SALT: Weaving the claim web

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

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

Abstract. In this paper we present a solution for "weaving the claim web", i.e. the creation of knowledge networks via so-called claims stated in scientific publications created with the SALT (Semantically Annotated LaTeX) framework. To attain this objective, we provide support for claim identification, evolved the appropriate ontologies and defined a claim citation and reference mechanism. We also describe a prototypical claim search engine, which allows to reference to existing claims and hence, weave the web. Finally, we performed a small-scale evaluation of the authoring framework with a quite promising outcome.

1 Introduction

Semantic metadata consitutes the foundation for the Semantic Web and Semantic Desktop. One way to create semantic metadata is by authoring and annotating semantic documents on the desktop. In a previous paper [1] we have described the first version of SALT (Semantically Annotated LaTeX), an authoring framework for creating semantic documents for scientific publication. In our approach we used the PDF file format as container, thus being able to store both content and metadata in a single file. One of our main goals was to support the creation of knowledge networks via claims stated in scientific publications by leveraging the document semantics. Each of the claims is addressable by a unique identifier, and a set of such structured documents can be integrated to form a network of interlinked resources. In this way a semantic web of scientific literature is growing, document by document. Metaphorically speaking, the information in the documents is liberated from their surrounding document and can be integrated and re-used in new and unexpected ways.

We define claims as the most important statements within a publications i.e. the statements that characterize best the publication's content. We believe that by reading only the claims, a reader will get a clear view of the message of the respective paper.

An example of a claim is: *[The document structure in general influences the reader's perception on the document]*. The claim here is not a particular occurrence of text or a particular utterance, but rather the abstract idea of a statement as such. That said, a claim can have a number of textual representations, e.g. in

different scientific publications: (i) *The structure of a document has an important influence on its perception*, or (ii) *The document structure influences the reader's perception on the document*, or (iii) *The way people understand and perceive a text is often infleunced by the way it is structured*.

In this paper, we propose a solution for representing and identifying the claims in a publication, and through this, we introduce a novel way for authoring and referencing. While until now everything was done at a macro-level by citing a full paper, our solution will give authors the possibility of working at a finer-grained level, by being able to cite claims within a publication. To achieve this goal, a mechanism for global and local claim identification is needed. Local identification refers the claim in the scope of the document: given the PDF file of a publications, how can one identify and reference a claim and point to its representation in that file. Global identification means identifying a claim in the entire pool of scientific publications, without referring to a particular representation.

We present our extension of SALT, which adds capabilities for specifying, referencing and searching of claims. SALT allows the author of a scientific publication to enrich the document with formal descriptions of claims, supports and rhetorical relation as part of their writing process. In the previous version of SALT, the identifiers of those elements were defined manually by the author. This approach had the following limitations: (i) the identifiers had have a local scope, (ii) the way to identify and reference a particular representation of a claim was lacking and (iii) there was no way to search for claims globally. These limitations meant that identical claims in different papers could not be linked.

SALT is based on a set of three ontologies: (i) the Document Ontology, (ii) the Annotation Ontology, and (iii) the Rhetorical Ontology. To address the limitations given above and allow the modelling of multiple claim representations, we mainly extend the Rhetorical Ontology. We also implement a means to identify a representation inside a PDF document (as a pointer). In addition we introduce a BibTeX-like claim bibliography format and a series of special LaTeX commands for claim reference.

In Sect. 2 we detail the motivation of our work. Then we describe the way in which we model identification in the new version of SALT (Sect. 3). In Sect. 4 we present a prototypical implementation for a claim seach engine. The evaluation of the SALT framework is described in Sect. 5. Before concluding, we give an overview of the related work 6.

2   Motivation

Claim finding in scientific publications and a means to create claim identifiers can be modelled at different levels. We discuss this under the following aspects: (i) general publication management, and (ii) claim identifier management. In both cases we make the assumption that the publications in question are annotated using the SALT framework.

