processes of morphogenesis and differentiation have been well characterized. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor (Lorem ipsum dolor sit) incididunt ut labore et dolore magna aliqua. Ut enim *ad minim* (*veniam*), quis *nostrud* (Lorem ipsum dolor sit; Lorem ipsum dolor sit).

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor

```
<a>
   <b>Lorem ipsum</b>
   <c>
      Lorem ipsum dolor [...]
      (<d href="#r1">Lorem ipsum dolor sit</d>)
      esse cillum dolore eu [...] et dolore magna aliqua.
   </c>
   <c>
      Lorem <f>ipsum</f> dolor sit amet [...]
      <f>ad minim</f> (<f>veniam</f>), quis <f>nostrud</f>
      (<d href="#r2">Lorem ipsum dolor sit</d>;
      <d href="#r3">Lorem ipsum dolor sit</d>).
   </c>
   <g>
      <h y="lorem.ipsum"></h>
      <i>
         <l>Lorem ipsum dolor [...]</l>
         tempor incididunt ut labore et [...] irure dolor
      </i>
   </g>
</a>
```

Fig. 2. The partial HTML version of a portion of the article entitled "The CrebA/Creb3-like transcription factors are major and direct regulators of secretory capacity" [14] [top panel (A)], and the same containment structure [bottom panel (B)] stored according to a fictional XML-based language, where no meaningful textual or visual content is explicit. The empty boxes with a red border, the blue-underlined strings, as well as the italic and strong strings, describe the various parts of the article. The pink rectangles delimit in-text reference pointers to some bibliographic references, while the meaning of the other parts is defined by means of the red labels with white text.

border, the blue-underlined strings, and the italic and strong strings are delimiting various parts of the scholarly article, represented by specific markup elements in the related XML sources;

- the *meaning* of each part is an informative specification of the type or property characterizing that part. In the example shown in Fig. 2, the red labels with white text, as well as the pink rectangles, define specific semantics of some of the article parts;
- a *function* is an associative rule that allows, given particular known premises, the specification of a meaning to a particular part of a scholarly article.

Using the aforementioned definitions, we can iteratively apply rules following the principles of compositionality and downward causation so as to associate various meanings to article parts. Starting from the pure syntactical containment of certain parts, we can derive their structural semantics, their rhetoric, and other semantic representations of the article. For instance, recalling the stratification into eight layers introduced above, it is possible to create rules that, starting from a low-level definition of the structure of an article (e.g. the organisation of the markup elements that have been used to describe its content – layer 1), permit each markup element to be defined according to more general compositional patterns depicting its structure (e.g. the fact that a markup element can behave like a block or an inline item – layer 2). Again, starting from the definitions in the first two layers, one can go on to characterize the semantics of each markup element according to specific categories defining its structural behaviour (e.g. paragraph, section, list, figure, etc. – layer 3). Along the same lines, one can further derive the rhetorical organisation of a scholarly article, identifying the argumentative role of each part: Introduction, Methods, Material, Experiment, inline reference, etc. (layer 4). In addition, as mentioned above, one can use some of the characterizations specified in layer 4 to specify more precisely the parts annotated in layer 3 (e.g. to identify the bibliography).

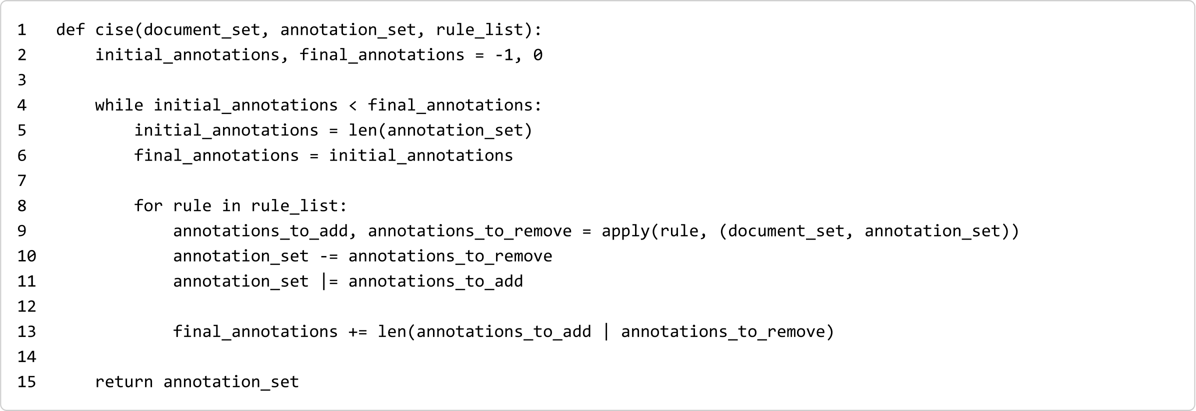
All these associations should be specifiable without considering either the natural language in which the paper is written or the particular markup language in which it is stored. For instance, the two examples depicted in Fig. 2 show that a mechanism developed to infer the semantics of the various parts of the first article (A) should be able to assign the same semantics to parts of the second article (B), since they share the same syntactic containment between article parts.

4. Compositional and iterative semantic enhancement of scholarly articles

Taking inspiration from the ideas introduced in Section 3, and restricting the possible input documents to the scholarly articles available in a reasonable markup language (e.g. an XML-like language), one can propose an approach for retrieving the higher-level characterizations of the parts of a scholarly article starting from their lower-level conceptualisations (by means of the principle of compositionality) and vice versa (by means of the downward causation). I have named this approach *compositional and iterative semantic enhancement* (or *CISE*, pronounced *size*) of scholarly articles. The following conditions should be satisfied when applying CISE:

- [**hierarchical markup**] the source of a scholarly article should be available in (or easily converted into) a markup language that can convey the hierarchical containment of the various parts of scholarly articles;
- [**language agnosticism**] there is no need to have a prior knowledge of the natural language used for writing the scholarly article;
- [**layer inter-dependency**] a layer describing a particular conceptualisation of the parts of a scholarly article should depend on the conceptualisation of at least one other lower or higher layer;
- [**inter-domain reuse**] some of the typical structural and semantic aspects of scholarly articles should be shared across different research domains (e.g. abstract, introduction, conclusions);
- [**intra-domain reuse**] scholarly documents within a specific domain should share several structural and semantic aspects, even if these are not adopted by other research domains (e.g. the "related works" section is used in Computer Science articles, while it is not used in Life Science articles).

