SALT: Semantically Annotated LaTeX

Tudor Groza Siegfried Handschuh Hak Lae Kim

Digital Enterprise Research Institute
IDA Business Park, Lower Dangan
Galway, Ireland
{tudor.groza, siegfried.handschuh, haklae.kim}@deri.org

ABSTRACT

Machine-understandable data constitutes the basis for the Semantic Desktop. We provide in this paper means to author and annotate Semantic Documents on the Desktop. In our approach, the PDF file format is the basis for semantic documents, which store both a document and the related metadata in a single file. To achieve this we provide a framework, SALT that extends the Latex writing environment and supports the creation of metadata for scientific publications. SALT lets the scientific author create metadata while putting together the content of a research paper. We discuss some of the requirements one has to meet when developing such an ontology-based writing environment and we describe a usage scenario.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: Miscellaneous; I.2.7 [**Artificial Intelligence**]: Natural Language Processing; I.7.1 [**Document and Text Processing**]: Document and Text Editing; I.7.2 [**Document and Text Processing**]: Document Preparation

General Terms

Semantic Authoring

Keywords

LaTeX, semantic annotation, semantic document, authoring

1. INTRODUCTION

The vision of the Semantic Desktop aims on the integrated personal information management as well as on information distribution and collaboration. This will be enabled by the use of ontologies, semantic metadata, which is machine-understandable data, and semantic web protocols. Hence, semantic metadata constitutes the basis for the Semantic Desktop. To author and annotate semantic documents on the desktop is one mean to create semantic metadata.

In this paper we provide means to author and annotate Semantic Documents on the Desktop. In our approach, the PDF file format is the basis for semantic documents, which stores both a document and the related metadata in a single file. To achieve this we provide a framework, SALT that extends the Latex writing environment and supports the creation of metadata for scientific publications. SALT lets the scientific author create metadata while putting together the content of a research paper.

Previous work in the creation of semantic metadata and annotation of documents is mainly concentrated on the annotation of HTML documents for the semantic web. Most of these HTML annotation tools [14, 26, 5] were following an a-posteriori annotation step. In order to provide metadata about the contents of a web page, the author must first create the content and second annotate the content in an additional, a-posteriori, annotation step.

The a-posteriori approach is reasonable when the annotator is not the owner of the web document, as it is a common use case in the web. However, a-posteriori annotation puts an additional load on the author, when he is identical with the annotator. As a way out of this problem is the possibility to easily combine authoring of a document with the creation of the metadata describing its content. First steps towards this for HTML documents in the web context are described in [13].

HTML is the document format for the web and thus research on semantic annotation is centered around this. But, an important and dominant format on the desktop is the portable document format. PDF can be seen at the moment as the *de facto* standard in terms of electronic publishing, especially in the research area. However, we observed that there exists a small number of solutions for a-posteriori semantic annotation of PDF documents ([7]). Also – to our knowledge – there is no clear defined approach yet for *a priori* PDF annotation.

Our approach proposes a method for creating *a priori* annotations for PDF documents, by exploiting the rich environment provided by LaTeX. We support the method with *a document ontology* mapping the internal structure of the document, an *rhetorical structure ontology* describing the argumentative structure of research papers, and an *annotation ontology* gluing the annotation to the document and providing additional metadata information. The annotation process takes place while writing and the actual integration is realized at syntax level by exploiting regular LaTeX command plus the introduction of special annotation commands. The final result is represented by a semantic PDF document encapsulating instances of the aforementioned ontologies.

In the following we describe the preliminaries of this work (Section 2), sketch a use-case in Section 3. Then, we give an overview (Section 4) of the annotation and publication process. In Section 5, we describe the modularization of the used ontologies and introduce the annotation syntax. Before we conclude, we give a overview of related work and discuss some aspects of our solution.

2. PRELIMINARIES

In this Section we provide definitions of important terms we use subsequently and we explain basic design decisions.

2.1 Terminology

- **Semantic Document** A semantic document includes any information regarding the document and its relationship with other documents. In our cases this is a PDF document enriched with semantic annotations. A Semantic Document can explicitly refer to another document by using ontological relations. For example, document A refers to a claim in Document B – by refereing to the URI of the claim – and provide counter arguments.
- **Semantic Annotation** The term Semantic Annotation describes a process as well as the outcome of the process. Hence it describes i) the process of the addition of semantic data or metadata to the document given an agreed ontology and ii) it describes the semantic data or metadata itself as a result of this process. In our context semantic annotation is a set of instantiations attached to a PDF document. We distinguish i) instantiations of RDF classes, ii) instantiated properties from one class instance to datatype instance – also called attribute instance, and ii) instantiated properties from one class instance to another class instance.
- **Annotation Ontology:** We use this term to denote a vocabulary which relates instance of an document ontology with annotations. The annotation could be instances of an arbitrarily ontology. In our case these are either instances of i) the rhetorical structure ontology or ii) a domain ontology associated to the topic of the document (e.g. about biology). The annotation ontology describes what an annotation is and which relations are possible between the subject and the object of annotation. Further, the annotation contains attributes, which are useable to describe the metadata of a document, such as author, title of the document (cf. Section 5.1.2).
- **Document Structure and Type Ontology:** in our context is a explicit shared formal specification of a document. This contains the document structure, the type, organization and the relationship between documents and other concepts. We will call this ontology **Document Ontology** (cf. Section 5.1.1) for short.
- **Rhetorical Structure Ontology:** We use this term to denote a vocabulary modeling the rhetorical structure of the text (RST) inside a document (cf. Section 5.1.3). RST captures the roles of every part of the text and tries to provide a plausible reason for its presence. RST describes the text on a *generic* and on a *specific* level.
  - The generic level describes parts of a scientific document such as motivation, background, scenario or contribution. The generic level is thus a modification and extension of the ABCDE format[6]. Apposed to the ABCDE format, we did not have an application for the Annotation and Entities part, since this is covered by our Annotation Ontology. But we missed other parts such as Motivation and Scenario.
  - The specific level is denoting rhetorical relations, for example, Concession, Circumstance or Means (cf. [22]) and thus allows a fine-grained description of the argumentation in a scientific document.

