

SALT - Semantically Annotated L^AT_EX for Scientific Publications

Tudor Groza, Siegfried Handschuh, Knud Möller, and Stefan Decker

DERI, National University of Ireland, Galway,
IDA Business Park, Lower Dangan, Galway, Ireland
{tudor.groza,siegfried.handschuh,knud.moeller,stefan.decker}@deri.org
<http://www.deri.ie/>

Abstract. Machine-understandable data constitutes the foundation for the Semantic Web. This paper presents a viable way for authoring and annotating Semantic Documents on the desktop. In our approach, the PDF file format is the container for document semantics, being able to store both the content and the related metadata in a single file. To achieve this, we provide a framework (SALT - Semantically Annotated L^AT_EX), that extends the L^AT_EX writing environment and supports the creation of metadata for scientific publications. SALT allows the author to create metadata concurrently, i.e. while in the process of writing a document. We discuss some of the requirements which have to be met when developing such a support for creating semantic documents. In addition, we describe a usage scenario to show the feasibility and benefit of our approach.

1 Introduction

The vision of the Semantic Web, as well as the personal Semantic Desktop aims at integrated personal information management, at information distribution and collaboration. This will be enabled by the use of ontologies, semantic metadata (machine-understandable data) and Semantic Web protocols. Hence, semantic metadata constitutes the foundation for Semantic Web and Desktop. Authoring and annotating semantic documents on the desktop is one of the possible means to create semantic metadata.

This paper introduces a new way for authoring and annotating Semantic Documents on the Desktop. In our approach, the PDF file format is used as the container for document semantics, being able to store both the content and the related metadata in a single file. To achieve this, we provide a framework (SALT - Semantically Annotated L^AT_EX¹) together with an associated ontology, that extends the L^AT_EX writing environment and supports the creation of metadata for scientific publications. SALT allows the author to create metadata while in the process of writing the content of a research paper.

Previous work in the creation of semantic metadata and annotation of documents has been mainly focused on the annotation of HTML documents for the

¹ Not to be confused with the SALT KA system by Marcus and McDermott.

Semantic Web. Most of these HTML annotation tools [1,2,3] are following an *a posteriori* annotation approach. In order to provide metadata about the content of a web page, the author must first create the content and then annotate it as an additional, *a posteriori* step. This approach is reasonable, when the annotator is not the creator of the web document, as it is a common use case in the web. However, if author and annotator are the same person, the possibility arises to easily combine authoring of a document with the creation of the metadata describing its content. We will call this approach *concurrent* annotation. First steps towards this approach for HTML documents in a web context are described in Handschuh et al. [4], or for blogs in Möller et al. [5].

HTML is the document format for the web, and thus research on semantic annotations is mainly centered around it. However, another important format is PDF – 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. We observed that there exist a small number of solutions for creating semantic annotations on PDF documents, most of them following the *a posteriori* approach ([6]). In the case of *concurrent* annotations – to our knowledge – there is no clear defined approach. Also, when it comes to embedding the semantic annotations in the document itself, the existing support is poor and rarely used.

Adobe has defined the Extensible Metadata Platform² (XMP), a platform (methodology, schemas, tools, ...) for embedding RDF metadata in data files. XMP supports metadata in a broad variety of file formats, among them PDF. However, even though it is possible to embed arbitrary metadata in a PDF's XMP field, in practice only shallow DublinCore³ descriptions are used. As a result, neither the inherent structure nor the semantic content of a document are reflected in the metadata.

Our approach proposes to extend the shallow metadata schemas currently used with a set of three ontologies which are able to capture the structural information of the document as well as the semantics of its content. The three ontologies are (i) the *Document ontology*, (ii) the *Rhetorical ontology* and (iii) the *Annotation ontology*. All three will be discussed in more detail in Sect. 3.1.

We support our proposal with a method for creating *concurrent* semantic annotations for PDF documents, by exploiting the rich environment provided by L^AT_EX. The annotation process takes place while writing and the actual integration is realized at syntax level by exploiting regular L^AT_EX commands plus a series of newly introduced special annotation commands. The final result is a semantically enriched PDF document encapsulating instances of the afore-mentioned ontologies together with associated visual annotations. We believe that the ontologies presented in our proposal can be used independently of the format used for the scientific publications. Therefore, we intend to use the current approach as a proof of concept and extend our investigations to other formats in the near future.