In the last decade the number of conferences, workshops and implicitly publications, increased substantially. For example, in the bio-medical domain, the
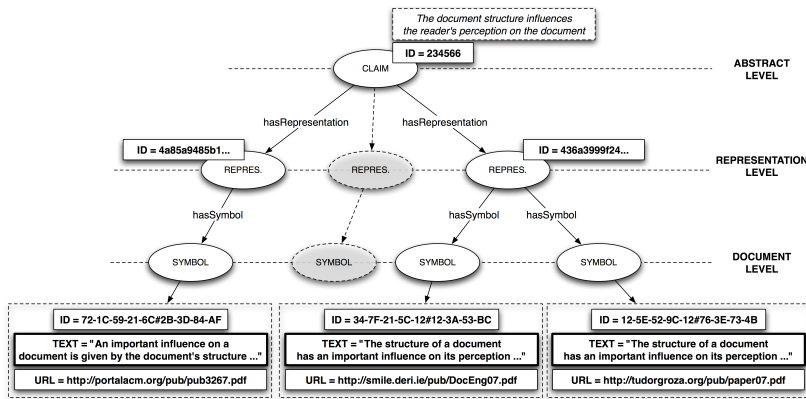
Fig. 1. Example of instantiation of the claim identification tree

number of publications in the MEDLINE 2004 database is currently growing by about 500.000 publications a year [2]. As a direct consequence, it has become very hard to find all relevant publications for a certain topic or stating a particular claim. The problem is also aggravated by the lack of uniformity in publishing papers or proceedings online. Publications are scattered on various Web sites, or, in the best case, on an online proceedings site. For the entirety of the publication space, there is no standard format for referring to publications.

Our model of claims and representations lends itself well to be mapped to the Semiotic Triangle [3], which illustrates the relations between (i) *objects* in the real world, (ii) *concepts* — abstract representations of objects in people's minds — and (iii) *symbols*, which are expressions of concepts (in the domain of language this is an utterance or a piece of written text). There is no direct relation between objects and symbols, but instead only an indirect one via the concept. In our model, *objects* correspond to *claims*, *concepts* to *representations* of a claim in a publication as such (e.g. "Joe's paper at ISWC2007") and *symbols* are the actual strings of text in a particular, physical PDF file. There are three layers that have to be taken into account regarding the identification of a claim: (i) the *abstract layer*, which represents the abstraction (the *object*) referred by the claim, (ii) the *representation layer*, which denotes the collection of forms that the claim can take. Each representation is uniquely identified and referenced by the source claim. (iii) The *document layer* represents the physical form of a representation (the *symbol*). There is a 1:n relation between the symbol and the representation. The symbol has a unique identifier based on the document where it occurs.

In Fig. 1 we show an example of instantiation of the claim identification tree. The claim is assigned a random identifier when it is (ID=234566). The symbol's identifier represents a pointer into the actual document, where the textual representation resides (here: ID=12-5E-52-9C-12#76-3E-73-4B). For the

representation in the text, we use the SHA1[1] sum of the symbol (for the symbol shown above, ID=436a3999f24...). The reason behind this approach, especially for the lower part of the tree is the following: the symbol's identifier will always be generated in the scope of the host document, independently of the actual text chunk representing the claim, and thus is modified each time the document is modified. The representation's ID is constant if the text chunk is the same. Thus, if the document is replaced but the text is unchanged, only the symbol needs to be updated.

By providing this layered approach to identification, the author can not only to reference the claim as such, but also a particular representation. As described in the following section, the identification at document level can be performed automatically.

3 Knowledge identification in SALT

Previously, SALT had support for claim identification, but did not take into account the issue of multiple representations. Also, from the document engineering point of view, SALT was lacking a proper solution for identifying a particular text chunk in the resulting semantic PDF document. In the following, we describe the improvements that we brought both to the semantic and syntactic layer of SALT, in order to introduce easy and useful support for claim referencing.
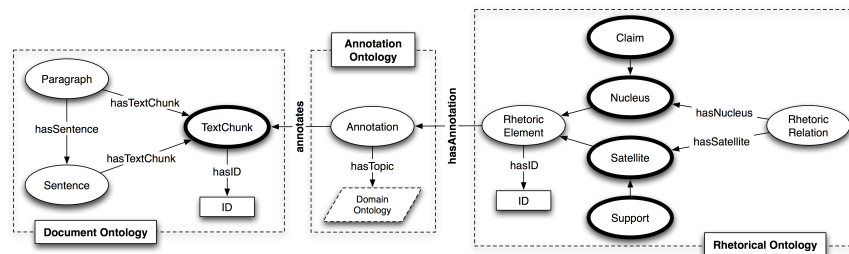
3.1 Improved Identification Support