The pseudocode shown in Listing 1 introduces the main procedure of CISE. It works by taking three objects as inputs (line 1): a set of marked-up documents to process, a set of annotations referring to the various parts contained in the documents, and a list of rules responsible for inferring new annotations from the existing annotations associated with the documents. Each rule is actually a function taking two parameters as input: a set of documents and a set of annotations. A rule can be run or not run according to the current status of the inputs it receives. For instance, some existing annotations activate the application of a particular rule on a certain document, while others do not. In principle, a rule can simultaneously



```
1   def cise(document_set, annotation_set, rule_list):
2       initial_annotations, final_annotations = -1, 0
3
4       while initial_annotations < final_annotations:
5           initial_annotations = len(annotation_set)
6           final_annotations = initial_annotations
7
8           for rule in rule_list:
9               annotations_to_add, annotations_to_remove = apply(rule, (document_set, annotation_set))
10              annotation_set -= annotations_to_remove
11              annotation_set |= annotations_to_add
12
13              final_annotations += len(annotations_to_add | annotations_to_remove)
14
15      return annotation_set
```

Listing 1. A Python-like pseudocode describing CISE

process more than one document in the input document set, and it could thus find common patterns across documents. The output of a rule is a tuple `to_add, to_remove`, that are two sets of annotations to be added to and removed from the current set of annotations.

Each annotation is a tuple $\mathtt{document}_i$, $\mathtt{layer}_j$, $\mathtt{property}_k$ where:

- $\mathtt{document}_i$ is the document where the annotation has been specified;
- $\mathtt{layer}_j$ is the layer defining the kind of information depicted by the annotation;
- $\mathtt{property}_k$ is a set of statements related to a particular part (i.e. a markup element) of $\mathtt{document}_i$.

After the initialization of some variables (line 2), the main loop of CISE (line 4) continues until no annotations are added or removed from the current set of available annotations. The rationale behind this choice is that the application of the rules can change the status of the specified set of annotations and, consequently, it can create the premises for running a rule that was not previously activated. The following lines (5–6) set the variables that are used for checking if some modifications to the set of annotations are introduced as consequence of an iteration.

The next loop (lines 8–11) is responsible for applying all the rules to all the documents using all the annotations specified as input. As anticipated, the application of a rule returns two sets (line 9): one containing the annotations that should be added, and the other containing the annotations that should be removed. These are then removed from (line 10, where - is the intersection operator between sets) and added to (line 11, where | is the union operator between sets) the current set of annotations. Finally, the variable `final_annotations` is incremented with the number of annotations added to/removed from the current set of annotations (line 13).

When the application of all the rules results in no further modifications to the current set of annotations, the algorithm terminates and returns the updated annotation set (line 15). Otherwise, the algorithm runs a new iteration of the main loop (starting from line 5).

In the following section, I describe the outcomes of some implementations of CISE that I have developed with colleagues in my research group. These outcomes provide the first evidence of the feasibility of CISE for inferring the characterizations of parts of a scholarly article characterized in different layers, each depicting a particular kind of information. These outcomes provide a partial positive answer to the research question introduced in Section 1 – namely whether it is possible to derive the semantic and rhetorical representation of a scholarly article from its syntactic organisation.

5. Implementations of CISE

In recent years, I have experimented extensively, with other colleagues in my research group, with possible paths for the implementation of CISE. Our goal, starting from the pure syntactic containment of the various parts comprising scholarly articles, was to derive additional semantic descriptions of them. Each implementation of CISE we developed aimed at inferring new annotations related to one layer only, describing a specific kind of information.

In our experiments, all the annotations of a layer are defined according to a particular ontology. To this end, we have developed a collection of ontologies that can be used to describe the first four layers introduced in Section 3, namely:

1. syntactic containment: EARMARK [11] provides an ontologically precise definition of markup that instantiates the markup of a text document as an independent OWL document outside of the text strings it annotates;
2. syntactic structures: the Pattern Ontology [9] permits the segmentation of the structure of digital documents into a small number of atomic components, a set of the structural patterns that can be manipulated independently and re-used in different contexts;
3. structural semantics: DoCO, the Document Components Ontology [5] provides a vocabulary for the structural document components (paragraph, section, list, figure, table, etc.); and
4. rhetorical components: DEO, the Discourse Elements Ontology [5] provides a structured vocabulary for rhetorical elements within documents (introduction, discussion, acknowledgements, etc.).

The identification of all the information related to the aforementioned layers is a quite complex work of analysis and derivation. These implementations are a clear evidence that the principles and the approach depicted by CISE are sound, at least to a certain extent, and that the automatic enhancement of scholarly articles by means of Semantic Publishing technologies can be achieved without necessarily using tools that rely on natural language processing or specific markup schemas. In the following subsections, we briefly introduce the outcomes of our experimentations with CISE.[4]

5.1. From containment to structural patterns

Understanding how scholarly documents can be segmented into structural components, which can then be manipulated independently for different purposes, is a topic that my colleagues and I have studied extensively in the past. The main outcome of this research [9] is the proposal of a theory of structural patterns for digital documents that are sufficient to express what most users need in terms of document constituents and components when writing scholarly papers.

The basic idea behind this theory – which has been derived by analysing best practice in existing XML grammars and documents [43] – is that each element of a markup language should comply with one and only one structural pattern, depending on the fact that the element:

- can or cannot contain text (+t in the first case, −t otherwise);
- can or cannot contain other elements (+s in the first case, −s otherwise);
- is contained by another element that can or cannot contain text (+T in the first case, −T otherwise).

[4] In the following section, only a brief introduction of the various implementations of CISE is provided, since I prefer to focus on the outcomes obtained by using such implementations by presenting some meaningful examples. Additional details about the theoretical foundations of these implementations and the precise explanation of all the algorithms can be found in [9] and [10].