² Adobe Systems Incorporated - XMP. <http://www.adobe.com/products/xmp/>

³ DublinCore Metadata Initiative. <http://dublincore.org/>

In Sect. 2 we will present the automatic creation of online-proceedings as a use case for our framework. Then, we describe a modularization of the used ontologies and define the support for creating annotations, i.e. the annotation syntax (Sect. 3). In Sect. 4 we give an overview of the annotation process and revisit the proposed use case from the implementation point of view. Before concluding, we present a discussion of the proposed solution in Sect. 5, give an overview of the related work in Sect. 6 and discuss some aspects of our solution in Sect. 7.

2 Use Case

An increasing number of applications make use of metadata contained in PDF documents, or otherwise analyze the document's content. The type of functionality offered by such applications is varying from Personal Information Management (e.g. Gnowsis [7]) or searching (e.g. Beagle++ [8]) to digital libraries (e.g. JeromeDL[9]). All these applications have in common: (i) that they either use the limited metadata captured by the DublinCore elements present in the XMP field – which offers only shallow information about the document, or (ii) they perform full-text indexing in order to maximize the searching capabilities. Even though richer semantic annotations are in theory possible (and also in practice, as we show in this paper), they are currently not used. As a result, none of the applications mentioned can make use of them.

We believe that by using semantic PDF documents (i.e. PDF documents encapsulating rich RDF, e.g. instances of our ontologies), all the afore-mentioned applications would bring more value to the user: more accurate results, better visualization, etc. In order to provide an example, we will describe how such semantic documents enable an easy, low-effort information distribution, collaboration and integration for the purpose of an innovative online workshop proceedings. The goal is not only to ease the process of creation of the online proceedings, but also provide added value to the reader of these proceedings.

The process for the online publication of accepted workshop papers is usually done manually. The editor typically creates a list containing the authors and the titles and afterwards they link the corresponding PDF document to it. However, additional information can easily be retrieved given that each author would use our framework while writing the scientific publication. SALT enables a combination of automatically retrieved annotations based on i) the analysis of the used \LaTeX commands, ii) the rhetorical structure of the document and iii) the arbitrary annotations included in the document.

For our use case, we took the following approach: we first create an individual HTML page for each annotated paper (cf. Fig. 1). The rich annotations in each paper can be visualized and exploited for navigation in many different ways. In our example, we chose to present each paper in such a way that the focus is on the linear structure, including information regarding the rhetorical structure. In addition, the page also contains some simple metadata associated with the publication (such as title, authors, etc), the link to the PDF document and if desired even the original instances of the ontologies associated with the publication.

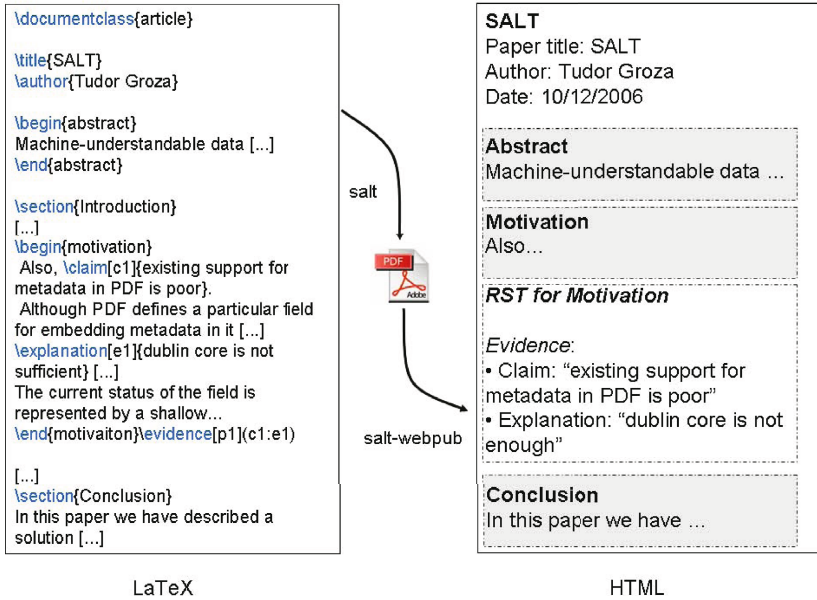


Fig. 1. HTML creation from annotated paper

The second phase of the process iterates over all the 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.

3 Ontological Foundation and Syntactical Support

There are two types of annotations that can be embedded in PDF documents: (i) visual annotations in the form of notes, bookmarks or markups and (ii) arbitrary metadata in the XMP field. Our proposal for the creation of Semantic Documents is exploiting and extending both possibilities. In the following, we will present both a semantic foundation — a set of three ontologies — and a means to express those semantics in an extended L^AT_EX syntax.

