Visualisation of Semantic Enrichment

Alexa Schlegel[1], Ralf Heese[1], Annika Hinze[2]

[1]Freie Universitaet Berlin, Germany
aschle, heese @inf.fu-berlin.de
[2]University of Waikato, New Zealand
hinze@cs.waikato.ac.nz

Abstract: Automatically creating semantic enrichments for text may lead to annotations that allow for excellent recall but poor precision. Manual enrichment is potentially more targeted, leading to greater precision. We aim to support non-experts in manually enriching texts with semantic annotations. Neither the visualisation of semantic enrichment nor the process of manually enriching texts has been evaluated before. This paper presents the results of our user study on visualisation of text enrichment during the annotation process. We performed extensive analysis of work related to the visualisation of semantic annotations. In a prototype implementation, we then explored two layout alternatives for visualising semantic annotations and their linkage to the text atoms. Here we summarise and discuss our results and their design implications for tools creating semantic annotations.

1 Introduction

With semantic technologies, annotations are no longer *about* the content (as in Web 2.0 tagging) but become *part of the content*. Such semantic enrichment typically consists of an annotated text passage (text atom) and related information (annotation). Clear visualisation of enrichment, that is, indication of text atom and linkage to annotation, is important for both the definition of annotations and the reading of enriched text. Furthermore, it is essential for the acceptance of semantic technologies that non-experts are enabled to produce and consume semantically enriched content.

Here we focus on the visualisation of enrichment during the annotation process. To the best of our knowledge, neither the visualisation of semantic enrichment nor the process of manually enriching texts has been evaluated before. Additionally, research on presentation of semantically annotated documents typically targets the rather passive reception aspects of data visualisation. Our research is motivated by experiences in two projects, TIP and loomp, addressing the annotation of content by non-experts. TIP is a mobile tourist information system that provides different information depending on a user' interests [Hin09]. Textual information in TIP has to undergo a semantic enrichment process to be prepared for this interest-based filtering [Hsi08]. Loomp is a tool for the management of semantic enrichment, which we applied to texts (predominantly) relating to museums and their exhibitions [Luc09]. In loomp, users enrich texts semantically by linking text passages to concepts, thus forming annotations with additional, structured information. While atoms typically consist of only a few words in loomp, in TIP, users assign categories to longer text passages. In both cases, the annotations may be used later for generating rich content and recommendations for tourist information systems. They open new ways of locating, accessing and reusing existing content as well as the generation of new semantic links.

In this paper, we report the results of our user study on visualisation of text enrichment during annotation. We implemented a system for light-weight semantic enrichment[1] that allowed users to create text annotations by referring to categories (e.g., history, architecture). We explored two layout alternatives for the visualisation of text atoms and their linkage to annotations along four characteristics of visual feedback. Here we discuss our results and design implications for tools creating semantic annotations.

The remainder of the paper is structured as follows: Section 2 discusses characteristics of annotations. In Section 3 we present the results of our analysis of current approaches for highlighting annotations. We then describe the methodology and set-up of our user study and the implementation of our highlighting approaches in Section 4. The results of our user study are discussed in Section 5. Finally, we summarize the contributions of this research and give an overview of future work in Section 6.

2 Characteristics of Annotations

The process of semantic enrichment links additional information into the main content of a text. We refer to an *atom* as a continuous portion of a text linked to an *annotation.* Atoms and annotations do not need to be represented in the same data format.

We here define a list of properties for characterizing (relationships between) annotations. All of these characteristics directly influence the visualization of atoms and annotations.

Cardinality. We differentiate between 1:1, 1:n, n:1, and n:m relationships between atoms and annotations. For example, n:1 means that an arbitrary number of atoms may refer to the same annotation. Commenting in word processors result mainly in 1:1 or 1:n relationships. Semantic annotations often have n:1 relationships.