Fig. 2. Improved SALT semantic layer to support better identification

In our previous work, each rhetorical element had an identifier, introduced manually by the author, during the writing process. At the document level, the elements were represented by a sentence which was entirely duplicated in the ontology. The relation between the rhetorical element and its textual representation was realized by an annotation which also offered support for denoting the actual part of the sentence representing rhetorical element.

---

[1] RFC3174: `http://www.faqs.org/rfcs/rfc3174.html`

With respect to claim identification, the modifications we introduced affect the Rhetorical and Document ontologies[2], as sketched in Fig. 2 (the newly introduced concepts are depicted with a thicker line). Conceptually, a rhetorical element still represents an annotation of a text chunk, but the way in which we model this relation has changed:
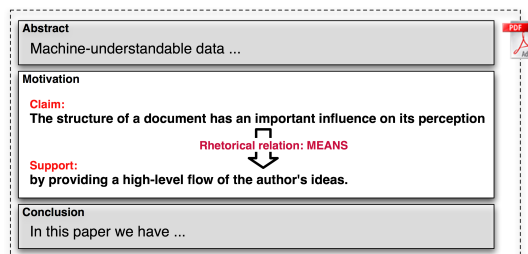

Fig. 3. Example of rhetorical elements in a scientific document

Rhetorical Ontology — Previously, a *Rhetorical Element* was represented by a *Claim*, or by an *Explanation* and had its own identifier. The ontology contained concepts for defining rhetorical relations between claims and explanations in the publication. It also contained the support for defining rhetorical blocks, such as *Abstract*, *Motivation* and *Conclusion*, and link the rhetorical elements and relations to them. We extended the ontology by adding two more concepts (*Nucleus* and *Satellite*). The *Nucleus* represents a generalization of a *Claim*, whereas a *Satellite* is the generalization of a *Support* (the renamed *Explanation*). As an example, given the *Claim* specified in Sect. 1 *[The document structure . . .]*, a possible *Support* is: *. . . by providing a high-level flow of the author's ideas.* The introduction of these concepts allows us to model the publication at a more fine-grained level: text chunks can now be annotated as nuclei or satellites, and the most important nuclei can be marked as being claims. This gives us the opportunity to build complete rhetorical structure trees for the annotated publication, and emphasize the place of the claims in these trees, in parallel with the existing support for rhetorical claim trees. An example for the use of some of the elements of this ontology is shown in Fig. 3.

Document Ontology — The main improvements for identification, were introduced in this ontology. Where previously the *Sentence* was the finest annotation granularity, we now introduced the *TextChunk* concept. This can be instantiated as part of a *Sentence* or of a *Paragraph*, and the major change is that it will not contain the annotated text, but a pointer in the document

---

[2] We also introduced a means to connect annotations to external domain ontologies, but this has no implications on the handling of claims.

for it. This pointer will serve as a local identifier at the document level, as described in Sect. 2.

3.2 Identification via Document Engineering

One of the major issues that we encountered in the discussion section in [1] was a proper mechanism for identifying a chunk of text inside a PDF document. By addressing this problem, we are now able to determine the exact position of the piece of text inside a PDF document and use this pointer as a local document identifier. In the following we describe the reasons why this represents a real issue.
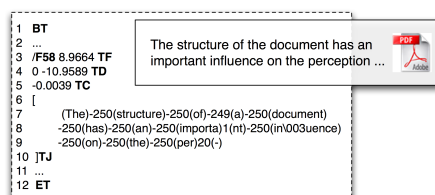


```
1  BT
2  ...
3  /F58 8.9664 TF
4  0 -10.9589 TD
5  -0.0039 TC
6  [
7      (The)-250(structure)-250(of)-249(a)-250(document)
8      -250(has)-250(an)-250(importa)1(nt)-250(in\003uence)
9      -250(on)-250(the)-250(per)20(-)
10 ]TJ
11 ...
12 ET
```

The structure of the document has an important influence on the perception ...