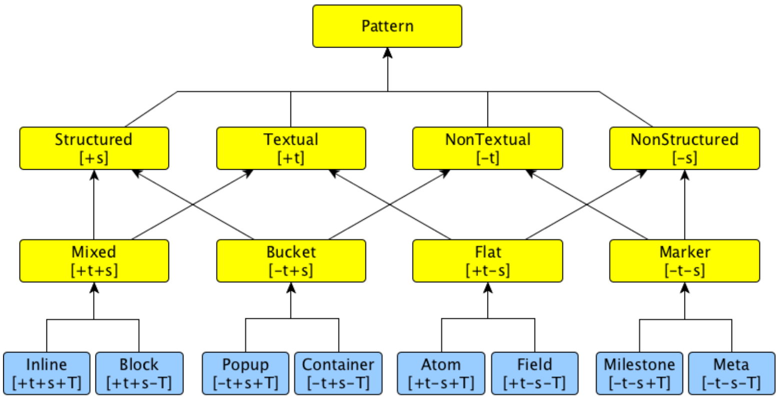


Fig. 3. The taxonomical relations between the classes defined in the Pattern Ontology. The arrows indicate sub-class relationships between patterns (e.g. Mixed is sub-class of Structured), while the values ±t, ±s, and ±T between square brackets indicate the compliance of each class to the theory of patterns introduced in [9]. In particular, the top yellow classes define generic properties that markup elements may have, while the bottom light-blue classes define the eight patterns identified by our theory. Note that no Block- and Inline-based elements can be used as root elements of a pattern-based document.

```
1   def implementation1(xml_documents)
2       layer1 = cise(xml_documents, set(), [ convert_into_earmark ])
3
4       patterns = cise(xml_documents, layer1, [
5           assign_t_s_properties,
6           assign_T_properties,
7           assign_patterns])
8
9       layer2 = cise(xml_documents, patterns, [
10          reach_local_coherence,
11          reach_global_coherence])
12
13      return layer2
```

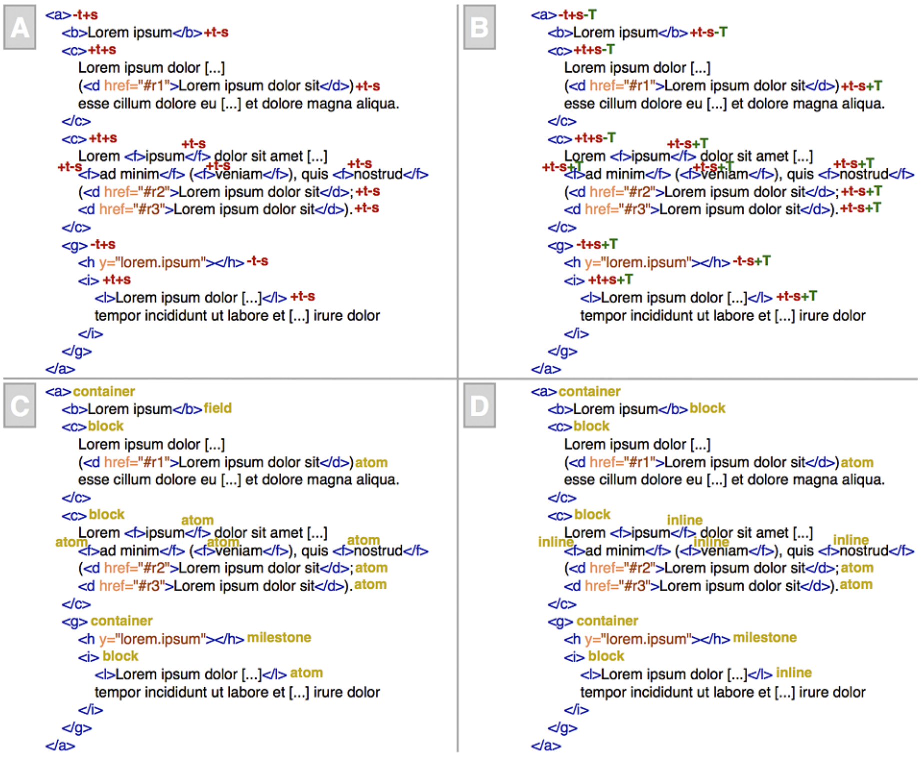
Listing 2. A Python-like pseudocode reusing the code introduced in Listing 1 for creating the annotations related to the first two layers introduced in Section 5

By combining all these possible values – ±t, ±s, and ±T – we obtain eight core structural patterns: inline, block, popup, container, atom, field, milestone, and meta. These patterns are described in the Pattern Ontology and are summarised in Fig. 3.

In [9], colleagues and I have experimented with a CISE implementation for assigning structural patterns to markup elements in XML sources, without relying on any background information about the vocabulary, its intended meaning, its schema, and the natural language in which they have been written. The main steps of the algorithm implemented are shown in Listing 2.

This new algorithm is organised in three main steps, each re-using the mechanism introduced in Listing 1. The first step (line 2) retrieves all the annotations referring to the first layer (i.e. syntactic containment) by representing the XML input documents in input using EARMARK [11]. This transformation is handled by the function `convert_into_earmark` which populates the initially empty set of annotations with statements that guarantee a complete ontological description of the document markup.

The second step (lines 4–7) assigns, to each instance of each element of each document, the ±t and ±s values according to the kinds of nodes (i.e. textual nodes and/or markup elements) such instance



A

```
<a>-t+s
  <b>Lorem ipsum</b>+t-s
  <c>+t+s
    Lorem ipsum dolor [...]
    (<d href="#r1">Lorem ipsum dolor sit</d>)+t-s
    esse cillum dolore eu [...] et dolore magna aliqua.
  </c>
  <c>+t+s
                    +t-s
    Lorem <f>ipsum</f> dolor sit amet [...]
                            +t-s                +t-s
+t-s<f>ad minim</f> (<f>veniam</f>), quis <f>nostrud</f>
    (<d href="#r2">Lorem ipsum dolor sit</d>;+t-s
    <d href="#r3">Lorem ipsum dolor sit</d>).+t-s
  </c>
  <g>-t+s
    <h y="lorem.ipsum"></h>-t-s
    <i>+t+s
      <l>Lorem ipsum dolor [...]</l>+t-s
      tempor incididunt ut labore et [...] irure dolor
    </i>
  </g>
</a>
```