Granularity. An atom could be defined as character, word, phrase, sentence, or whole document. If a document is the smallest atom, annotations are referred to as metadata.

Positioning. We distinguish between overlapping and adjacent atoms (cf. Fig. 1). Overlapping annotations additionally may be the special cases of inclusion or identity.

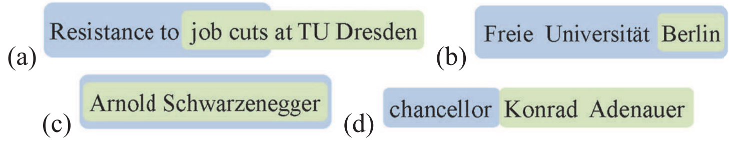

Figure 1: Positioning of annotations: (a) Overlapping, (b) inclusion, (c) identity, and (d) adjacent

[1] Available at https://github.com/aschle/OverlappingAnnotations

Highlighting atoms and annotations. Typical approaches to visually distinguish atoms within a text are modification of text styles (e.g., underline or bold), change of background colour, and use of graphical elements (e.g., boxes or icons).

Position of annotations. Options for positioning annotations are as an overlay near the corresponding atom (sometimes freely movable), in the left or right margin of the text next to the atom, or below the document.

Visualising overlapping atoms. To indicate overlapping atoms, different strategies are employed: mix of colours, stack view, stripes, and vertical lines in the margin.

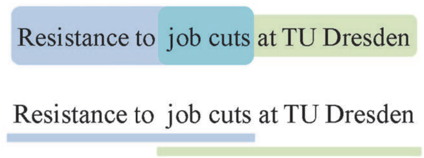

Figure 2: Overlapping annotations: mixture of colours and stack view

Connections. To indicate the relationship between atoms and annotations, the same background colour may be assigned to both atoms and annotations, annotations may be positioned nearby or mouse-over effects are used.

3 Related Work

We evaluated work related to the visualisation of semantic annotations. Not much research has been performed in this area and most available annotation tools are available on the Web for annotating ebooks or other texts. We used the characteristics introduced in Section 2 to analyse the visualisation supported by available tools.

The summarised results of our evaluation are listed in Figure 3. In the table in Figure 3, the supported alternatives between options in a characteristic are indicated by an x (otherwise left blank). For overlapping annotations, a system's visualisation may not support all four options. Each option that is fully supported is marked with +; o indicates that a visualisation is supported but has design limitations, and – indicates that although this type of overlapping may occur in the text, the software does not provide a specific visualization.

We now discuss each of the analysed systems and tools in turn. Some support both manual and automatic annotations (e.g., Gate, OpenCalais, rdfquery), others provide options for manual annotation only. Most of the systems analysed support free-text annotation, some support categories (as done in our approach).

Property	*Booktate*	*A.nnotate*	*Crocodoc*	*diigo*	*Bible+*	*rdfquery*	*GATE*	*Atlas.ti*	*veeeb*	*OpenCalais*	*TIP*	*loomp*
Cardinality												
$1:1$		×										
$1:n$	×		×	×	×							
$n:1$						×						×
$n:m$							×	×	×	×	×	
Granularity												
character			×	×	×	×	×	×			×	×
word	×	×							×	×		
Overlapping types												
overlap	+	○	○	−	+	−	+	+	+	○	+	−
inclusion	+	○	○	−	+	○	+	+	+	○	+	−
identity	+	−	○	−	+	○	+	+	○	○	+	−
adjacent	○	○	○	+	+	+	○	+	+	○	○	○
Position of annotations												
overlay		×		×	×						×	
left margin						×				×		
right margin	×	×	×				×	×			×	×
top margin									×			
bottom margin		×					×					
Highlighting atoms												
font style	×				×				×	×		
background color	×	×	×	×	×		×	×	×	×	×	×
graphical elements				×	×	×						
Visualising overlapping atoms												
color mixing	×				×		×		×			
horizontal lines (stack view)							×					
vertical lines								×				
mixing font styles	×				×					×		
patterns in background color			×								×	
numbers (icons)				×								
Connections												
connection lines			×									
identical colors	×						×	×		×	×	×
nearby positioning	×	×	×									
mouse effects (interaction)	×	×	×	×	×		×	×	×	×		×