2.2 HTML, PDF and XMP

While HTML documents offer the possibility of accessing their composing *objects*, like the text or images, because of its implicit structured text-based format, not the same thing can be said about PDF documents. They have a totally different internal organization representing a combination of several types of complex objects and streams [17] together with their associated properties. Thus, post-creation analysis of the content depends on a handful of parameters, as accessing rights, image analysis or text retrieval algorithms accuracy.

A similar situation can be found also when analyzing the annotation support. HTML documents enable metadata (annotation) storage directly inside them, without the need of complex operations (including instances of ontologies). In the PDF documents case, this support is split between capturing metadata using a limited set of DublinCore [1] elements, in the XMP [16] field and creating annotations in forms of, for example, notes or markups. There is no natural way of embedding instances of ontologies in PDF documents, without either changing the document internal structure, which can be done using Adobe SDK [15], either re-modeling the XMP field. Our approach follows the second possibility, by encapsulating in the XMP field instances of the document, the annotation ontology and the rhetorical structure ontology, as well as arbitrary annotations of the user.

3. USE-CASE

In the following, we describe a use-case[1] that is supported by SALT and that has guided our development of the framework. The use-case requires the generation of metadata given by a PDF document.

The Use-Case shows how a semantic document enables an easy, low-effort information distribution, collaboration and integration for the purpose of an innovative online workshop proceeding. The goal is not only to ease the process of the creation of the online proceedings but also to provide added-value to the reader of the proceedings. In a way that the scientific contributions in the papers are easier to read and browse in the online proceedings.

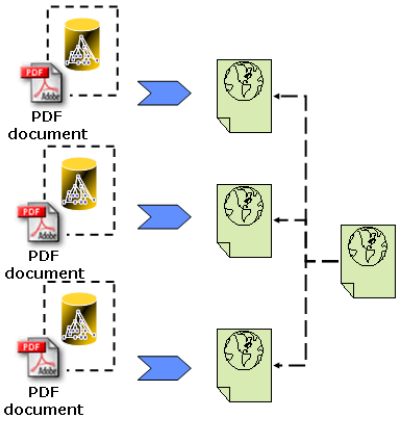

Figure 1: Information workflow in the workshop proceedings publication scenario.

The process for the online publication of the accepted workshop papers is usually done manually. The editor creates typically a list

[1]The use-case is inspired by discussions with Anita de Waard, see also http://wiki.ontoworld.org/index.php/ABCDEF

composed of the authors and the titles and links the corresponding PDF document to it [2].

However, additional information can easily be retrieved given that each scientific author will utilize the SALT framework for the writing of his scientific document. SALT enables a combination of automatic retrieved annotation based on i) an analysis of the used Latex commands, ii) annotation from the user about the rhetorical structure of the document, and iii) arbitrary annotation of the document. Hence, among other the semantic metadata will describe the underlaying ideas in the paper which can easily be exploited when presenting the proceeding.

Figure 1 depicts the information workflow in the current scenario. We assume that the accepted papers are enriched with our *rhetorical structure ontology*, thus we take advantage of it the first processing phase and generate an individual HTML page for each paper, containing the usual metadata plus the annotations captured by rhetorical structure. The second phase of the process, iterates over all created pages and generates an entry point in the form of an index page.

The index page gives a short overview of all papers, but more information – generated from the metadata – is available. Readers can quickly glance through the contribution and skip to the section they are interested in. For example, the context of each paper is shown, the background and the contribution, but also the individual claims are available.

4. ANNOTATION AND PUBLISHING

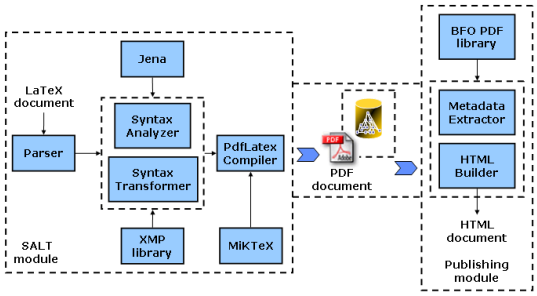

Figure 2: Component view.

We implemented SALT and the workshop proceedings publication scenario as two independent modules. The first module creates and embeds the metadata into the document, while the second one is using them to achieve the needed functionality. In figure 2 we present the organization of the two modules together with the third-party used components. Following, we will detail them separately.

4.1 The SALT Process

The SALT module is responsible for embedding the instances of the mentioned ontologies into the resulting PDF document. In order to reach the final result, there are a series of processing steps that need to be taken described as following.

Syntax analysis and annotation extraction. This first step takes as input the LaTeX document, parses it (Parser component) and extracts the annotations present in it (Syntax analyzer component), based on the three types of syntactic modifications detailed in Section 5.2. The result of this analysis process is a second LaTeXdocument (in an intermediary stage) and two sets of metadata: one which serves the population of the semantic layer, i.e. the ontologies with instances and the other creating the foundation for the PDF visual notes. Based on the output provided by this step, the following 2 steps could be theoretically performed in a parallel manner.

PDF notes embedding. The Syntax Transformer component takes the second metadata set (as described above) and based on its analysis creates the appropriate PDF visual notes, by making use of the special commands provided by LaTeX. All the annotations are then introduced into the LaTeX intermediary document in their original positions (extracted together with the original annotations).

Annotation analysis and ontology instantiation. In parallel to the previous step, the first metadata set is also analyzed. In this case, the focus is on the N3-like statements introduced in the usual LaTeX commands and on the commands pointing to the *rhetorical* structure of the document. Using the Syntax Analyzer in combination with Jena's [4] N3 to RDF transformer, the result of this step is the creation of the appropriate instances of the ontologies, in RDF format.

Final PDF document compilation. This final step has as input the LaTeX intermediary document and the instances created in the previous step. Its goal is to combine a PdfLatex compiler (in our case MiKTeX[3]) with the XMP LaTeX package [19] and transform the input set from LaTeX to PDF. The resulted PDF document will have incorporated the instances of the two ontologies and the visual notes.