B

```
<a>-t+s-T
  <b>Lorem ipsum</b>+t-s-T
  <c>+t+s-T
    Lorem ipsum dolor [...]
    (<d href="#r1">Lorem ipsum dolor sit</d>)+t-s+T
    esse cillum dolore eu [...] et dolore magna aliqua.
  </c>
  <c>+t+s-T
                    +t-s+T
    Lorem <f>ipsum</f> dolor sit amet [...]
                            +t-s+T              +t-s+T
+t-s+T<f>ad minim</f> (<f>veniam</f>), quis <f>nostrud</f>
    (<d href="#r2">Lorem ipsum dolor sit</d>;+t-s+T
    <d href="#r3">Lorem ipsum dolor sit</d>).+t-s+T
  </c>
  <g>-t+s+T
    <h y="lorem.ipsum"></h>-t-s+T
    <i>+t+s+T
      <l>Lorem ipsum dolor [...]</l>+t-s+T
      tempor incididunt ut labore et [...] irure dolor
    </i>
  </g>
</a>
```

C

```
<a>container
  <b>Lorem ipsum</b>field
  <c>block
    Lorem ipsum dolor [...]
    (<d href="#r1">Lorem ipsum dolor sit</d>)atom
    esse cillum dolore eu [...] et dolore magna aliqua.
  </c>
  <c>block
                    atom
    Lorem <f>ipsum</f> dolor sit amet [...]
                            atom                atom
atom<f>ad minim</f> (<f>veniam</f>), quis <f>nostrud</f>
    (<d href="#r2">Lorem ipsum dolor sit</d>;atom
    <d href="#r3">Lorem ipsum dolor sit</d>).atom
  </c>
  <g>container
    <h y="lorem.ipsum"></h>milestone
    <i>block
      <l>Lorem ipsum dolor [...]</l>atom
      tempor incididunt ut labore et [...] irure dolor
    </i>
  </g>
</a>
```

D

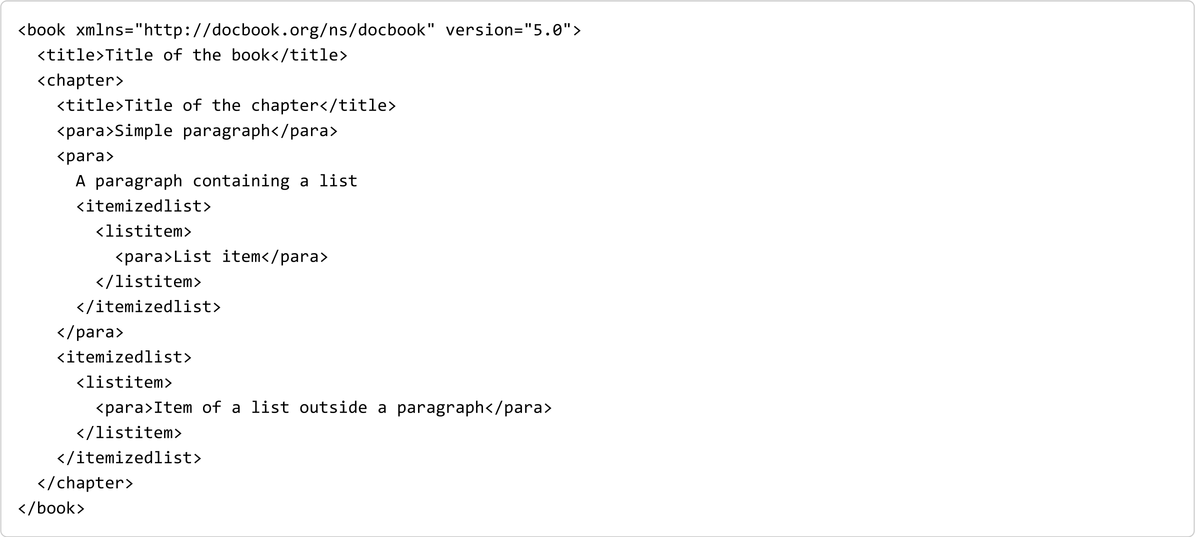
```
<a>container
  <b>Lorem ipsum</b>block
  <c>block
    Lorem ipsum dolor [...]
    (<d href="#r1">Lorem ipsum dolor sit</d>)atom
    esse cillum dolore eu [...] et dolore magna aliqua.
  </c>
  <c>block
                    inline
    Lorem <f>ipsum</f> dolor sit amet [...]
                            inline              inline
inline<f>ad minim</f> (<f>veniam</f>), quis <f>nostrud</f>
    (<d href="#r2">Lorem ipsum dolor sit</d>;atom
    <d href="#r3">Lorem ipsum dolor sit</d>).atom
  </c>
  <g>container
    <h y="lorem.ipsum"></h>milestone
    <i>block
      <l>Lorem ipsum dolor [...]</l>inline
      tempor incididunt ut labore et [...] irure dolor
    </i>
  </g>
</a>
```

Fig. 4. Four snapshots of the same excerpt of an XML document, describing the execution of the algorithm introduced in Fig. 4.

contains (function assign_t_s_properties), and the ±T values according to the previous assignments (function assign_T_properties). Finally, the full structural pattern is specified using all the aforementioned assignments (function assign_patterns).

The third step (lines 9–11) is responsible for harmonising the pattern association of all the instances of the same element. For instance, it would be possible that, in the same document, two instances of an element “x” are assigned to two different patterns, e.g. *atom* and *inline*. However, occasionally, it would be possible to generalise these assignments so as to have the same pattern specified for all the instances of the same element – e.g. in the previous example, all the instances of “x” assigned to the pattern *atom* can be safely assigned to the pattern *inline* since it is more flexible, allowing the containment of both text nodes and markup elements. This operation of harmonisation, applied to all the pattern assignments of a document, aims at reaching the *local coherence* of the assignments (function reach_local_coherence) – i.e. the situation in which all the instances of each markup element in a document are assigned to the same pattern. It is worth mentioning that there are situations where such local coherence cannot be reached for some documents. In these cases, all their pattern assignments are removed, so as not to be processed by the following rules. In addition, this harmonisation operation can be applied to all the instances of all the markup elements contained in all the documents that have been found locally coherent. In this case, we talk about reaching the *global coherence* of the pattern assignments across the corpus of documents (function reach_global_coherence). Finally, the results of these associations are returned (line 13).