Figure 3: Tools for creating annotations and their characteristics

Booktate (www.booktate.com) is a system for annotating eBooks. An annotation is typically set at word level; however, annotations on section level are also possible. Users can create several annotations for a single atom (1:n relationships between atoms and annotations). Atoms are highlighted by assigning a background colour or by underlining. An annotation is placed in the margin directly beside the paragraph containing the corresponding atoms and has the same background colour as the atom. If there is not enough space for the annotations next to a paragraph, a larger empty space may be created between two paragraphs. To visualize the overlapping of two atoms a subtractive mixture of colours is used (similar to Figure 2, top). If users hover with their mouse over such an annotation then the original colour is restored.

A.nnotate (a.nnotate.com) is a Web application for annotating PDF files. Example activities that are supported are creating of comments, striking text, and inserting characters. The background colour of text atoms can be changed to one of seven colours. Users can select one of three visualizations for annotations, overlay, right margin, and bottom margin (overlay and margin layout shown in Figure 4) and also choose their background colour. Depending on the visualization, the link between atom and annotation is indicated differently. In case of an overlay the application places annotations near the atom over the text. In the other cases it uses the annotated text as title of the annotations. If the annotation is placed on the right margin then it is position at the same height as the atom. If it is located at the bottom then the atom is shown within its context. The order of annotation corresponds to the order of atoms. The application supports all types of overlapping annotations except identity. However, later annotations may cover earlier ones leading to problems accessing older annotations. If users assign the same or no background colour to the atoms, they would not be able to distinguish overlapping atoms without selecting one of them (i.e., the system does not supports a special strategy for overlapping annotations).

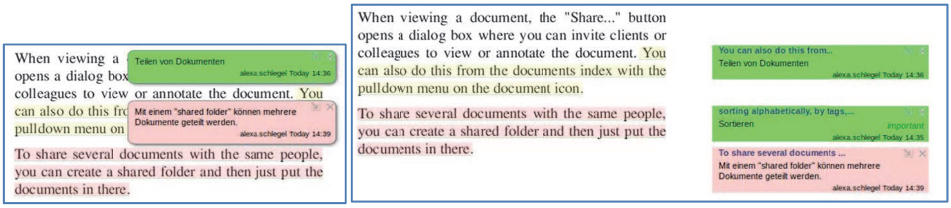

Figure 4: A.nnotate overlay options

Crocodoc (crocodoc.com) is a Web application that converts office documents and PDF into HTML5 format and supports annotation of these documents. Users can select one of four background colours for highlighting atoms. Annotations appear in the right-hand margin at the same height as the corresponding atom. Additionally, the system draws a line between atom and its annotation. If users hover with the mouse pointer over an annotation the corresponding atom is indicated and the annotation is additionally shown as an overlay near the atom.

Although users can select the background colour of an atom, all annotations have the same background colour (see Figure 5). The system supports two kinds of annotations:

highlight and comment. Overlapping annotations are supported, but older annotated atoms are covered by newer ones and the overlapping part of two atoms cannot be selected. In case of identity, none of the overlapping atoms can be selected.