The whole module is packed as a stand-alone component and it can be used from a command line interpreter or integrated as a library in a writing environment.

4.2 The publishing Process

The publishing module takes as input a PDF document, or a list of PDF documents and provides as output one or several HTML documents. The transformation process contains the following steps:

- extraction of the instances of the ontologies embedded in the PDF document(s)
- interpretation of the extracted metadata
- creation of the HTML documents based on some preferences expressed by the user

The first step is realized using the BFO PDF library [4] which provides the means for metadata extraction from PDF files. The resulted stream is passed to the Metadata Extractor component which separates the instances of the document, annotation and rhetorical ontologies and prepares them for interpretation.

For publication, the user can specify a series of parameters dealing with visual aspects of the publication, like font sizes, positioning or color, and with content aspects, for example, which annotated parts (or metadata) should be published. All these preferences are taken into account when interpreting the extracted instances and applied during the creation of the HTML documents. The whole process is iterative, starting from the first specified file to the last one. The finishing touch done by the HTML Builder is the creation of the index file pointing to all previously created HTML documents.

[2]For examples see the workshop online proceedings at CEUR-WS.org

[3]http://www.miktex.org/

[4]http://big.faceless.org/products/pdf/

5. THE SYNTACTIC AND SEMANTIC LAYERS

As briefly discussed in Section 2.2 one can embed annotations in PDF documents by filling the XMP field with DublinCore metadata elements or by making use of notes, bookmarks or markups. We propose a method for the creation of Semantic Documents by exploiting and extending the two aforementioned approaches. The actual transformation combines two interlinked layers: a semantic layer and a syntactic layer.

The semantic layer consists of the three ontologies, the document ontology, the annotation ontology and the rethorical structure ontology (cf. 2.1). The metadata based on these ontologies is places in the XMP field and thus extending the regular DublinCore elements of a PDF document.

The syntactic layer proposes the enrichment of the LaTeX syntax with i) an analysis of the used commands, ii) the provision of additional commands and iii) arbitrary annotation of the document based on N3 statements. This level has the goal to create a semantic bridge between the actual document and its metadata.

The motivation for introducing these two layers relies in the necessity of a much richer platform for embedding semantic annotations, which should also profit by the visual impact offered by the usual PDF annotation means. Following we will detail both the semantic and syntactic layers.

5.1 The semantic layer

The goal of the semantic layer is to define a proper semantic framework supporting the entire annotation process. We used three levels, each level represented by an ontology:

Document structure level capturing the ordinary structure of the document.

Annotation level , creating the bridge between the rhetorical structure and ordinary structure. It also captures additional metadata about the document.

Rhetorical level which models the document in terms of rhetorical elements and builds its rhetorical structure.

An overall image of the organization of the semantic layer is presented in Figure 3. In the following we will detail each of the three ontologies.

5.1.1 The Document Ontology

The document ontology, depicted in Figure 4, captures the structural layout of the document and to maintain instances of the annotated parts of the document. This represents an intermediary solution, until we will be able to use the XPointer framework [11] (cf. Section 7).

The motivation behind this level of decomposition is given by the need of to instantiate the annotated parts of the text. Also, the sentence represents at the moment the finest granularity for creating annotations and the referenced base for the construction of rhetorical structure. As an example, an populated instance of the document ontology will contain instances for all the words annotated during the writing process.

5.1.2 The Annotation Ontology

As mentioned before, the main role of the annotation ontology (Figure 5) is to relate the document ontology and the rhetorical structure ontology. Conceptually, the rhetorical structure represents an annotation of the ordinary structure. Thus, one is able to enrich the document with rhetoric elements by attaching semantic annotations to it. In ontological terms, this would translate to creating

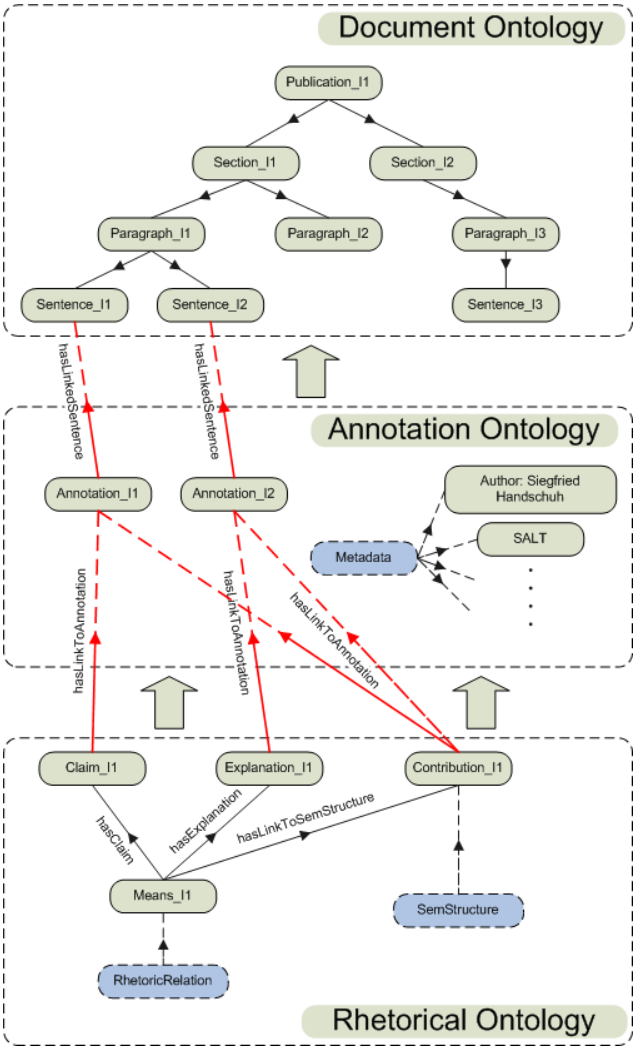

Figure 3: The internal organization of the semantic layer.

instances of the `Annotation` concept and attaching them to the appropriate parts of the text.