In Fig. 4, I show an execution of the algorithm in Listing 2 using the document introduced in the panel B of Fig. 2. In particular, panel A of Fig. 4 shows the outcomes of the function assign_



```
<book xmlns="http://docbook.org/ns/docbook" version="5.0">
  <title>Title of the book</title>
  <chapter>
    <title>Title of the chapter</title>
    <para>Simple paragraph</para>
    <para>
      A paragraph containing a list
      <itemizedlist>
        <listitem>
          <para>List item</para>
        </listitem>
      </itemizedlist>
    </para>
    <itemizedlist>
      <listitem>
        <para>Item of a list outside a paragraph</para>
      </listitem>
    </itemizedlist>
  </chapter>
</book>
```

Listing 3. An example of a DocBook document that is not pattern-based. The only possible configuration is to associate the Inline pattern with all the elements except `book`, that must be associated with the pattern Block. However, as explained in Fig. 3, a Block-based element cannot be the root element of a pattern-based XML document

`t_s_properties`, supposing that the containment relations of the elements have been already described by the annotations included in layer 1. Panel B completes the previous assignments with those introduced by the function `assign_T_properties`. Panel C depicts only the outcomes of the function `assign_patterns` (without showing again the previous assignments). Finally, panel D shows some modifications of the pattern assignments to reach local and global coherence (functions `reach_local_coherence` and `reach_global_coherence`).

In order to understand what extent structural patterns are used in different communities, we have executed this CISE-based algorithm on a large set of documents stored using different XML-based markup languages. Some of these markup languages, e.g. TEI [41] and DocBook [44], are not inherently pattern-based – i.e. it is possible in principle to use them to write locally/globally incoherent documents. For instance, a DocBook document that is not pattern-based in introduced in Listing 3.

This experimentation allowed us to reached the following conclusions:

- in a community (e.g., a conference proceedings or a journal) that uses a permissive non-pattern-based markup language, most authors nevertheless use a pattern-based subset of such language for writing their scholarly articles. This conclusion has been derived by analysing the outcomes of the algorithm execution, as illustrated in [9];
- only a small number of pattern-based documents coming from different communities of authors, all stored using the same markup language, is needed for automatically generating generic visualisations for all documents stored in that markup language (regardless of whether they are pattern-based or not) included in the communities in consideration. This conclusion has been empirically demonstrated by developing a prototypical tool, called PViewer, which implements such automatic visualisation mechanism, as introduced in [9].

Once the algorithm has identified that all the individuals of the markup element “x” included in pattern-based documents of a certain community comply with a particular pattern, it is then possible implicitly to associate the same pattern to all the individual of the same element included in non-pattern-based documents of the same community. As a consequence, these assignments can be used to provide a guide to authors (or even to automatic tools) for adjusting the current organisation of non-pattern-based documents so as to convert them into proper pattern-based ones.

5.2. From structural patterns to structural semantics

The systematic use of the patterns introduced in Section 5.1 allows authors to create unambiguous, manageable and well-structured documents. In addition, thanks to the regularity they provide, it is possible to perform complex operations on pattern-based documents (e.g. visualisation) even without knowing their vocabulary. Thus, it should be possible to implement reliable and efficient tools and algorithms that can make hypotheses regarding the meanings of document fragments, that can identify singularities, and that can study global properties of sets of documents.

Starting from these premises, my colleagues and I have implemented another algorithm to infer layer 3 annotations from the structural patterns retrieved using the algorithm introduced in Listing 2 [10]. Specifically, we use the entities defined in the Document Components Ontology (DoCO) (excluding, on purpose, those defined in other imported ontologies) to mark up elements of documents that have specific structural connotations, such as paragraphs, sections, titles, lists, figures, tables, etc.

A pseudocode of this new algorithm is introduced in Listing 4. It is organised in two steps. First, starting from the input XML documents, it retrieves all the annotations of the first two layers by reusing the algorithm discussed in Section 5.1 (line 2). Then, the following step (lines 4–9) reuses the CISE algorithm introduced in Section 4, by specifying the annotation retrieved previously and the list of functions, where each is responsible for identifying all the markup elements that act according to a specific structural behaviour (e.g. paragraph) in the documents. Each of these rules implements some heuristics that have been derived by analysing how the structural patterns are used in scholarly documents, as detailed in [10] with more detail. For instance, `identify_paragraphs` associate the type `doco:Paragraph` to all the instances of the same markup element “x” that is the block element (in terms of structural patterns) with most occurrences (i.e. instances) in the document. If we consider the example in the panel D of Fig. 4, all the instances of the element “c” will be annotated as paragraphs.

We have run the algorithm introduced in Listing 4 on the XML sources of all the articles published in the Proceedings of Balisage (http://balisage.net), which are marked up in DocBook [44]. We obtained quite high overall values of precision and recall (i.e. 0.887 and 0.890 respectively) when comparing the

```
1   def implementation2(xml_documents):
2       layer2 = implementation1(xml_documents)
3
4       layer3 = cise(xml_documents, layer2, [
5           identify_paragraphs,
6           identify_sections,
7           identify_section_titles,
8           identify_body_matter,
9           ...])
10
11      return layer3
```

Listing 4. A Python-like pseudocode reusing the code introduced in Listing 1 and Listing 2, for adding the annotations related to the third layer

results of the algorithm with a gold standard we created manually by assigning structural characterizations to all the markup elements defined in DocBook. By analysing the outcomes of the algorithm execution, as illustrated in [10], we have reached the following conclusion:

- only a small number of pattern-based documents, written by different authors of the same community, is needed for extracting the structural semantics of the main components of all the documents produced by that community.

5.3. *From structural semantics to rhetorical components and back*

The work presented in [10] was further extended by applying two additional algorithms that have allowed us to retrieve specific rhetorical components in the document, i.e. the *references*. We then used this rhetorical characterization to assign more precise definitions to the structural entities retrieved by the algorithm introduced in Listing 4.

According to the Discourse Element Ontology (DEO), the ontology used to represent the annotations of layer 4, a *reference* is an element that references to a specific part of the document or to another publication. Thus, this category describes any bibliographic reference, any in-text reference pointer to a bibliographic reference, and any pointer to other article items such as figures or tables. These references are recognised by means of the algorithm introduced in Listing 5.