The semantic layer consists of three ontologies: document ontology, annotation ontology and rhetorical ontology. Instances of these ontologies will be placed in the XMP field, thus extending typical current use of PDF XMP, and providing a much richer environment for capturing the document's semantics.

The syntactical implementation proposes an enrichment of the L^AT_EX syntax. This is done by considering the existing commands and performing analysis and metadata extraction on them, and by introducing a series of new commands. These commands provide the support for creating rhetoric elements, creating

implicit and explicit visual annotations and for inserting arbitrary annotations in the document. In effect, the semantic layer creates a bridge between the actual document and its metadata.

3.1 The Semantic Layer

The goal of the semantic layer (see Fig. 2) is to define a proper semantic framework able to support the entire annotation process. As a result, we created a federation of three ontologies, enumerated as follows:

- Document ontology**⁴ – Capturing the internal structure of the document (sections, paragraphs, sentences, etc).
- Rhetorical ontology**⁵ – Modelling the document in terms of rhetorical elements and rhetorical structure (claims, evidence, etc).
- Annotation ontology**⁶ – Creating the bridge between the rhetorical structure and the ordinary structure. It also captures additional metadata about the document.

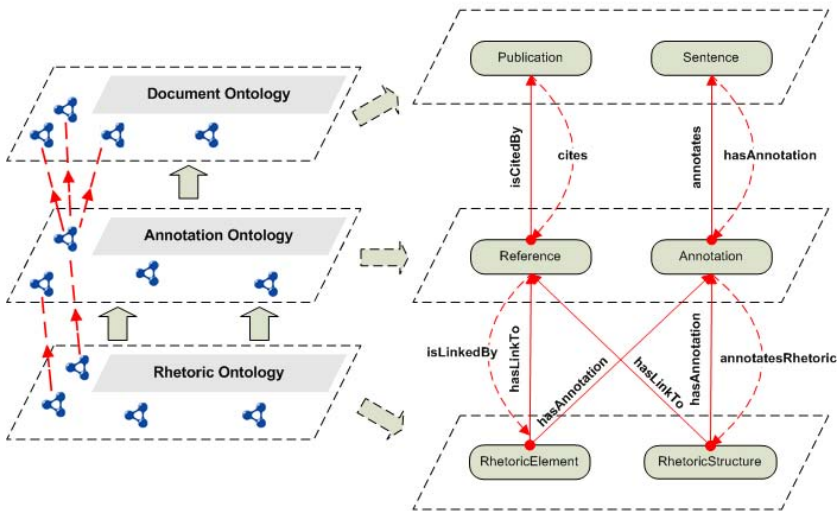


Fig. 2. Ontology Layers

The Document Ontology. The document ontology, depicted in see Fig. 3, captures the structural layout of the document and provides hooks to its annotated parts. The motivation behind the current level of decomposition is given by the need of instantiating the annotated parts of the text. The sentence

⁴ <http://salt.semanticauthoring.org/onto/2006/12/document-ontology.rdfs>

⁵ <http://salt.semanticauthoring.org/onto/2006/12/rhetoric-ontology.rdfs>

⁶ <http://salt.semanticauthoring.org/onto/2006/12/annotation-ontology.rdfs>

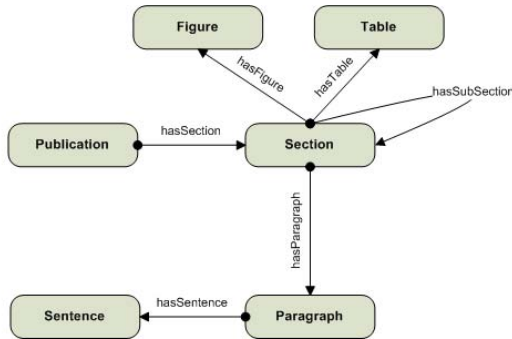


Fig. 3. The Document Ontology schema

currently represents the finest granularity of physical structure. However, they are mapped to specific substrings of a document using the `Annotation` class, which also allows to map arbitrary sub-phrases of a sentence.

The Rhetorical Ontology. The rhetorical structure ontology (see Fig. 4) represents a union of (i) the knowledge captured by the rhetorical relations within the text, (ii) the rhetorical structure modeling the positioning of the contained information chunks and (iii) the argumentative support providing the mean for building a stable foundation for the rhetoric elements. In the following, we will analyze the three parts of the ontology.