A second role of the ontology is to provide metadata about the publication as a whole. This part can be seen as an alignment to the DublinCore initiative, showing also our support for it. Each of the concepts, part of the metadata, has a direct correspondence in a DublinCore element. For the future, we intend to maintain this alignment by extending the ontology in parallel with the evolution of the DublinCore schema.

5.1.3 The Rhetorical Structure Ontology

The rhetorical structure ontology represents a perfect union between the knowledge captured by the rhetorical relations created between some parts of the text, the rhetorical structure modeling the positioning of the contained information chunks and the argumentative support providing the mean for building a stable foundation for the rhetoric elements. Following, we will analyze the three mentioned sides of the ontology.

The first side of the ontology deals with modeling the information chunks present in the document as rhetoric elements. This approach has its roots in the Rhetoric Structure of the Text (RST) theory [22], which describes the text in terms of the rhetoric rela-

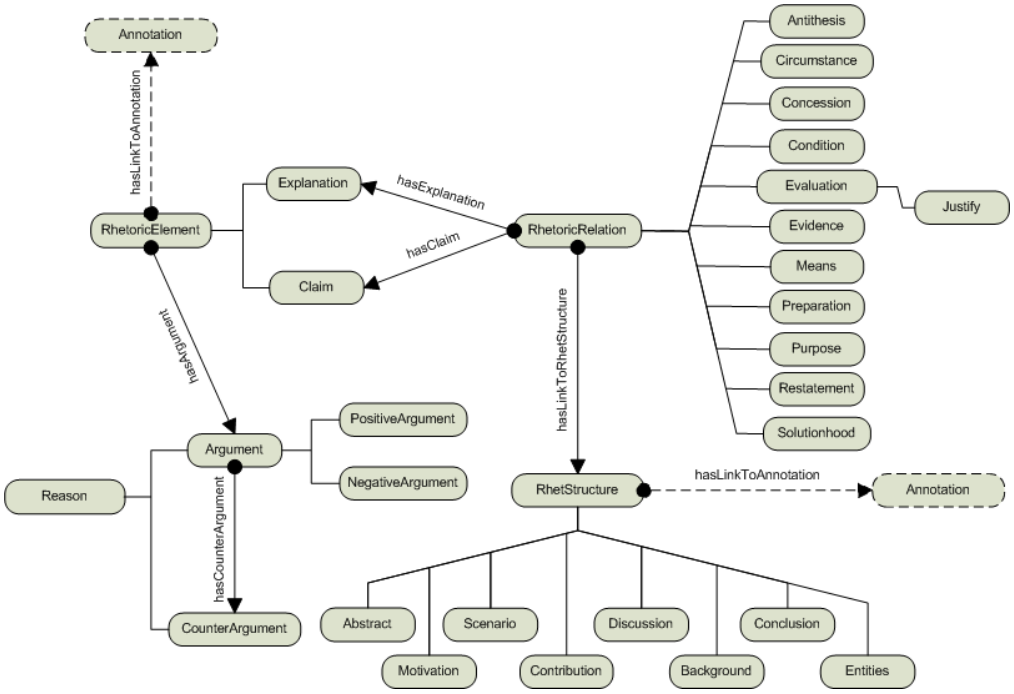

Figure 6: The Rhetorical Structure Ontology.

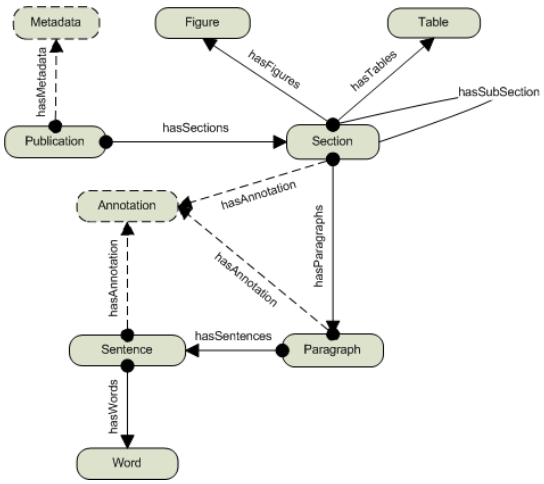

Figure 4: The document ontology.

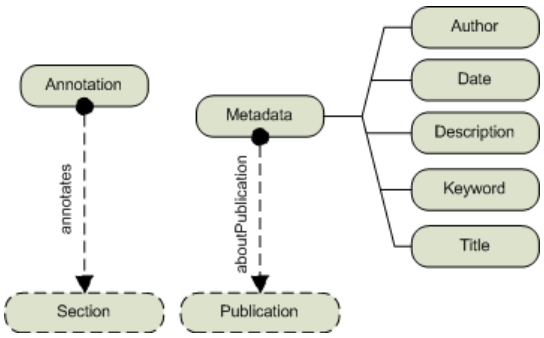

Figure 5: The annotation ontology.

tions existing between a Nucleus (modeled by us as the *Claim*) and a Satellite (in our case, the *Explanation*). Although the theory contains around 30 such relations, we considered only the ones which have a bigger impact (and relevance) when annotating a scientific document (e.g. *Antithesis, Concession* or *Means*). The main role of these rhetoric relations (modeled by us as concepts) is to provide a reason for the existence of the claims and the explanations in the text. Furthermore, we considered their placement in the frame created by the rhetorical structure (captured by the second side of the ontology) as a natural integration and thus we introduced a relation between the rhetorical relation concept and rhetorical structure concept.

The second side of the Rhetorical Structure Ontology takes care of capturing the rhetorical structure of the document. It represents an extension of the ABCDE format proposed in [6] that stands for: **A**nnotation, **B**ackground, **C**ontribution, **D**iscussion, **E**ntities.

As a starting point, this organization reflects a good image of a typical scientific document. But we argue that it is not enough. Therefore we propose its modification and extension with a small number of concepts, giving birth to a comprehensive rhetorical structure which could be adopted for all scientific documents.