This new algorithm is organised in two steps. First, starting from the input XML documents, it retrieves all the annotations of the first three layers by reusing the algorithm discussed in Section 5.2 (line 2). Then, the following step (line 4) reuses the CISE algorithm introduced in Section 4, by specifying the annotations retrieved previously and a particular function, `identify_references`, that is responsible for annotating all the markup elements that act as references. In particular, this function associates the type `deo:Reference` to all the instances of elements that are compliant either with the pattern *atom* or the pattern *milestone*, and that have an attribute "x" with value "#" + "v", where "v" is the value of another attribute "y" of another element.

Although this rhetorical characterization of article parts only identifies references, these new annotations allow us to infer new meanings for the elements annotated in layer 3, by applying downward causation from layer 4. Specifically, this information about the references allowed us to identify the Bibliography section of an article, by identifying the element of type `doco:Section` (layer 3) whose children elements, except the section title, are all referenced in the main text.

The algorithm in Listing 6 implements this: after retrieving all fourth layer annotations (line 2), it reuses the CISE algorithm introduced in Section 4 so as to annotate all the lists that are bibliographic reference lists (function `identify_bibliographic_reference_lists`) and all the sections that are Bibliography sections (function `identify_bibliographies`).

```
1   def implementation3(xml_documents):
2       layer3 = implementation2(xml_documents)
3
4       layer4 = cise(xml_documents, layer3, [ identify_references ])
5
6       return layer4
```

Listing 5. A Python-like pseudocode reusing the code introduced in Listing 1 and Listing 4 for adding the annotations related to the fourth layer

```
def implementation4(xml_documents):
    layer4 = implementation3(xml_documents)

    layer3 = cise(xml_documents, layer4, [
        identify_bibliographic_reference_lists,
        identify_bibliographies])

    return layer3
```

Listing 6. A Python-like pseudocode reusing the code introduced in Listing 1 and Listing 5, that adds annotations related to the third layer using downward causation

6. Limitations and future directions

In Section 5 I have shown how the approach proposed in Section 4 can be implemented to return annotations belonging the first four layers introduced in Section 3. It also presents some issues that cannot be addressed by looking only at the syntactical containment of article parts. The most important issue is this: since CISE focusses on article parts that must be clearly marked-up in some way, all the other portions of the article text are simply discarded. By design, CISE does not undertake the analysis of natural language text. There are several existing applications, such as Named Entity Recognition tools (NER) [15] and other NLP technologies (e.g. [34]), that can be used for this purpose.

CISE is intrinsically limited by some of the conditions, introduced in Section 4, necessary for its implementation. The most limiting is the requirement that the article to be analysed should be appropriately marked up by means of some (e.g. XML-based) markup language. While it is not necessary to know the particular grammar and vocabulary of the markup language used, it is crucial that the article parts are organised hierarchically according to appropriate containment principles. For instance, in an HTML-based article, all the section-subsection relations should be explicitly defined by means of nested `section` elements. The common usage of headings of different levels (i.e. `h1`–`h6` elements) within a flat structure (e.g. an `article` element) is inadequate for this, since it does not explicitly define the intended hierarchical organisation of the sections.[5]

CISE depends on the theoretical backgrounds introduced in Section 3, namely the principles of compositionality and downward causation. The other conditions introduced in Section 4 are less strict and need a more careful investigation. In particular, while scholarly documents can be written in different natural languages, their main constituents common across the whole scholarly communication domain. Of course, the presentation of research articles can vary according to the particular domain – e.g. Life Sciences articles usually follow a well-defined structure, while Computer Science articles are usually organised in a less conservative way. However, they all follow generic underlying ways of organising the arguments supporting the research conclusions – even if some aspects are more commonly found in one domain rather than another (e.g. Life Science articles typically have a Background section, while Computer Science articles have a Related Works section, whose function, while similar, is different in important aspects). Nevertheless, my hypothesis (partially supported by existing studies, e.g. [26]) is that many of the ways of organising research articles are shared across the various research areas.

Future extensions and new implementations of CISE will permit me to study such intra- and inter-domain patterns and similarities, identifying which particular structures and semantic connotations are

[5] In the particular example introduced, it would be possible, in principle, to reconstruct automatically the section-subsection containment by looking at the headings included in such a flat structure. This flat way of organising sections is adopted in existing markup languages for textual documents, such as LaTeX and ODT [20].

shared between different research areas, and determining how much argumentative information is hidden behind quite simple syntactic structures. From such analysis, it might be possible to identify particular patterns of organisation of document parts that characterize certain domains or particular the types of papers (research papers, reviews, letters, etc.).

It might also be possible to extend CISE to the remaining higher layers introduced in Section 3, each defined by a specific ontology: annotations about citation functions (target ontology: CiTO [32]), argumentative organisation (target ontology: AMO), article categorization (target ontology: FaBiO [32]), and discipline clustering (target ontology: DBpedia Ontology [23]).

7. Conclusions

In this article, I have introduced the *compositional and iterative semantic enhancement* (CISE) approach for the automated enrichment of scholarly articles with meaningful semantic annotations by means of Semantic Publishing technologies. Taking inspiration from past approaches that rely on the principles of compositionality and downward causation, the CISE strategy enables the automatic enhancement of the various parts comprising a scholarly article by means of an iterative process, providing additional semantic descriptions by combining the enhancements obtained in previous executions of the approach.

I have also discussed the outcomes of past CISE experimentations that my colleagues and I have developed for the automatic annotation of document components in scholarly articles according to particular structural patterns (inline, block, etc.), structural semantics (paragraph, section, figure, etc.), and rhetorical components (i.e. references), as introduced in Section 5. While these do not address all the layers of annotations sketched in Section 3, I think the outcomes described in Section 5 are acceptable pointers for claiming that some level of automatic semantic enhancement of scholarly articles is possible even in the presence of specific constraints such as the independence from the natural language used for writing such articles and the markup language used for storing their contents. Thus CISE provides at least a partial positive answer to the research question presented in Section 1: “Can the purely syntactic organisation of the various parts of a scholarly article convey something about its semantic and rhetorical representation, and to what extent?”. In the future, I plan to perform additional studies and experiments to further validate this claim.