The first part of the ontology (*Rhetorical Relations*) 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 [10], which describes the text in terms of the rhetoric relations 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 currently only consider those that seem most relevant when annotating scientific documents (e.g. *Antithesis*, *Concession* or *Means*). The main role of these rhetoric relations (modeled as concepts) is to provide a reason for the existence of claims and explanations in the document. Furthermore, we considered their placement in the frame created by the rhetorical structure (captured by the second part of the ontology) as a natural integration and thus we introduced a relation between the rhetorical relation concept and rhetorical structure concept.

The second part of the ontology (*Rhetorical Structure*) takes care of capturing the rhetorical structure of the document. It represents an extension of the ABCDE format for the annotation of scientific papers [11]. ABCDE stands for: **A**nnotation, **B**ackground, **C**ontribution, **D**iscussion, **E**ntities. In SALT, we build on ABCDE, but propose a more comprehensive and fine-grained set of concepts. The simple metadata like title and authors is covered by the `Annotation` concept in ABCDE. In SALT, this is covered elsewhere (see the next section), which is why our **A** is the *Abstract* of the document. Furthermore, we extend ABCDE with the concepts *Motivation*, *Scenario* and *Conclusion*. Finally, the

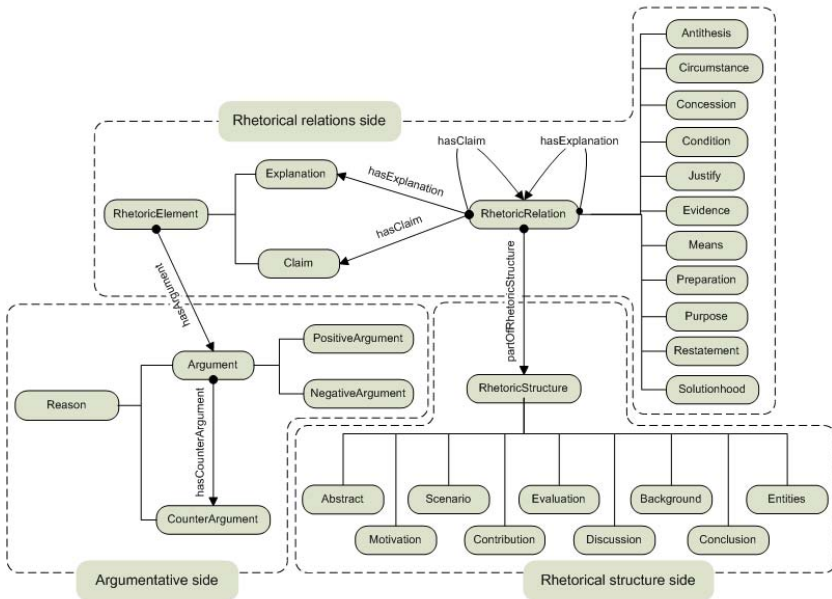


Fig. 4. The Rhetorical Ontology schema

Argumentative part of the ontology allows the further modeling of scientific discourse in the form of *Arguments* and *Counter Arguments*.

The Annotation Ontology. The main role of the annotation ontology (see Fig. 5) is to create the link between the document ontology and the rhetorical ontology. Conceptually, the rhetorical structure represents an annotation of the physical 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 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 and to the SWRC ontology [12], as each of the concepts corresponds directly to a DublinCore element or to a concept of the SWRC ontology.

3.2 Syntactical Support

Providing a syntactical implementation of the ontologies discussed above is done in two ways: the extraction of metadata from existing \LaTeX commands and the introduction of new commands. The new set of commands was kept small in order to avoid a steep learning curve for new SALT users. Functionality is extended in three directions:

Insertion of arbitrary annotations. This possibility was introduced in order to allow the authors to freely insert arbitrary metadata about the publication

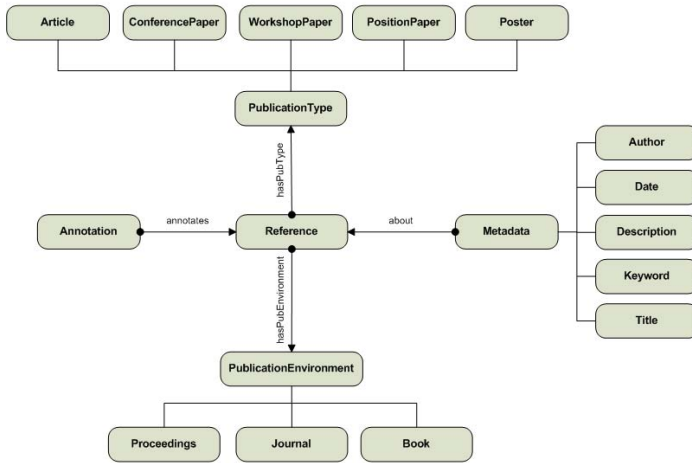


Fig. 5. The Annotation Ontology schema

by using the N3 notation⁷. As body of the `\N3` command, one can insert valid N3 statements, for example by referring to a specific domain ontology.