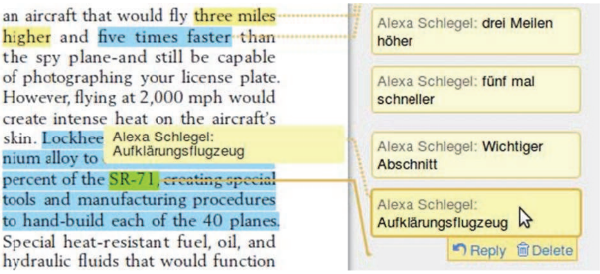

Figure 5: Crocodoc annotations

Using *diigo* (www.diigo.com) users can add annotations to web-pages. Similar to Crocodoc, users can select one of four background colours for highlighting atoms. The application allows users to add comments to atoms. The presence of such an annotation is indicated by a speech bubble showing the number of comments linked to an atom. When opening a document, all annotations are initially hidden and are only displayed as a pop-up on mouse-over. Only identical overlapping annotations are supported.

Bible+ is an iphone app for Bible reading. Users can take notes, highlight, and bookmark passages as well as search in the annotated text. Annotations can be synchronised between different texts and different devices. Between the analysed systems, Bible+ offers most versatile support of overlapping annotations (see Figure 6).

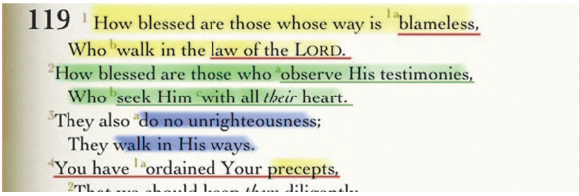

Figure 6: Bible+ overlapping annotations (here on ipad)

rdfquery (code.google.com/p/rdfquery) is a JavaScript library for parsing, querying, and generating RDFa elements. It uses frames for highlighting atoms in the text and displays corresponding annotations (e.g., facts about named entities) in the left-hand margin. Independently of their semantics, all frames have the same colour. The library supports inclusions as overlapping annotations; adjacent atoms are well distinguishable.

Gate (gate.ac.uk) is a toolkit for analyzing and processing texts. Annotation types (e.g., categories) are displayed in the right-hand margin (see area (1) in Figure 7); atoms are highlighted in the same background colour as the corresponding annotation. All types of overlapping atoms are supported using a mixture of their background colours. Additionally, users can open a stack view showing the atoms as horizontal bars having the corresponding background colour. In case of overlapping atoms the bars are on different levels (see (2) and (3) in Figure 7).

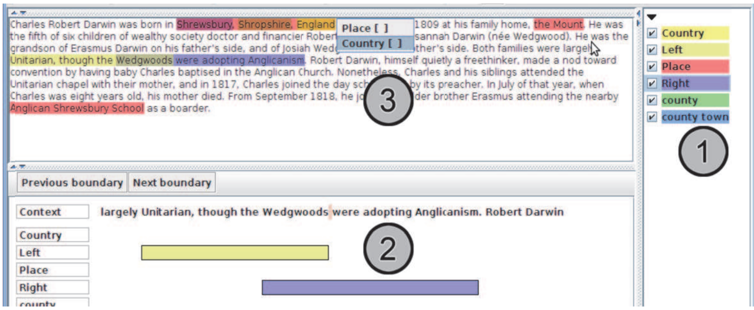

Figure 7: Gate annotations including the stack view

Atlas.ti (www.atlasti.com) is a tool for evaluating textual data conducted in social science [Muh94]. The authors identify the support for both annotations on character level and an arbitrary number of overlapping annotations as the main features of their software. In contrast to other annotation software, atoms are not highlighted in the text but are indicated by vertical bars in the right-hand margin. In the margin, the system also shows the assigned annotations and displays the content of an atom as an overlay on mouse-over nearby the corresponding bar. The related atom is highlighted in the text only when users select a bar. The bars are arranged into columns, using one column for each available colour.

veeeb (www.veeeb.com) is a Web-based tool for analyzing texts and generating semantic annotations. All recognized entities are highlighted using the same orange colour. The tool implements a special technique for indicating overlapping atoms: the more atoms overlap the darker is the orange colour (see Figure 8).