Acknowledgements

I would like to thank all the colleagues working at the Digital and Semantic Publishing Laboratory (DASPLab) – namely Angelo Di Iorio, Francesco Poggi, and Fabio Vitali – who have shared with me their time and effort on these topics. I would also like to thank the Editors-in-Chief and the reviewers of the *Data Science Journal* for their comments and suggestions – they have been fundamental to the improvement of the quality and the narrative of this article. Last but not least, two big thanks to my Semantic Publishing mentor and fellow, David Shotton. Some years ago, he gave me a wonderful book, “*The Music of Life: Biology Beyond Genes*” written by Denis Noble, that has been the primary source of inspiration for CISE. In addition, David has carefully proofread of the whole text, improving drastically the readability of this article.

References

[1] A. Bagnacani, P. Ciancarini, A. Di Iorio, A.G. Nuzzolese, S. Peroni and F. Vitali, The semantic lancet project: A linked open dataset for scholarly publishing, in: *EKAW (Satellite Events) 2014*, 2014, pp. 101–105. doi:10.1007/978-3-319-17966-7_10.

[2] T. Berners-Lee, J. Hendler and O. Lassila, The semantic web, *Scientific American* **285**(5) (2001), 34–43. doi:10.1038/scientificamerican0501-34.

[3] D.T. Campbell, Downward causation in hierarchically organised biological systems, in: *Studies in the Philosophy of Biology*, F.J. Ayala and T. Dobzhansky, eds, 1974, pp. 179–186. doi:10.1007/978-1-349-01892-5_11.

[4] D.C. Comeau, R. Islamaj Doğan, P. Ciccarese, K.B. Cohen, M. Krallinger, F. Leitner, Z. Lu, Y. Peng, F. Rinaldi, M. Torii, A. Valencia, K. Verspoor, T.C. Wiegers, C.H. Wu and W.J. Wilbur, BioC: A minimalist approach to interoperability for biomedical text processing, *Database* **2013** (2013), bat064. doi:10.1093/database/bat064.

[5] A. Constantin, S. Peroni, S. Pettifer, D.M. Shotton and F. Vitali, The Document Components Ontology (DoCO), *Semantic Web* **7**(2) (2016), 167–181. doi:10.3233/SW-150177.

[6] F. Crick, *Of Molecules and Men*, University of Washington Press, 1966, ISBN 1591021855, https://philpapers.org/rec/CRIOMA.

[7] R. Cyganiak, D. Wood and M. Lanthaler, RDF 1.1 concepts and abstract syntax, W3C Recommendation 25 February 2014, 2014, https://www.w3.org/TR/rdf11-concepts/.

[8] A. Di Iorio, A.G. Nuzzolese and S. Peroni, Characterizing citations in scholarly documents: The CiTalO framework, in: *ESWC 2013 Satellite Events – Revised Selected Papers*, 2013, pp. 66–77. doi:10.1007/978-3-642-41242-4_6.

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

[10] A. Di Iorio, S. Peroni, F. Poggi, F. Vitali and D. Shotton, Recognising document components in XML-based academic articles, in: *Proceedings of the 2013 ACM Symposium on Document Engineering (DocEng 2013)*, 2013, pp. 181–184. doi:10.1145/2494266.2494319.

[11] A. Di Iorio, S. Peroni and F. Vitali, A semantic web approach to everyday overlapping markup, *Journal of the American Society for Information Science and Technologies* **62**(9) (2011), 1696–1716. doi:10.1002/asi.21591.

[12] A. Di Iorio, S. Peroni, F. Vitali and J. Zingoni, Semantic lenses to bring digital and semantic publishing together, in: *Proceedings of the 4th Workshop on Linked Science 2014 (LISC2014)*, 2014, pp. 12–23, http://ceur-ws.org/Vol-1282/lisc2014_submission_6.pdf.

[13] J.L. Fink, P. Fernicola, R. Chandran, S. Parastatidis, A. Wade, O. Naim, G.B. Quinn and P.E. Bourne, Word add-in for ontology recognition: Semantic enrichment of scientific literature, *BMC Bioinformatics* **2010**(11) (2010), 103. doi:10.1186/1471-2105-11-103.

[14] R.M. Fox, C.D. Hanlon and D.J. Andrew, The CrebA/Creb3-like transcription factors are major and direct regulators of secretory capacity, *Journal of Cell Biology* **191**(3) (2010), 479–492. doi:10.1083/jcb.201004062.

[15] A. Gangemi, A comparison of knowledge extraction tools for the semantic web, in: *Proceedings of the 10th Extended Semantic Web Conference (ESWC 2013)*, 2013, pp. 351–366. doi:10.1007/978-3-642-38288-8_24.

[16] A. Gangemi, V. Presutti, D.R. Recupero, A.G. Nuzzolese, F. Draicchio and M. Mongiovì, Semantic web machine reading with FRED, *Semantic Web* **8**(6) (2017). doi:10.3233/SW-160240.

[17] P. Groth, A. Gibson and J. Velterop, The anatomy of a nanopublication, *Information Services and Use* **30**(1–2) (2010), 51–56. doi:10.3233/ISU-2010-0613.

[18] W.A. Howard, The formulae-as-types notion of construction, in: *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, J.P. Seldin and J.R. Hindley, eds, Academic Press, Boston, MA, 1980, pp. 479–490, ISBN 978-0-12-349050-6, http://www.dcc.fc.up.pt/~acm/howard.pdf (last visited May 30, 2017).

[19] R. Jha, A.-A. Jbara, V. Qazvinian and D.R. Radev, NLP-driven citation analysis for scientometrics, *Natural Language Engineering* **23**(1) (2017), 93–130. doi:10.1017/S1351324915000443.

[20] JTC1/SC34 WG 6, ISO/IEC 26300:2006 – information technology – open document format for office applications (OpenDocument) v1.0, International Organization for Standardization, Geneva, Switzerland, 2006, http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=43485.

[21] J. Kim, D.X. Le and G.R. Thoma, Automated labeling in document images, in: *Proceedings of Document Recognition and Retrieval VIII*, 2000, pp. 111–122. doi:10.1117/12.410828.

[22] E. Koh, D. Caruso, A. Kerne and R. Gutierrez-Osuna, Elimination of junk document surrogate candidates through pattern recognition, in: *Proceedings of the 2007 ACM Symposium on Document Engineering (DocEng 2007)*, 2007, pp. 187–195. doi:10.1145/1284420.1284466.