Creation of rhetoric elements. We allocated a special command for each type of rhetorical relation and a special environment for each type of rhetorical structure. As a foundation for all these, there exist also the commands for creating the basic rhetorical elements, i.e. the **Claim** and the **Explanation**. Here are some command examples: `\claim`, `\explanation`; rhetorical relations: `\antithesis`, `\concession`; rhetorical environments: `\begin{motivation}` ... `\end{motivation}`.

Explicit creation of visual annotations. The author themselves can create visual annotations by using the `\note` command, which has three parameters: the subject, the author and the content of the visual annotation. These annotations will e.g. show up as a little post-it in the PDF rendering.

A more detailed description of all the concepts present in the ontology, as well as of the annotation syntax can be found on the SALT web page: <http://salt.semanticauthoring.org/>

4 Annotation and Publishing

We implemented SALT and the workshop online proceedings publication scenario as two independent applications. SALT itself can be used stand-alone from the command line, or can be integrated in different L^AT_EX editors. For example, we integrated it in Kile⁸. However, one can integrate it in any editor which provides the flexibility of choosing a custom L^AT_EX - PDF compiler, not the implicit

⁷ <http://www.w3.org/DesignIssues/Notation3>

⁸ <http://kile.sourceforge.net/>

one. The second application, called SALT-WebPub, is a stand-alone application with an easy to use graphical user interface. In the following we will detail both applications separately.

4.1 The SALT Process

The SALT application is responsible for analyzing the annotations and embedding the ontology instances into the resulting PDF document. In order to create the final document, a series of processing steps need to be performed:

Syntactic analysis and annotation extraction. As a first step, the syntax tree of the L^AT_EX document is searched for elements which will add to the document metadata (both ordinary commands and new commands defined by SALT). The result are two separate metadata graphs representing both the physical document structure (according to the document ontology) and the rhetorical structure (according to the rhetorical ontology).

Annotation analysis and ontology population. In this step, both metadata graphs are joined. Also in this step the arbitrary RDF triples extracted from the N3 command are added to the graph.

PDF document compilation. In the final step, the PDF document is created using an ordinary PDF_Latex compiler (the user can choose which compiler to use). Afterwards, the complete metadata graph is added in the document's XMP field, and visual annotations (notes, etc) are added.

4.2 The Publishing Process

SALT-WebPub, the publishing application, takes as input a list of semantic PDF documents and generates a set of corresponding HTML files, together with the associated index. In order to provide flexibility to the format of the resulting HTML files, we let the user specify a template for the page associated with each publication and a template for the index file. This way, it is possible to customize the presentation of the online proceedings without affecting the web page content generation.

The process of generating generating the proceedings from the PDF documents is split into a series of steps. The first step is to extract the ontology instances out of the document. The second step is to interpret the extracted metadata and to prepare it for the final output format. The last step creates the associated HTML page by taking the user's template and filling it with the output from the previous step. Finally, the index page is created, based on the information extracted from each individual document.

5 First Experiences

To get a better idea of how SALT works “in real life”, we performed a test with a group of six authors from this the SAAW2006 workshop⁹. Together with the

⁹ <http://saaw2006.semanticweb.org>

authors, we annotated their L^AT_EX source code and generated semantic PDF documents using our tools. Taking all documents, we produced a richly annotated online proceedings¹⁰. Both the author's feedback and our own observations are summarized in the following discussion.

Ontological foundation – The three ontologies discussed in Sec. 3.1 did undergo small modifications as a result of the evaluation. For a more comprehensive capturing of the shallow metadata about scientific publications we felt the necessity of adding several concepts in the *Annotation Ontology*, like *PublicationType* or *PublicationEnvironment*. Also, in order to build the support for creating semantic network between the annotated documents, we had to introduce the *Reference* concept in the same ontology and link the rhetorical elements and the rhetorical structure to it. Regarding the *Rhetorical Ontology*, the only necessary modification needed was to allow the rhetorical relations to act as rhetorical elements in more complex relations. This modification provides a better degree of flexibility and allows the creation of rhetorical structure of text trees (one of the possible views over a Semantic Document – see Sect. 2). The ontology layer as a whole was seen as comprehensive enough to capture both shallow metadata and the content's semantics.