Fig. 4. Example of text encoded in PDF format

The PDF file format [4] allows representation of documents independently of the software, hardware or operating system they are created on, and of the device (display, printer) used for output. A PDF document consists of a collection of objects that describe the appearance of one or more pages, and can be accompanied by interactive elements and high-level application data. A PDF file contains the objects that make up the document and associated structural information, in the form of a self-contained sequence of bytes. Objects are incrementally added to the end of the file as the document is edited. The document pages can contain any combination of text, graphics and images grouped into articles. The appearance of a page is described by a content stream (as shown for example in Fig. 4), which contains the sequence of graphics objects that are to appear on the page. The layout and formatting is fully specified in the content stream, and it is common for the stream to be encrypted, either for security or for space-saving purposes.

Fig. 4 B. shows the encoding of a part of a page in a PDF document. The entire page is encoded between the BT and ET tags [3], and every line is introduced by a TJ operator and placed between squared brackets. Before the actual content, some other parameters can be set, like the font (in this case F58 or alignment and spacing corrections, by applying the TD operator (which moves the text a little bit down in this case — line 4) or the TC operator (which specifies the spacing

---

[3] The PDF tags are conforming with the PDF 1.6 Reference document.

between words with respect to the standard setting — line 5). The real text to be analyzed is separated depending on the relative space between the individual glyphs. If the space remains the same, a sequence of glyphs is represented within the same text block between parentheses. If the distance between the glyphs is not standard, it is specified by a value written between glyphs, as shown in lines 7–9.

As mentioned in Sect. 2 the symbol's identifier is represented by a pointer in the PDF document. The identifier presented in the example has two parts, delimited by a hash (#). Based on the internal organization of the PDF document, the two parts represent the beginning and the end of the text, as follows: pagecode — article in page — stream inside article — position in the stream. The first part has attached at the end the length of the text.

3.3 Claim citation and reference

In the following we use the example introduced in Sect. 2 to describe the improvements that we brought to SALT at the syntactical level, to support the referencing and storing of claims.
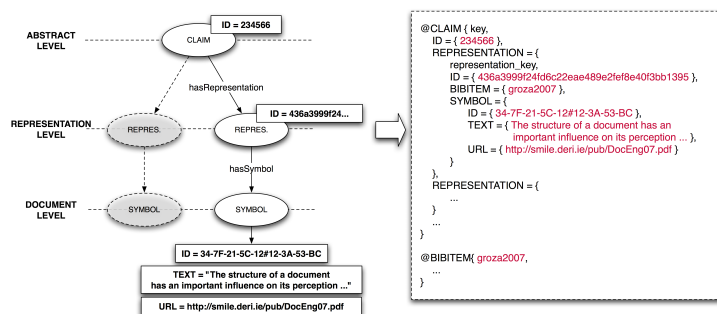

Fig. 5. Example of transformation of a claim identification tree into a CBIB item

In our vision of a Web of documents, an author will want to refer to and cite claims, not whole publications. Once the author finds the claim to be cited or referred, they need support for compiling a list of claim references and syntactical support for referencing them. For the first issue, we propose a solution similar to the current approach for storing bibliography items, i.e. defining *claim bibliography items* and storing them in format similar to the BibTeXfiles. Each claim bibliography item (CBIB item) can have a key assigned for easier referencing. An example of such a claim bib item in our proposed format and based on the example introduced earlier can be seen in Fig. 5. Claim items and bib items can me mixed in the same file, as shown on the figure.

To support referencing of claims in a LaTeX document, we introduced the \claimref command, which functions exactly like the usual \cite command, only

that it refers to claim items, not bib items. To provide a maximum of usability, the author can use the command not only to cite a claim, but also to cite a particular representation of the claim. This is the reason why each of the representations has its own key, as shown in Fig. 5.