The modification is the replacement of the Annotation concept with the Abstract concept, since the whole rhetorical structure represents in the end an annotation of the document. In terms of extension, we propose the introduction of the following concepts: Motivation, Scenario and Conclusion, which have as foundation rhetorical relations, but we considered that by using them as concepts of the rhetorical structure, we are able to model a complete best practice structure for scientific documents.

The two sides of the ontology described above are part of the *a priori* annotation process. This third side, deals with the discussions in terms of Arguments and CounterArguments, that can be initiated based on the existing claims. The motivation relies on

building a stable foundation for the claims by augmenting them with positive and negative argumentations. Therefore, we have foreseen the need for *a posteriori annotations* modeling these discussions and provided as part of the *Rhetorical Structure Ontology* the `Argument` and `CounterArgument` concepts, together with their subconcepts and relations.

In order to have a full understanding of how the result of the annotation process looks like from the Rhetorical Structure Ontology point of view, we provided in Figure 7 an example of instantiation. The example shows how a part of the text can be modeled in terms of rhetorical elements, and how can the rhetorical relations be created.

Consider the given phrase: ... *the visual system resolves confusion by applying some tricks that reflect a built-in knowledge of properties of the physical world.* The writer splits it into the `Claim` and the `Explanation`, and therefore instantiates two rhetorical elements, which can be further identified by their unique associated ID. Now, based on the definition of the *Means* rhetorical relation[5], the writer can make the reader aware of it, and thus emphasize his idea, by creating an instance of the concept modeling this relation. This instance is then linked with the appropriate concept from the rhetorical structure, exemplified in this case by `Contribution`. In terms of argumentative discussions, the example shows how can the claim be afterwards linked to instances of positive or negative arguments and how are the counter arguments instances modeled in relation to the initial arguments.

5.2 The syntactic layer

The second layer introduced for embedding annotations into the PDF documents, is the syntactic layer. Since we are targeting *a priori* annotations, created manually during the writing process, our approach proposes the enrichment of the LaTeX syntax in three ways:

- through command syntax extension
- by embedding N3-like statements in usual commands
- by introducing new commands

Our goal for this modified syntax structure is to have a lightweight form and as close as possible to the usual one, in order to avoid an overkill for the ordinary users. Therefore, the first two types of modification, i.e. command syntax extension and N3-like statements integration, maintain the syntactical core of the command, while the third one introduces simple new commands having similar syntax as the usual ones. The resulted mixture of commands has the most natural LaTeX form possible.

Command syntax extension. The syntax extension process was developed for the commands which have as main goal the structuring of the document. Therefore, commands like **abstract**, **section** or **subsection** were extended with a new field meant for assigning comments – free text annotation – to the corresponding part of the document. The field is delimited by a pair of curly brackets.
Example: `section`{Introduction}{*[...]*}

N3-like statements integration. This second type of modifications is the usage of N3-like statements in conjunction with LaTeX commands. These statements model information about the document as subject, or about an arbitrary subject. This enrichment was inspired by the N3 notation [2] and we believe that it represents a lightweight and easy enough notation to be adopted for creating semantic annotations in a scientific document.

New commands. In order to be able to manually annotate the document with the rhetorical structure, we introduced a series of new commands, similar to the usual LaTeXones. For example, *Background* or *Motivation*. All the newly introduced commands support also the extension described above.

Figure 8 depicts the result of the overall annotation process using SALT. The first operation is the parsing of the existing LaTeXdocument and the metadata extraction from the usual commands and the N3-like statements. In the figure these are represented by the *Author* and *Title* commands and the N3 statements about the topic of the paper, having as foundation the SWRC ontology.

The second operation is the instantiation of the Document Ontology (also presented in the figure), based on the document's structural information. Following, SALT analyzes the command extesions (like the one for *Use-case section* in the figure) and the newly introduced commands and environments (like *claim, explanation* or the *scenario* environment). It builds the rhetorical structure based on them and represents it as an RDF graph. To be remarked that each rhetoric element has a label attached (here, *c1, e1* and respectively *p1*) with the purpose of future referencing.

The final step is embedding the necessary information in the PDF document for the creation of the visual notes. The example shows the visual note attached to the *Use-case section* and the visual notes representing the beginning and the end of the *Scenario* rhetorical branch. The latter cotains also the information about the rhetorical relations found as part of this branch.

In general, the three main phases of the overall process are:

The creation of the semantic annotations and thus the document enrichment during the authoring process.

The ontology instantiation from the created annotations, together with the creation of the semantic links between the three levels of the semantic layer.

The visual representation of some of the annotations in the resulted PDF document.

In conclusion, we make a short analysis of the modifications. In the first case, switching from the usual LaTeXcommands to the extended ones by adding the annotation field should be straightforward, and should be considered an enrichment rather than a way for confusing the ordinary users. The second modification, i.e. the introduction of the `N3`-like statements, enables the author to insert arbiters annotations. The last category of modifications represented by the addition of new commands, was necessary in order to represent the rhetorical structure of document.

6. RELATED WORK

To ease the reasoning or retrieval of documents published on the Desktop or Web, the documents should be classified in a way that users find helpful and meaningful. There exist several activities focused on semantic annotation as a way to enrich a document, making it machine-readable and also accessible to humans. The Writing in the Context of Knowledge(WiCK) project aims to produce a novel writing tool to help authors improve the coherence and consistency of the documents they are creating by helping to assimilate key knowledge in each new document[3]. CREAM is a comprehensive framework which is specialized for populating HTML

[5]The *Means* rhetorical relation states that the *Explanation* presents a method or instrument which tends to make realization of the *Claim* more likely