Annotation syntax – The proposed L^AT_EX syntax was very well received. The number of newly introduced commands was small enough not to create any significant extra workload on the authors. Based on this, we intend to leave the syntax untouched. Some small modifications were necessary, in order to reflect the actual status of the ontologies.

A general issue that we discovered was that we need to take more into consideration the semantics of the existing L^AT_EX commands and the overall structure of the L^AT_EX documents. Therefore, our framework will support automatic information extraction from bibliographical items and from citing commands. In this way it will automatically create possible relations between instances present in the currently annotated scientific publication and the cited ones.

6 Related Work

As already mentioned, our ontologies have their roots in the Rhetorical Structure of Text (RST) Theory [10]. The paper provides the underlying semantics of the concepts modelled by the theory together with their definitions. A second publication by the same authors [13] provides a deep analysis of the application domains in which RST was used until a certain point in time. It is interesting to observe that the mentioned range of domains varies from computational linguistics, cross-linguistic studies and dialogue to multimedia presentations.

A similar approach is presented by Tempich et. al in [14]. The *DILIGENT Argumentation Ontology* was designed in line with the terminology proposed by

¹⁰ <http://salt.semanticauthoring.org/experiment/saaw2006/>

the IBIS methodology [15] and captures the argumentative support for building discussions in DILIGENT processes. In DILIGENT, the argumentative support is equivalent to one of the three parts of our Rhetorical Ontology and so is less expressive. Uren et. al [16] describe 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, which has its roots in the CCR (Cognitive Coherence Relations) Theory and models the rhetorical links in terms of similarity, causality and challenges. Although the ontological foundation is very similar, the application approach is different (see below).

In terms of applications, we found the approach by Peter et al. [17] to be one of the most interesting ones in terms of similarity with our research. Their goal is to extract semantics from a \LaTeX document content based on the references and index present in the document (for example *see* and *see also* references). We have a similar approach when it comes to extracting the structural information, but our focus is more oriented on the rhetorical structure of the text and the semantic links between claims placed in different documents.

MMISS [18] fits into the category of using \LaTeX as a development environment. The project aims at building an internet-based, adaptive multimedia educational system. Based on a series of custom \LaTeX commands they are able to build ontologies and semantically link the resulting lecture slides (via a central repository). SALT also uses custom \LaTeX commands to semantically annotate the document, the difference being that we embed the annotations in their natural environment (i.e. the resulting PDF document) and not in a central storage place. This is also one of the main differences when compared to the system developed by Uren et. al [16]. Their goal is to create and visualize claim networks using scholarly documents (represented as HTML files) using a central knowledge server. One of our goals is also to create such knowledge networks, but using active reference embedded in the semantic document.

Another interesting system is described by Geurts et al. [19]. It models the process of transforming semantic graphs into multimedia presentations, using domain knowledge and discourse analysis. Their work is focussing more on using parts of the text for presentation purposes. SALT on the other hand provides a method for enriching the normal documents with semantic annotations, based also on discourse analysis. However, in their approach it is not very clear how the knowledge base is structured and how they chose the domain knowledge effectively.

7 Discussion

In this section we will discuss a number of relevant issues that appeared while researching the concepts presented before focussing on: i) annotation instance generation and maintenance and ii) object identification and reference.

Annotation instance generation and maintenance refers to the mechanism of generating and mapping the ontology instances which annotate information chunks to the actual content present in the document. This issue is especially

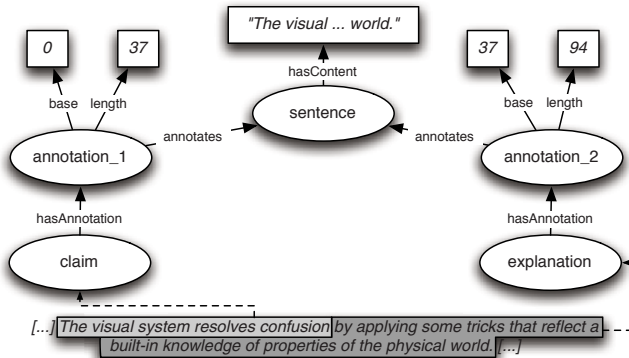


Fig. 6. Metadata Graph for a Sentence

sensitive when it comes to the document structure objects. In order to have a clear view over the subject, we will compare possible solutions for HTML and PDF documents.