4   Prototypical implementation of a claim search engine

In the previous section, we described how an author can create and reference claims, based on the SALT syntactical support and the claim bibliography format. We imagine that, when an author writes a piece of text that represents a claim, they will use a claim lookup service to find out if this claim already exists. Based on the result of the lookup, they will either create a new claim or reference an existing one. Similarly, when they want to reference a particular claim representation, e.g. made by another author, they will need a means to search for and find it.
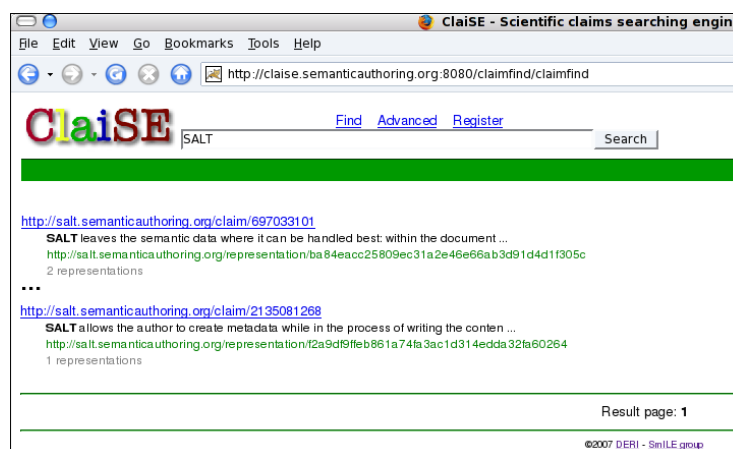

Fig. 6. Example of the claim search engine's results page

To make this possible, we designed and implemented a claim registration and search engine (Fig. 6 depicts the results page in the case of searching for the word $SALT$)[4]. The engine fulfills a series of requirements, which we considered important and extracted from relevant literature in the domain:

Distribution, persistence and un-ambiguity of identifiers. In order to ensure persistence and uniqueness for identifiers, someone must take the responsibility of managing them [5]. Our system takes the responsibility of managing identifiers for claims, representations and symbols. It provides a unique identifier

---

[4] The prototype can be found at `http://claise.semanticauthoring.org:8080/claimfind`

for each claim and takes care of creating the appropriate relations between the claim and the representations specified by authors.

Retrievability. The result of dereferencing a URI. [6–8] In our case, if the URI represents a claim, the user receives information about the representations, whereas if the URI denotes a representation, the user received information about the symbol and the claim.

Clear determination. [9] The identification schema, as depicted in Fig. 1 shows that our solution determines the concept that the URI is identifying and its place in the hierarchy.

URI ownership. [10] The user creating the URI is its owner.

The architecture of our system is strictly split into two main functionalities, which we detail as follows:

Registration represents the process through which the system acquires new knowledge by extracting metadata from publications. We provide two options of registering claims: (i) single publication, by providing the URL of the publication, and (ii) online proceedings, by providing the URL which contains all the publications accepted at that event. In other words, claims are registered by registering the publications that contain them. Due to the fact that the metadata is generated during the authoring process, the registration involves only the introduction of URL pointing to publications, whereas the metadata extraction and claim processing are performed automatically by the system. In the case of the online proceedings, the system extracts the metadata by iterating over all publications.

Searching allows a user to retrieve the URI and representation information of a claim, given a textual query. Alternatively, a textual representation can be retrieved by providing the URI. The real challenge will represent the design and implementation of a component to deal with word sense disambiguation and other related NLP techniques in order to return the closest match for a textual representation of a claim. The current prototype offers limited keyword based search.

5 Evaluation

We performed a small-scale evaluation of the SALT framework with a group of eight researchers, who all had knowledge of Semantic Web and Semantic Annotations technologies. Each participant annotated one of their own papers and was given a set of explanations of the model and instructions on how to perform the annotations beforehand. The evaluation consisted of seven annotations tasks as follows: (i) *Rhetorical block annotation*, (ii) *Claims markup*, (iii) *Support markup*, (iv) *Annotation of rhetorical relations between claims and supports*, (v) *Nuclei and sattelites markup*, (vi) *Annotation of rhetorical relations between nuclei and sattelites*, and (vii) *Annotation of domain knowledge*.