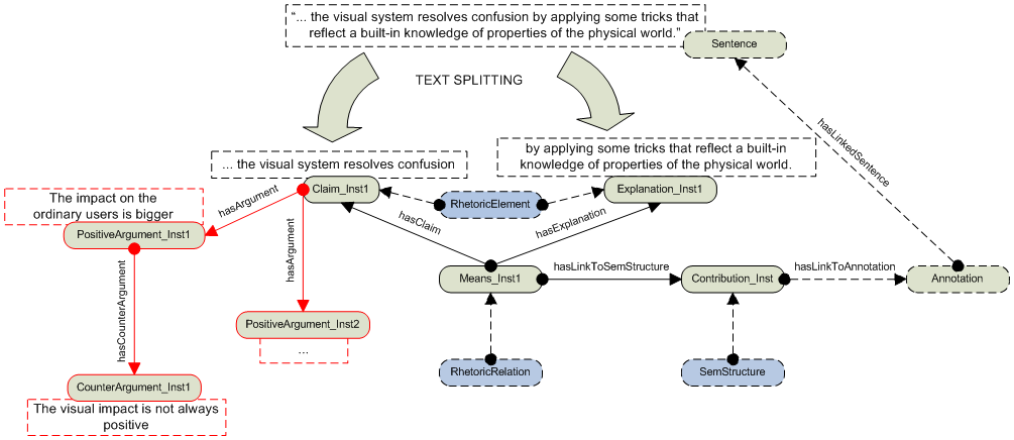

Figure 7: An example of instantiation of the Rhetorical Structure Ontology.

pages with ontological concepts. It allows authors to build the documents by dragging and dropping concepts and property from the ontology browser to a text editor [13].

Most activities have proposed their own semantic structure based on ontologies. Ontological structures allows not only fundamental values for semantic annotation, but also for additional possibilities such as inferencing or semantic retrieval [20]. The Semantic Web Research Community (SWRC) ontology is originating from OntoWeb, which can be used to provide detailed information about research work. It models the Semantic Web research community included researchers, publications, tools, and topics.

Generally speaking, semantic documents include any information regarding the document and its relationship with other documents [12]. Therefore, a semantic annotation of documents formally identifies concepts and relations between concepts in documents, and is intended primarily for use by machines[24]. There are several efforts relevant related semantic documents such as SemanticWord[23], OntoOffice[10], and SemTalk[8]. Eriksson[7] propose the PDF backend approach which is to use PDF as the basis for Protege storage backend. It allows users to store ontologies and knowledge bases inside PDF files. In some previous work however, an ontological information or metadata would exist in a different place than the document itself. XMP is a formats for embedding knowledge in documents[3]. Adobe's XMP[16] is a labeling technology that allows RDF constructs to be embedded in HTML, PDF documents and all Adobe formats.

In terms of the rhetorical structure of the text, [21] provides a deep analysis of the application domains in which it is used, e.g. computational linguistics, cross-linguistic studies or dialogue and multimedia. From our perspective, the work done by Geurts et. all[9] and Uren et. all[25] seems interesting, because they are among the only reference – to our knowledge – which try to model the rhetorical structure as an ontology.

[25] describes a framework for sensemaking tools in the context of the Scholarly Ontologies Project. Their starting point is represented by the requirements for a discourse ontology, having as foundation the structure of the claim. The resulted ontology finds its roots in the CCR (Cognitive Coherence Relations) Theory and models the rhetorical links in terms of similarity, causality or challenges. Their goal is to create and visualize claim networks using scholarly documents (represented as HTML files) using a central knowledge server. One of our future goals is also to create such knowledge networks, but using active reference embedded in the semantic document as an opposition to their central approach. Also, our solution places the annotations in their natural environment, i.e. as part of the document to which they are attached, and thus transforming it into a semantic document.

The second mentioned interesting reference was [9]. It models the process of transforming semantic graphs into multimedia presentations, using domain knowledge and discourse analysis. Their work is focusing more on using parts of the text for presentation purposes, as compared with our, which provides a method for enriching the normal documents with semantic annotations, based also on discourse analysis.

In this paper, we propose the document ontology to express much richer semantics in documents including the extension of the ABCDE format[6] for semantic structures of the document. From a representational and technical perspective, our approach differs from other approaches, in that ontologies support more sophisticated modeling for specifying relations of scientific documents. Moreover, an embedding technology using XMP provides efficient sharing support which makes it possible to share about the document itself.

7. DISCUSSION

In this Section we will raise some of the most interesting issues that appeared while researching the concepts presented in the current paper. Although the list could be much longer, we will resume ourselves to two of them: i) document instance maintenance and ii) object identification and reference, the latter being the source of problem also for the first one.

Our current approach solves the document instance maintenance issue by creating an instance for every annotated information chunk, the finest granularity being the *word*. The main reason is the (general) lack of a proper reference mechanism inside the PDF document, especially when created from LaTeX. Analyzing the provided solution, we could argue that it presents an possible advantage for a future development but in the same time also a quite clear disadvantage. The advantage consists in the possibility of representing the whole documents as instances of the document ontology and then using the instances for versioning purposes and semantic diff operations. Obviously, the semantics of the diff operation has to be firstly defined as part of a proper context, maybe in a similar way as realized in [27]. The disadvantage of this approach is the explosion in space of the document, considering the number of triples that need to be created for each word.

The second issue deals with object identification and reference. PDF documents have an internal organization represented by tree-