In the case of HTML, if we would like to create an annotation instance and connect that instance to the piece of text being annotated, we could do this directly by referencing an element within the documents DOM tree. Thus, a simple pointer solves the problem, which is not the case for PDF documents. Although the internal organization of the document is represented by a tree of complex objects and streams, referencing inside this tree is not straightforward. The reasons are mainly related to accessing rights, image analysis or text retrieval algorithms' accuracy. At the same time, we also have to consider the fact that we're dealing with *concurrent* annotations, which makes the situation even more complex. Because the annotation process is interleaved with the writing process in the L^AT_EX environment, the targeted PDF document does not even exist yet.

Our current approach solves this issue by creating an instance for every annotated information chunk, the finest granularity being part of sentence. To give an example, consider the sentence in Fig. 6: The first part of the sentence is annotated as a claim, and the second part as an explanation. SALT will now (i) create a *Sentence* instance for the entire sentence, having the *hasContent* property set to the sentence's content, (ii) create a *Claim* instance, (iii) an *Explanation* instance and (iv) two *Annotation* instances connecting the claim and the explanation to the sentence. Finally (v) the *base* and *length* of the annotation instances are set to reflect their position in the sentence.

Obviously this solution presents two disadvantages: on one side, it increases the space of the document (linearly by the number of annotated sentences), and on the other side, it generates redundancy. In order to correct this issue, we intend to implement the XPointer Framework [20] applied for our case. The result will replace the actual content of a sentence with pointers in the PDF document to it, and thus the redundancy will be eliminated and the document space decreased, but it will increase the complexity of the metadata analysis

process, since it will introduce a pre-PDF-creation and a post-PDF-creation analysis step.

The second discussion issue which we would like to raise is the object identification and reference. One of our goals is to be able to create references between different rhetorical elements placed in different semantic documents, and to be able to provide arguments and counter-arguments based on the ontology support. In order to achieve this, the first step that we took is to impose the definition of a unique identifier for each rhetorical element present in the document. Thus, the identification inside one document is solved. There are two problems that appear now: (i) how can the actual reference between rhetorical elements be realized and (ii) what happens if there are two versions of the same document between which the element identification is different.

A possible solution for the first issue could be to impose the presence of a valid URL pointing to the original document for each cited publication. Thus, when referencing a rhetoric element in a particular publication, for example *[Handschuh2006]#claim1*, it will be possible to create a valid reference by resolving *[Handschuh2006]* to e.g. <http://example.org/handschuh2006.pdf>.

This solution brings us to the second problem. Usually a publication resides in more than one place, and there are cases in which the version of the publication differs from one place to another. The simplest solution to solve the issue would be not to care about the version. When citing a document, the author would provide a direct link to that document, and thus, all the references will be created based on that document, and presuming that the author realizes the referencing correctly. Of course, one could continue the discussion and raise another issue, i.e. what happens with journal articles and copyright issues regarding them, but we will tackle this point in future.

8 Conclusion and Future Work

In this paper we have described a solution for authoring and annotation of semantic documents. SALT leaves the semantic data where it can be handled best: within the document. Also, it provides a means to create Semantic Documents in a simple and intuitive way for L^AT_EX authors.

To attain this objective, we have defined the SALT process, the appropriate ontologies and the architecture of the application. We have incorporated the means for rhetorical markup of a document that allows the scientific authors to explicitly markup their contribution, the claims they made and the support for their claims. The framework brings added value to the applications using PDF documents and to the users, as shown in our online proceedings scenario, where we used it to automate the presentation and improve the navigation of scientific publications. Used in this way, SALT could also be integrated in the workflow of generating the semantic metadata for conferences such as ESWC or ISWC, a process that can be long and tedious if performed manually.

For the future, there is a list of open issues concerning the authoring of semantic PDF documents that we will consider: (i) PDF referencing or creation

of semantic knowledge networks by means of PDF documents and using active references, as we described it in Section 7, (ii) integrating the framework with existing (semantic) digital libraries and semantically-interlinked online communities and (iii) automatic derivation of markup. We believe that these options make SALT a good approach for the authoring of scientific semantic documents.

Our ultimate goal is to convince the community about the value that semantic documents bring and to transform the ontological framework into a *de facto* standard for annotating scientific publications. Thus, the next step that we will take is to perform an intensive evaluation phase with researchers coming from different backgrounds. This phase will provide us with both a better understanding of the weak points in our framework and with a perfect environment for improvement.

Acknowledgements. This work has been funded by the European Commission 6th Framework Programme in the context of the NEPOMUK IP - The Social Semantic Desktop, FP6-027705. We would like to thank Alexander Schutz for the fruitful discussions.