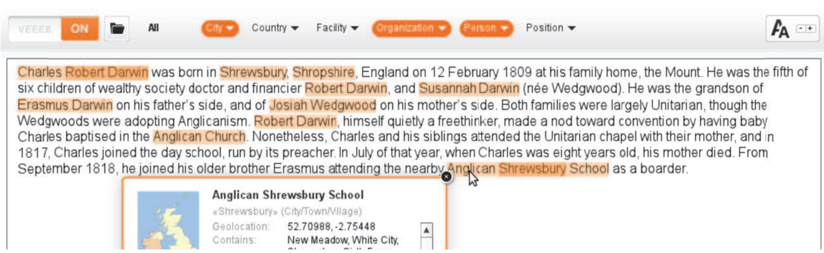

Figure 8: veeeb annotations

OpenCalais (viewer.opencalais.com) is a service for analyzing and enriching texts. The OpenCalais viewer supports only two types of annotations: “entities” and “events & facts”. The atoms are underlined or assigned a background colour, respectively. The system assigns different colours to different annotations. Overlapping atoms can only be clearly identified between atoms of different types. For atoms of the same type, users need to hover the mouse over an atom to see the complete atom.

TIP is a mobile tourist information system [Hin09]. Its information import service supports users in the semantic mark-up of texts. The text snippets within each annotation category are then stored separately to be available should a user be interested in tourist information in this category. In the import service, overlapping annotations are indicated by background patterns (see Figure 9). Because the use of the final annotation is no longer linked to the complete text, TIP avoids some of the display issues faced by other tools. The interface was evaluated in a simple paper prototype study.

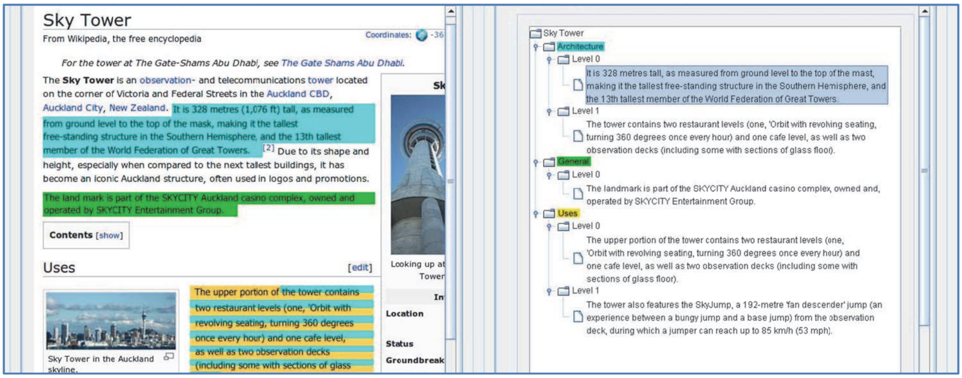

Figure 9: TIP import service – text annotations (left) and concept structure (right)

loomp is a Web-based editor for creating semantic annotations in texts [Luc09]. It uses a single background colour for highlighting annotations and annotations are displayed on the right margin. Overlapping atoms are currently not supported (except adjacent ones).

Summary

We examined several tools for annotating texts. Almost all tools highlight atoms by assigning a background colour. Only few change the font style or add graphical elements (e.g., icons). In contrast, no common approach for indicating overlapping atoms could be identified; the most frequent techniques are mixed background colours and mixed font styles. However, only few tools provide a clear visualization of overlapping atoms. The main problem is distinguishing overlapping atoms of the same category of annotation (as they typically use the same style).

The examined tools typically apply similar visualizations to annotation and corresponding atoms. Additionally, a mouse-over effect may highlight corresponding annotations and atoms. All tools position annotations near the related atoms (where possible). To the best of our knowledge, none of the tools and annotation interfaces has been evaluated for their ease of use (beyond a simple study reported in [Hsi08].)