With respect to these seven annotation tasks, we asked the participants about (i) their complexity, (ii) the plausibility and soundness of the underlying model
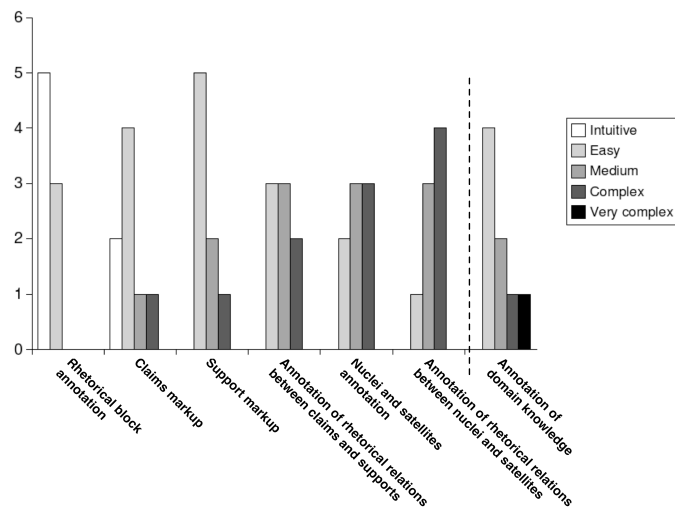
Fig. 7. Annotation tasks evaluation

and (iii) their perceived benefit. The results for each of those areas is shown in Figs. 7 and 8 A and B. The questionnaire and results can be found online[5]. In the following, we will elaborate our findings and discuss what can be deduced from them.
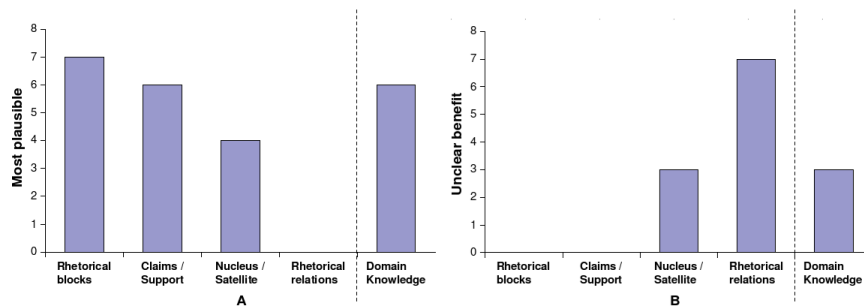

Fig. 8. A. Soundness evaluation; **B.** Benefit uncertainty evaluation

5.1 Findings

The target of the questions in the evaluation was to find out the researchers' impression on how intuitive the framework is, whether it is well adapted to sci-

---

[5] http://salt.semanticauthoring.org/evaluation/2007/05

entific writing or if it introduces a significant overhead in the writing process, and how plausible the overall model is perceived. In general, we observed that *with increasing complexity of the annotation task at hand, the participants' overall impression decreased*. The annotation of rhetorical blocks, which is also the simplest of the tasks, was deemed most intuitive, while the annotation of rhetorical relations at the other end of the scale was deemed not intuitive, but instead rather complex by most participants (see Fig. 7). Similarly, Fig. 8 A shows that the plausibility of terminology and soundness of the model were considered high by 7 out of 8 participants, whereas fewer and fewer participants said this about the more complex annotation tasks (6 found annotation of claims and support plausible, 4 found nucleus/satellite annotation plausible, while no one found the handling of rhetorical relations plausible). Figure 8 B supports the same tendency from yet another angle: none of the participants found the benefit of simple annotations (rhetorical blocks and claim/support) unclear, while 3 were unsure about the benefit of nucleus/satellite annotation and 7 about rhetorical relations.

This general tendency was expected — it seems intuitive that simpler annotations are easier to grasp, and as a result their the usefulness is much more apparent. As a consequence, we see the development of tools which aid the user in the authoring process as crucial.

The annotation of domain knowledge within SALT, which is not on the scale of complexity of the other annotations, but rather a seperate entity, was well received by most participants (4 found it easy, 2 said the difficulty was medium, and only 2 people found it complex or very complex to handle). Similarly, most people (6) found the supporting infrastructure plausible, and only 3 people had difficulty in seeing a benefit in the annotation of domain knowledge.