[23] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P.N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer and C. Bizer, DBpedia – a large-scale, multilingual knowledge base extracted from Wikipedia, *Semantic Web* **6**(2) (2015), 167–195. doi:10.3233/SW-140134.

[24] M. Liakata, S. Teufel, A. Siddharthan and C. Batchelor, in: *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC 2010)*, 2010, pp. 2054–2061, http://www.lrec-conf.org/proceedings/lrec2010/pdf/644_Paper.pdf.

[25] K.M. Livingston, M. Bada, L.E. Hunter and K. Verspoor, Representing annotation compositionality and provenance for the semantic web, *Journal of Biomedical Semantics* **2013**(4) (2013), 38. doi:10.1186/2041-1480-4-38.

[26] S. Maswana, T. Kanamaru and A. Tajino, Move analysis of research articles across five engineering fields: What they share and what they do not, *Ampersand* **2** (2015), 1–11. doi:10.1016/j.amper.2014.12.002.

[27] R. Montague, Universal grammar, *Theoria* **36**(3) (1970), 373–398. doi:10.1111/j.1755-2567.1970.tb00434.x.

[28] L. Moreau and P. Missier, PROV-DM: The PROV Data Model, W3C Recommendation 30 April 2013, 2013, http://www.w3.org/TR/prov-dm/.

[29] D. Noble, *The Music of Life: Biology Beyond Genes*, Oxford University Press, 2008, ISBN 9780199228362, https://global.oup.com/academic/product/the-music-of-life-9780199228362.

[30] F.J. Pelletier, The principle of semantic compositionality, *Topoi* **13**(1) (1994), 11–24. doi:10.1007/BF00763644.

[31] S. Peroni, F. Osborne, A. Di Iorio, A.G. Nuzzolese, F. Poggi, F. Vitali and E. Motta, Research articles in simplified HTML: A web-first format for HTML-based scholarly articles, *PeerJ Computer Science* **3** (2017), e132. doi:10.7717/peerj-cs.132.

[32] S. Peroni and D. Shotton, FaBiO and CiTO: Ontologies for describing bibliographic resources and citations, *Web Semantics* **17** (2012), 33–43. doi:10.1016/j.websem.2012.08.001.

[33] R. Sanderson, P. Ciccarese and B. Young, Web annotation data model, W3C Recommendation 23 February 2017, 2017, https://www.w3.org/TR/annotation-model/.

[34] B. Sateli and R. Witte, Semantic representation of scientific literature: Bringing claims, contributions and named entities onto the linked open data cloud, *PeerJ Computer Science* **1** (2015), e37. doi:10.7717/peerj-cs.37.

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

[36] D. Shotton, K. Portwin, G. Klyne and A. Miles, Adventures in semantic publishing: Exemplar semantic enhancements of a research article, *PLoS Computational Biology* **5**(4) (2009). doi:10.1371/journal.pcbi.1000361.

[37] M. Sperberg-McQueen and C. Huitfeldt, GODDAG: A data structure for overlapping hierarchies, in: *Proceedings of the 5th International Workshop on Principles of Digital Document Processing (PODDP 2000)*, 2004, pp. 139–160. doi:10.1007/978-3-540-39916-2_12.

[38] K. Taghva, A. Condit and J. Borsack, Autotag: A tool for creating structured document collections from printed materials, in: *Proceedings of the 7th International Conference on Electronic Publishing (EP 2007)*, 2006, pp. 420–431. doi:10.1007/BFb0053288.

[39] X. Tannier, J.-J. Girardot and M. Mathieu, Classifying XML tags through "reading contexts", in: *Proceedings of the 2005 ACM Symposium on Document Engineering (DocEng05)*, 2005, pp. 143–145. doi:10.1145/1096601.1096638.

[40] S. Teufel, A. Siddharthan and D. Tidhar, Automatic classification of citation function, in: *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)*, 2006, pp. 103–110. doi:10.3115/1610075.1610091.

[41] Text Encoding Initiative Consortium, TEI P5: Guidelines for electronic text encoding and interchange, 2016, http://www.tei-c.org/release/doc/tei-p5-doc/en/html/ (last visited May 30, 2017).

[42] S. Toulmin, *The Uses of Argument*, Cambridge University Press, Cambridge 1958, ISBN 9780521827485, http://johnnywalters.weebly.com/uploads/1/3/3/5/13358288/toulmin-the-uses-of-argument_1.pdf (last visited May 30, 2017).

[43] F. Vitali, A. Di Iorio and D. Gubellini, Design patterns for document substructures, in: *Proceedings of the Extreme Markup Languages 2005*, 2005, http://conferences.idealliance.org/extreme/html/2005/Vitali01/EML2005Vitali01.html (last visited May 30, 2017).

[44] N. Walsh, DocBook 5: The Definitive Guide, O'Reilly Media, Sebastopol, 2010, ISBN 9780596805029, http://tdg.docbook.org/tdg/5.0/docbook.html (last visited May 30, 2017).

[45] M.D. Wilkinson, M. Dumontier, I.J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. Bonino da Silva Santos, P.E. Bourne, J. Bouwman, A.J. Brookes, T. Clark, M. Crosas, I. Dillo, O. Dumon, S. Edmunds, C.T. Evelo, R. Finkers, A. Gonzalez-Beltran, A.J.G. Gray, P. Groth, C. Goble, J.S. Grethe, J. Heringa, P.A.C't Hoen, R. Hooft, T. Kuhn, R. Kok, J. Kok, S.J. Lusher, M.E. Martone, A. Mons, A.L. Packer, B. Persson, P. Rocca-Serra, M. Roos, R. van Schaik, S.-A. Sansone, E. Schultes, T. Sengstag, T. Slater, G. Strawn, M.A. Swertz, M. Thompson, J. van der Lei, E. van Mulligen, J. Velterop, A. Waagmeester, P. Wittenburg, K. Wolstencroft, J. Zhao and B. Mons, The FAIR guiding principles for scientific data management and stewardship, *Scientific Data* **3** (2016). doi:10.1038/sdata.2016.18.

[46] J. Zou, D. Le and G.R. Thoma, Structure and content analysis for HTML medical articles: A hidden Markov model approach, in: *Proceedings of the 2007 ACM Symposium on Document Engineering (DocEng 2007)*, 2007, pp. 199–201. doi:10.1145/1284420.1284468.