4 Implementation and Study Setup

The implemented system was purpose-built to explore two layout alternatives for the visualisation of text atoms and their linkage to annotations. The following design features were included to make the system suitable for annotations of tourism-related texts such as the ones used in TIP and loomp. The main usage scenario is annotation of texts (not multi-media objects), where annotations may span over a few words or some lines. The system needs support for categories, that is, be able to explicitly distinguish annotation sets (such as expressed by different ontologies). Only a limited number of categories had to be considered; however, each category may have several (up to ten) subcategories.

Figure 10: Bar layout

Figure 11: Border Layout

We explored the two alternatives of bar layout and border layout. Both layouts were implemented as simple prototypes using HTML and JavaScript. In the bar layout, each atom within the text is indicated by a vertical bar in the left margin (Figure 10). The colour of the bar reflects the annotation concept (e.g., orange=architecture, purple=history). The bars are ordered by length and order in the text. Atoms in the text are highlighted by a mouse-over of the corresponding bar and the annotations appear as a speech bubble near the atom. The border layout highlights annotations by enclosing an atom in a coloured frame (Figure 11), where the colour corresponds to the selected category of the annotation. The background colour of an atom changes on mouse-over and the annotation appears as a speech bubble. Both layouts allow for many-to-many relationships between atoms and annotations, and for atoms to span several lines. Atoms may overlap or be adjacent (see Section 2). The number of atoms overlapping the same portion of text was restricted to three and the number of categories to four.

Semantic annotations are meant to be created by non-experts with respect to semantic technologies. We therefore observed 12 non-exert participants (P1 to P12) interacting with both interfaces (alternatively starting with bar or border layout). During a *learning phase*, participants familiarized themselves with the system using a short practice text. During the *application phase*, they had to execute a number of annotation tasks on a longer text. The participants were encouraged to think out loud as they were making decisions in interaction with the prototype, instead of asking for the 'correct' procedure. Each study concluded with a guided interview.

5 Study Results and Discussion

We here briefly summarise the main findings of our user study.

Atom definition: All 12 participants found it easy to select text atoms for annotation. Three participants noted that it was not possible to select letters or parts of words. In particular, P3 wished to select `Libeskind' (as part of Libeskind-Bau), and P5 aimed to annotate `Beton' (Engl: concrete) within `Betonstehlen' with category material. Restricting selections to whole words was supposed to make it easier to create meaningful atoms but in this cases prevented the creation of what would have been acceptable atoms. Two participants (P2, P10) wished to correlate atoms with each other (e.g., the name `Daniel Libeskind' and the profession `architect'). A similar desire to create cross-references between atoms was also observed in another study we performed on semantic annotations.

Layout and ordering of bars: 11 of 12 participants liked the bars' position in the left-hand margin (P5: "Needs to be definitely on the left-hand side!"). P4 suggested placing some bars to the left and others to the right of the text (depending on the position of the atom within each line). She additionally suggested placing bars left or right depending on length or category. All participants preferred the bars to be ordered by length. Seven suggested ordering largest to smallest (Figure 12, top), four suggested from smallest to largest (Figure 12, bottom); one was indecisive. People who preferred ordering largest to smallest argued that it would be easier to identify the lines of text (atoms) belonging to the smaller bars. The other group felt the design was clearer when the longer bars were close to the text.

Figure 12: Ordering of bars

Interaction with bars and borders: All participants interacted successfully with the bar annotations. During annotating, they used the mouse-over to identify which parts of the text were already annotated. P8 remarked: "I do not like that I always have to look to the left to find out what was already annotated and what not." Using the border layout four of the 12 participants felt it was very easy to identify the category of an annotation. Five thought it was not quite easy, as the atom's space may be very small and it is "hard to click the right border" (P9). P10 expected that he could extend an atom by dragging its borders. Three were undecided.