Apart from the findings which were directly taken from the questionnaire, we also observed a number of other things during the evaluation: most of the participants needed to revisit the documentation numerous times during the annotation process, and some had in fact admitted that they had not read the documentation at all before the evaluation. These observations fit very well with the general tendency we noted: users, even if they come from a technical background like the ones in the evaluation, tend to be lazy. Good documentation is useful, but we cannot make the assumption that is will actually be read thoroughly. Instead, a platform such as SALT should hide most of its complexity and guide the user with an interface that can be understood intuitively.

6 Related work

The ontologies which are part of our framework have their roots in the Rhetorical Structure of Text (RST) Theory [11]. The paper provides the underlying semantics of the concepts modelled by the theory together with their definitions. A second publication by the same authors [12] 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 com-

putational linguistics, cross-linguistic studies and dialogue to multimedia presentations. There were several attempts to ontologize RST, and one of the first ones, including also some implications based on temporal relations is presented in [13].

Most of the similar ontologies present in the domain, capture only a part of the RST foundation, or use parallel theories as input. One of the closest to our research is presented in [14]. The authors describe the 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. A similar approach is presented by Tempich et. al in [15]. The DILIGENT *Argumentation Ontology* was designed in line with the terminology proposed by the IBIS methodology [16] 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 therefore is less expressive.

None of the above mentioned approaches combines the logical and rhetorical structure of the text, in the way SALT does. This direction was mostly pursued in the NLG (Natural Language Generation) field. Here, however, the analysis performed is strictly based on NLP (Natural Language Processing) techniques. For example, Bouayad-Agha et. al analyze in [17] if the logical structure of the text can be incompatible with RST. The same authors later provide in [18] a deep analysis on how can the document structure help in the language generation process. Other approaches following the same direction but more focused on a particular domain are the ones presented in [19] and [20].

In terms of applications, we found the Compendium methodology [6] and the set of sensemaking tools described by Uren et. al [21] as the most interesting one in terms of similarity with our research. Their set contains tools for creating, visualizing and searching claims in scholarly documents (represented as HTML files) using a central knowledge server. Although very close to our methodology, the focus is different. We focus on the publication as an entity and on the methods for referencing claims directly, by using the metadata embedded in the publication. In regards to the claim searching engine, we used the centralized approach only for storing the minimum information about the claims and thus creating a distributed knowledge network rather than a central pool of publications.

Another interesting approach is the one of Peter et al. [22]. Their goal is to extract semantics from a LaTeX document 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 towards the rhetorical structure of the text and the semantic links between claims placed in different documents.

---

[6] http://www.compendiuminstitute.org/

7 Conclusion and future steps

In this paper we described a solution for "weaving the claim web", i.e. the creation of knowledge networks via the claims stated in scientific publications created with the SALT (Semantically Annotated LaTeX) semantic authoring framework. Like resources on the Semantic Web, each of the claims is addressable by a unique identifier.

We provide support for claim identification, evolved the appropriate ontologies and defined a claim citation and reference mechanism. We did not only describe how an author can create and reference claims, but we also implemented a prototypical claim search engine, which allows to find references to existing claims and hence, weave the web. Finally we performed a small-scale evaluation of the authoring framework with a quite promising outcome. Since the authoring process is highly incremental, we got the expected result that simpler annotations are easier to grasp, and as a result their the usefulness is much more apparent. As a consequence, we see the development of tools which aid the user in the authoring process as crucial. Those tools will actively support the author in marking up their document, and will visualize them at the same time. We believe e.g. that authors will feel much more comfortable with the rhetorical relations if they can *see* a graph of those relations that is built up incrementally, while they are authoring them. This will give the direct benefit of being able to see their documents from a birds-eye view. Similarly, tools and applications that make use of those complex annotations *after* the document has been published will be needed to fully convince authors of their usefulness. We envision e.g. that the claim search engine could be enhanced by functionality that allows a user to view a summaries of documents based on their nuclei.