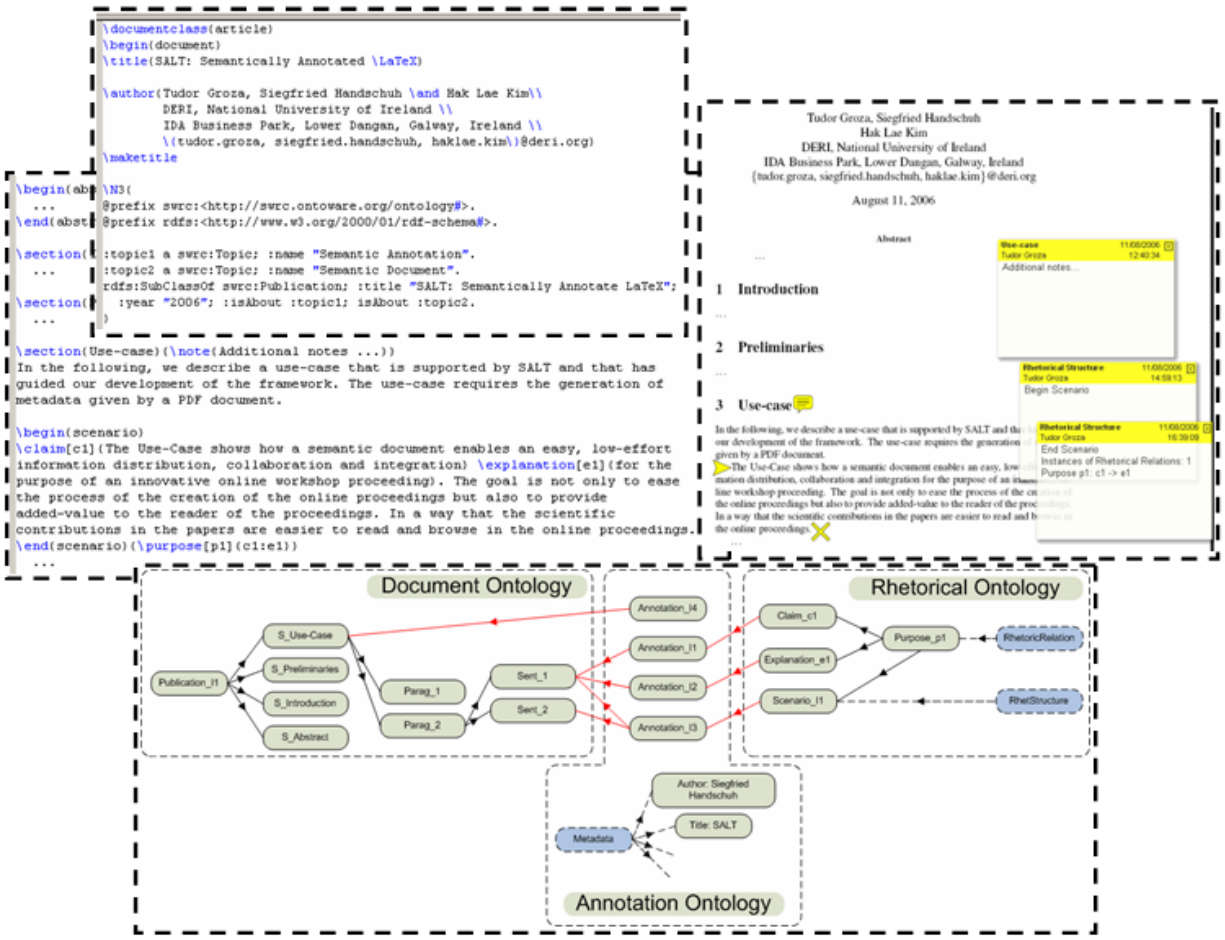

Figure 8: The result of the annotation process using SALT.

based structures of complex objects and streams [17], together with their associated properties. Post-creation analysis of the document, and thus the reconstruction of this internal organization, represent a hard task, due to the dependency on a handful of parameters, such as accessing rights, image analysis or text retrieval algorithms' accuracy. As a consequence, object referencing inside the document becomes also hard to accomplish.

On the other hand, we are dealing with *a priori* annotations, which makes the situation even more complex. The annotation process takes place during the writing process, in the LaTeX environment, and thus, the targeted PDF document does not even exist yet. Still, to be able to reference the annotated parts of the document, we adopted the following solution: The document structure was captured in the document ontology, and therefore giving us the means of referencing the information chunks having a sentence granularity. For referencing inside the sentence (word granularity) we introduced a base and an offset, pointing to the needed part of the sentence.

As a future improvement of this process, i.e. reference inside the document, we intend build a DOM-like model (or a B-Tree model) of the LaTeX document and map its structure to the tree-based internal structure of the PDF document. This approach would give us the following advantages:

- In terms of identification, we would be able to provide a unique identification for each information chunk, in the context of the document.
- In terms of reference, we would have the opportunity of using the XPointer framework [11] in conjunction with the document's model.

The combination of the two afore mentioned issues could start a new direction for creating semantic knowledge networks using information chunks from documents, by means of active references, rather than the existing static links. One would we able to directly embed a certain information object, or discuss a certain claim, in her scientific document, by providing only its active reference. The resulted semantic network tends come close to Ted Nelson's Xanadu vision[18].

8. CONCLUSION

In the paper we have described the authoring and annotation of a semantic documents to provide semantic annotation for the desktop. SALT leaves semantic data where it can be handled best, together with the document. Thus SALT provides a means to create Semantic Documents in a comparatively simple and intuitive way to use for LaTeX authors.

To attain this objective we have defined a SALT process, the appropriate Ontologies and the architecture. We have incorporated the means for rhetorical markup of a document that allows for example the scientific author to explicit markup his contribution and

the claims he made and the support for this claims. This explicit annotation provides, as shown in our scenario, a innovate and improved presentation and navigation of online proceedings. Furthermore, it will enables other authors to explicit and directly reference these claims and other related information. In the end this will lead to interconnected Semantic Documents.

For the future, there is a long list of open issues concerning the authoring of semantic PDF documents – from the more mundane, though important ones (top) to far-reaching ones (bottom):

1. PDF referencing, as we described it in Section 7
2. Creation of semantic knowledge networks using PDF document, by active references, also introduced in Section 7.
3. Automatic derivation of markup.
4. Other information structures (or formats), for example, incorporating not only the annotations created on the text, but also the ones created for the pictures, part of the Semantic Document.

We believe that these options make SALT a rather intriguing approach on which a considerable amount of scientific semantic documents might be build.

Acknowledgments

This work is funded by the European Commission 6th Framework Programme in context of the EU IST NEPOMUK IP - The Social Semantic Desktop Project, FP6-027705. Special thanks to Big Faceless Organization (`big.faceless.org`) for providing the PDF library used in the metadata analysis process. Further we thank Anita de Waard for fruitful discussions at ISWC 2005 and ESWC 2006.