Clarity of layout: On a 5-point Likert scale, 2 participants completely agreed during the interview, 6 mainly agreed and 4 agreed partially with the statement "the bar layout is clearly arranged." (1/2/3 started with the bar layout; 1/4/1 with the border layout). On the equivalent question about the clarity of the border layout, 6 completely agreed, 4 mainly agreed and 2 partially agreed (4/1/1 started with bar layout; 2/3/1 with border layout). This indicates that participants seem to prefer the border layout.

During the study it was noted that overlapping annotations constitute a considerate proportion of all created annotations (used by 8 of 12; up to 30% of all annotations). In the guided interviews, we observed that the participants saw the bar layout to be more suitable for annotating larger text passages because many (small) bars on the left side potentially make the interface less clear. Participants also found that the bar layout was somewhat imprecise as atoms are only identified by line but not by position in each line. However, the bar layout was found to be well suited for reading and annotating since texts themselves do not contain any highlighting.

Participants found the border layout to be more suited for annotating short text passages because they could easily recognize the atoms, and the relationship between atoms and annotations was clear. However, participants noted that users may get confused by the borders if they are confronted with too many atoms.

6 Conclusion and Future Work

The success and rapid uptake of Web 2.0 concepts was largely due to and driven by the availability of applications for non-expert users (i.e., users with little knowledge about Web technologies). We believe that the success of the Semantic Web similarly depends on the availability of applications for non-expert users (i.e., users with little knowledge about semantic concepts and technologies). Many semantic web researchers focus on creating applications for producing and consuming semantically enriched content. However, only few ensure the usability of their user interfaces for the large group of non-expert users.

In this paper we present our analysis of visual tools for creating annotations and describe the results of an initial user study on the highlighting of annotations. The results of our study form a first step towards formulating recommendations and best-practice examples for the design of annotation systems with manual components.

The indication of overlapping annotations was identified as the main issue for visualisation of annotations. None of the tools and annotation interfaces had been previously evaluated for their ease of use. In our user study, two layout alternatives for the visualisation of text atoms and their linkage to annotations. Our user study confirmed that overlapping annotations constitute a considerate proportion of all created annotations. They were identified as part of a typical annotation process and should not be treated as special cases. The border layout supports clear identification of overlapping annotations, whereas their identification is more complicated in the bar layout. We also found that the bar layout is more suitable for annotating larger text passages whereas the border layout is more suitable for annotating words and short passages. We therefore recommend that systems should implement two views on annotated texts: One view for unhindered reading, a quick overview of the text and locating atoms and annotations at a glance (e.g., bar layout) and another one for creating annotations in the text and retrieving detailed information about the annotated text passages (e.g., border layout).

The work presented in this paper considered mainly the visualisation of (simple semantic) annotations (e.g., assigning a category). However, full semantic mark-up requires the additional assignment of semantic identifiers. The understanding of complex semantic annotations (e.g., assigning and interpreting the linkage to resources) by non-expert users is more complicated and needs to be explored further. Moreover, so far only annotations created by single users were analysed. The concurrent annotation of texts by a group of users (e.g., in a crowd-sourcing approach) will most likely lead to more overlapping and potentially contradicting annotations. Appropriate resolution of these cases still needs further research.

References

[Hin09] Annika Hinze, Agnès Voisard, George Buchanan: Tip: Personalizing Information Delivery in a Tourist Information System. Journal of IT & Tourism 11(3): 247-264, 2009

[Hsi08] Ping-Ju Hsieh. Administration Service for the Tourist Information System. Master's thesis, Computer Science Department, The University of Waikato, June 2008, available online at http://researchcommons.waikato.ac.nz/handle/10289/2478.

[Luc09] Markus Luczak-Rösch and Ralf Heese. Linked data authoring for non-experts. In Proceedings of the Linked Data on the Web Workshop (co-located to WWW'2009). LNCS, March 2009

[Muh94] Thomas Muhr. ATLAS.ti: Ein Werkzeug für die Textinterpretation. In Schriften zur Informationswissenschaft, pp 317-324. Univ.-Verlag Konstanz, 1994.