References

1. Handschuh, S., Staab, S., Maedche, A.: CREAM — creating relational metadata with a component-based, ontology-driven annotation framework. In: The First International Conference on Knowledge Capture (K-Cap 2001), Victoria, B.C., Canada (2001) 76–83
2. Vargas-Vera, M., Motta, E., Domingue, J., Lanzoni, M., Stutt, A., Ciravegna, F.: MnM: Ontology Driven Semi-Automatic and Automatic Support for Semantic Markup. In: The 13th International Conference on Knowledge Engineering and Knowledge Management, Sigüenza, Spain (2002) 379–391
3. Ciravegna, F., Dingli, A., Petrelli, D., Wilks, Y.: User-system cooperation in document annotation based on information extraction. In: The 13th International Conference on Knowledge Engineering and Knowledge Management. (2002) 122+
4. Handschuh, S., Staab, S.: Authoring and annotation of web pages in CREAM. In: 11th International World Wide Web Conference, WWW 2002, Honolulu, Hawaii, ACM Press (2002) 462–473
5. Möller, K., Bojars, U., Breslin, J.G.: Using semantics to enhance the blogging experience. In: The third European Semantic Web Conference (ESWC2006), Budva, Montenegro (2006)
6. Eriksson, H.: A PDF storage backend for Protege. In: Proceedings of the 9th Protege International Conference, Stanford, California, USA (2006)
7. Sauer mann, L.: The Gnowsiss Semantic Desktop for information integration. In: The 1st Workshop on Intelligent Office Appliances: Knowledge-Appliances in the Office of the Future (IOA 2005), at WM 2005, Kaiserslautern, Germany (2005)
8. Brunkhorst, I., Chirita, P.A., Costache, S., Gaugaz, J., Ioannou, E., Iofciu, T., Minack, E., Nejd, W., Paiu, R.: The Beagle++ toolbox: Towards an extendable desktop search architecture. Technical report, L3S Research Centre, Hannover, Germany (2006)
9. Kruk, S.R., Decker, S., Zieborak, L.: JeromeDL - adding semantic web technologies to digital libraries. In: DEXA 2005, Copenhagen, Denmark (2005)

10. Taboada, M., Mann, W.C.: Rhetorical structure theory: looking back and moving ahead. *Discourse Studies* **8**, No. 3 (2006) 423–459
11. de Waard, A., Tel, G.: The ABCDE format - enabling semantic conference proceeding. In: *Proceedings of 1st Workshop: "SemWiki2006 - From Wiki to Semantics"* at ESWC2006, Budva, Montenegro (2006)
12. Sure, Y., Bloehdorn, S., Haase, P., Hartmann, J., Oberle, D.: The SWRC ontology - semantic web for research communities. In: *Proceedings of the 12th Portuguese Conference on Artificial Intelligence (EPIA 2005)*, Covilha, Portugal (2005)
13. Taboada, M., Mann, W.C.: Applications of rhetorical structure theory. *Discourse Studies* **8**, No. 4 (2006) 567–588
14. Tempich, C., Pinto, H.S., Sure, Y., Staab, S.: An Argumentation Ontology for Distributed, Loosely-controlled and evolInG Engineering processes of oNTologies (DILIGENT). In: *The Second European Semantic Web Conference, (ESWC 2005)*, Heraklion, Crete, Greece (2005) 241–256
15. Kunz, W., Rittel, H.: Issues as elements of information system. Working paper 131, Institute of Urban and Regional Development, University of California (1970)
16. Uren, V., Shum, S.B., Li, G., Bachler, M.: Sensemaking tools for understanding research literatures: Design, implementation and user evaluation. *Int. Jnl. Human Computer Studies* **64**, No.5 (2006) 420–445
17. Peter, H., Sack, H., Beckstein, C.: Document indexing - providing a basis for semantic document annotation. In: *XML-Tage 2006*, Berlin (2006)
18. Krieg-Brückner, B., Lindow, A., Lüth, C., Mahnke, A., Russell, G.: Semantic interrelation of documents via an ontology. In: *Proceedings of DeLFI 2004: Die 2. e-Learning Fachtagung Informatik*, Paderborn, Germany (2004)
19. Geurts, J., Bocconi, S., van Ossenbruggern, J., Hardman, L.: Towards ontology-driven discourse: From semantic graphs to multimedia presentations. Technical report, Centrum voor Wiskunde en Informatica (INS-R0305) (May 31, 2003)
20. DeRose, S., Maler, E., Jr., R.D.: XPointer xpointer() scheme (2002) <http://www.w3.org/TR/xptr-xpointer/>.