9. REFERENCES

[1] DublinCore Metadata Initiative. http://dublincore.org/.

[2] Tim Berners-Lee. An readable language for data on the web - notation 3, 1998. http://www.w3.org/DesignIssues/Notation3.

[3] L. Carr, T. Miles-Board, G. Wills, A. Woukeu, and W. Hall. Towards a Knowledge-Aware Office Environment. In D. Karagiannis and U. Reimer, editors, *Proceedings of 5th International Conference on Practical Aspects of Knowledge Management (PAKM 2004)*, volume LNAI 3336, pages 129–140, 2004.

[4] J. Carroll, I. Dickinson, C. Dollin, D. Reynolds, A. Seaborne, and K. Wilkinson. Jena: Implementing the Semantic Web Recommendations. Technical Report HPL-2003-146, Hewlett-Packard, Dec 2003. http://www.hpl.hp.com/techreports/2003/HPL-2003-146.html.

[5] Fabio Ciravegna, Alexiei Dingli, Daniela Petrelli, and Yorick Wilks. User-System Cooperation in Document Annotation Based on Information Extraction. volume 2473, pages 122+, January 2002.

[6] Anita de Waard and Gerard Tel. The ABCDE format - Enabling Semantic Conference Proceeding. In *Proceedings of 1st Workshop: "SemWiki2006 - From Wiki to Semantics", Budva, Montenegro*, 2006.

[7] Henrik Eriksson. A PDF Storage Backend for Protege. In *Proceedings of the 9th Protege International Conference, Stanford, California, USA*, 2006.

[8] C. Fillies, G. Wood-Albrecht, and F. Weichardt. A Pragmatic Application of the Semantic Web using SemTalk. In *Proceedings of the Eleventh International World Wide Web Conference, Honolulu, Hawaii, USA.*, pages 686–692, 2002.

[9] Joost Geurts, Stefano Bocconi, Jacco van Ossenbruggern, and Lynda Hardman. Towards Ontology-driven Discourse: From Semantic Graphs to Multimedia Presentations. Technical report, Centrum voor Wiskunde en Informatica (INS-R0305), May 31, 2003.

[10] Ontoprise GmbH. OntoOffice Tutorial, 2003. http://www.ontoprise.de/documents/tutorial ontooffice.pdf.

[11] P. Grosso, E. Maler, J. Marsh, and N. Walsh. XPointer element() Scheme, 2003. http://www.w3.org/TR/xptr-element/.

[12] W. Guoren, W. Bin, H. Donghong, and Q. Baiyou. Design and Implementation of a Semantic Document Management System. *Information Technology Journal 4*, 1:21–31, 2005.

[13] S. Handschuh and S. Staab. Authoring and Annotation of Web Pages in CREAM. In *Proceedings of the 11th International World Wide Web Conference, WWW 2002, Honolulu, Hawaii, May 7-11, 2002*, pages 462–473. ACM Press, 2002.

[14] S. Handschuh, S. Staab, and A. Maedche. CREAM — Creating Relational Metadata with a Component-Based, Ontology-Driven Annotation Framework. In *Proceedings of the First International Conference on Knowledge Capture (K-Cap 2001)*, pages 76–83, Victoria, B.C., Canada, October 2001. ACM Press.

[15] Adobe Systems Incorporated. Adobe Acrobat SDK. http://partners.adobe.com/public/developer/acrobat/sdk/index.html.

[16] Adobe Systems Incorporated. Extensible Metadata Platform. http://www.adobe.com/products/xmp/.

[17] Adobe Systems Incorporated. PDF Reference - Adobe Portable Document Format, April 2004. http://partners.adobe.com/public/developer/en/pdf/PDFReference16.pdf.

[18] Ted Nelson. *Literary Machines: The report on, and of, Project Xanadu concerning word processing, electronic publishing, hypertext, thinkertoys, tomorrow's intellectual... including knowledge, education and freedom*. Mindful Press, Sausalito, California, 1981 edition: ISBN 089347052X, 1981.

[19] Maarten Sneep. The XMP inclusion package, 2005.

[20] S. Staab, A. Maedche, and S. Handschuh. An annotation framework for the semantic web. In *Proceedings of the First Workshop on Multimedia Annotation, Tokyo, Japan*, January 30-31 2001.

[21] Maite Taboada and William C. Mann. Applications of Rhetorical Structure Theory. *Discourse Studies*, 8, No. 4:567–588, 2006.

[22] Maite Taboada and William C. Mann. Rhetorical Structure Theory: looking back and moving ahead. *Discourse Studies*, 8, No. 3:423–459, 2006.

[23] Marcello Tallis. Semantic Word Processing for Content Authors. In *Proceedings of the Knowledge Markup & Semantic Annotation Workshop, Florida, USA, Part of the Second International Conference on Knowledge Capture, K-CAP 2003.*, 2003.

[24] Victoria Uren, Philipp Cimiano, Jos Iria, Siegfried Handschuh, Maria Vargas-Vera, Enrico Motta, and Fabio

Ciravegna. Semantic Annotation for Knowledge Management: Requirements and a Survey of the State of the Art. *Journal of Web Semantics 4*, 1:14–28, 2006.

[25] Victoria Uren, Simon Buckingham Shum, Gangmin Li, and Michelle Bachler. Sensemaking Tools for Understanding Research Literatures: Design, Implementation and User Evaluation. *Int. Jnl. Human Computer Studies*, 64, No.5:420–445, 2006.

[26] M. Vargas-Vera, E. Motta, J. Domingue, M. Lanzoni, A. Stutt, and F. Ciravegna. MnM: Ontology Driven Semi-Automatic and Automatic Support for Semantic Markup. In *EKAW02, 13th International Conference on Knowledge Engineering and Knowledge Management*, LNCS/LNAI 2473, pages 379–391, Sigüenza, Spain, October 2002. Springer.

[27] Max Voelkel and Tudor Groza. SemVersion: RDF-based Ontology Versioning System. In *Proceedings of the IADIS International Conference WWW/Internet (ICWI 2006), Murcia, Spain*, 2006.