We should also point out SALT does not confront the user with an either-or situation: it is not required to perform annotations at *all* levels. Partial annotations are not a problem, and each type of annotation will give a different benefit by itself. Rhetorical blocks and relations, as well as the marking up of nuclei and satellites will allow new views on a document and aid both readers in understanding it, as well as a authors in checking their own argumentative structure as they write. Annotating claims and their representations, as well as linking to domain ontologies, will result in the weaving of a fine-grained web of references between documents, and offer new ways of searching and finding relevant literature in an otherwise overwhelming mass of publications.

References

1. Groza, T., Handschuh, S., Möller, K., Decker, S.: SALT - Semantically Annotated LaTeX for Scientific Publications. In: ESWC 2007, Innsbruck, Austria (2007)

2. Mitchell, J., Aronson, A., Mork, J., Folk, L., Humphrey, S., Ward, J.: Gene indexing: Characterization and analysis of NLM's GeneRIFs. In: Proc AMIA Symp. (2003) 460—464

3. Ogden, C.K., Richards, I.A.: The Meaning of Meaning: A Study of the Influence of Language upon Thought and of the Science of Symbolism. Magdalene College, University of Cambridge (1923)

4. Incorporated, A.S.: PDF reference - Adobe portable document format. http://partners.adobe.com/public/developer/en/pdf/PDFReference16.pdf (2004)

5. Walsh, N.: Names and addresses. http://norman.walsh.name/2006/07/25/ namesAndAddresses (2006)

6. Parsia, B., Patel-Schneider, P.: Meaning and the semantic web. In: "Identity, Reference, and the Web" Workshop, WWW 2006, Edinburgh, Scotland. (2006)

7. Clark, T., Martin, S., Liefeld, T.: Globally distributed object identification for biological knowledgebases. In: "Identity, Reference, and the Web" Workshop, WWW 2006, Edinburgh, Scotland. (2006)

8. Lawrence, S., Pennock, D.M., Flake, G.W., Krovetz, R., Coetzee, F.M., Glover, E., Nielsen, F., Kruger, A., Giles, C.L.: Persistence of web references in scientific research. Computer **34**(2) (2001) 26–31

9. Ginsberg, A.: The big schema of things. In: "Identity, Reference, and the Web" Workshop, WWW 2006, Edinburgh, Scotland. (2006)

10. Pepper, S.: Towards the semantic superhighway: A manifesto for published subjects. In: "Identity, Reference, and the Web" Workshop, WWW 2006, Edinburgh, Scotland. (2006)

11. Taboada, M., Mann, W.C.: Rhetorical structure theory: looking back and moving ahead. Discourse Studies **8, No. 3** (2006) 423–459

12. Taboada, M., Mann, W.C.: Applications of rhetorical structure theory. Discourse Studies **8, No. 4** (2006) 567–588

13. Nicholas, N.: Problems in the application of rhetorical structure theory to text generation. Master's thesis, The University of Melbourne (1994)

14. Mancini, C., Shum, S.J.B.: Modelling discourse in contested domains: a semiotic and cognitive framework. Int. J. Hum.-Comput. Stud. **64**(11) (2006) 1154–1171

15. Tempich, C., Pinto, H.S., Sure, Y., Staab, S.: An Argumentation Ontology for Distributed, Loosely-controlled and evolvInG Engineering processes of oNTologies (DILIGENT). In: ESWC 2005, Heraklion, Crete, Greece, May (2005) 241–256

16. Kunz, W., Rittel, H.: Issues as elements of information system. Working paper 131, Institute of Urban and Regional Development, University of California (1970)

17. Bouayad-Agha, N., Power, R., Scott, D.: Can text structure be incompatible with rhetorical structure? In: INLG2000, Mitzpe Ramon, Israel (2000) 194–200

18. Power, R., Scott, D., Bouayad-Agha, N.: Document structure. Computational Linguistics **29** (2003) 211 – 260

19. Linden, K.V., Martin, J.H.: Expressing rhetorical relations in instructional text: A case study of the purposes relation. Computational Linguistics **21**(1) (1995) 29–57

20. Silva, N.D., Henderson, P.: Narrative support for technical documents: Formalising rhetorical structure theory. In: ICEIS, Miami, FL, USA (2005)

21. 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

22. Peter, H., Sack, H., Beckstein, C.: Document indexing - providing a basis for semantic document annotation. In: XML-Tage 2006, Berlin (2006)