Deliverable 11.8

Working with living documents

Authors: Johannes Busse
Ontoprise
busse@ontoprise.de

Bernt Bremdal
Cognit
Bernt.Bremdal@cognit.no

Work-package: WP11
Type: Deliverable
Distribution: Public
Status: Version 1.0 (final)
Release Date: 2009-02-27
Deliverable Coordinator: Johannes Busse
Reviewer: Bernt Bremdal
Area Coordinator: Luca Gilardoni - Quinary Spa, Milan, Italy
Project Coordinator: Fabio Ciravegna, University of Sheffield
EU Project Officer: Dirk Van der Eecken

Abstract

D11.8 reports findings of the year 2008 w.r.t. knowledge management aspects of text annotations in X-Media. In order to learn more about the "semantics" and common world understandings of WYSIWYG text annotations (i.e. coloured highlighted regions) we describe our understanding of the process of annotating text in detail.

This deliverable primarily reports work related to task 11.8 "Working with ‘living’ documents". In this task we analyze how to relate the (rather dynamic) task of annotating of documents “@work” to the (rather static) task of annotating archived documents, i.e. documents which are in a stable state.

While in phase one of the X-Media project the X-Media annotation methodology was focusing rather on archived documents, the challenge of annotating living documents asks for clarifying some more complex and tangled challenges.

Note: The reviewer of the deliverable added a lot of important comments plus an in depth analysis of some arguments that it was decided (a) to include his contributions into the text and (b) to make him a co-author of the deliverable. We interpret this being a lucky outcome of a thorough review process.

Table of Content

1 Introductory Example

In the following introductory example we discuss in detail how a user might want (and might mean) by annotating a text.

The following text was gathered from the (html rendered) Wikipedia page http://en.wikipedia.org/wiki/Johannes_Brahms. It was transformed first to plain vanilla ascii by copying the rendered html text from firefox with `ctr-a ctrl-c` and dropping it with `ctrl-y` to to xemacs, thus removing all given xml markup completely. We then threw out most of the text (focusing mainly on the relationship between Brahms and Clara Schuman). The text then was transformed to OOo by copy-and-paste from xemacs to OOo. Afterwards the text was formatted by applying OOo paragraph and char formats.)

> *Johannes Brahms ... (May 7, 1833 – April 3, 1897), composer and pianist,* was one of the leading musicians of the Romantic period. Born in Hamburg, Brahms spent much of his professional life in Vienna, Austria, where he was a leader of the musical scene. ... Brahms composed for piano, for chamber ensembles, for symphony orchestra, and for voice and chorus. ...
>
> *He also worked with the leading performers of his time, including the virtuoso pianist Clara Schumann and the violinist Joseph Joachim. ...*
>
> *Joachim had given Brahms a letter of introduction to Robert Schumann*, and after a walking tour in the Rhineland Brahms took the train to Düsseldorf, and was welcomed into the Schumann family on arrival there. ...
>
> *While he was in Düsseldorf, Brahms participated with Schumann and Albert Dietrich in writing a sonata for Joachim; this is known as the F-A-E Sonata.* He became very attached to Schumann's wife, the composer and pianist Clara, fourteen years his senior, with whom he would carry on a lifelong, emotionally passionate, but probably platonic, relationship. Brahms never married, despite strong feelings for several women ...

In this text you can find some interesting annotation challenges:

Johannes Brahms ... (May 7, 1833 – April 3, 1897), composer and pianist, ...
The text starts with a normal language enumeration of classical CV data of a person. While these data are embedded here in prose text, we would expect to have such data typically available in a more formalized form, i.e. in a relational database or in a corporate semantic back with a controlled and standardized schema (like it is the case in X-Media). If you are using the Semantic Media Wiki (SMW) you probably would attach these data to a so called fact box, which either can be displayed on a page itself or in a form view onto a page. The point here is not so much that a text annotation component has the task to transform informal data in a more formal representation.

The interesting point is that we are dealing not with metadata but with genuine data itself: Giving the day and town of birth of a person is simply giving data.

He also worked with the virtuoso pianist Clara Schumann and the violinist Joseph Joachim. If you are interested in classical music you most probably know that Robert Schumann's wife was called Clara. But if you are not an enthusiast, you probably have not heard of Joseph Joachim already. If we assume that the author of this sentence assumes that we know Clara Schumann but not Joseph Joachim, we get different pragmatics of the annotations "the virtuoso pianist" and "the violinist".

In the case of Joseph the addendum "the violinist" is an information given to the reader in order to allow for disambiguating an ambiguous name into an unambiguous concept ID.

The case of Clara the addendum "the virtuoso pianist" has a completely different semantics. The author knows that we know that there is exactly one Clara Schumann given in our context. There is no need for giving disambiguation information. Even more: Such precise information like "the *virtuoso* pianist" can be interpreted as a disambiguation information only then if there are really many Clara Schumanns around, at least that many, that only knowing "the pianist" wouldn't be enough to yield a unique Clara Schumann. (In fact the Germany phone book of Telekom yields zero hits for this name.) What we have here instead is additional information about Clara – here: that Clara is not only one of many pianists (like Brahms?), but one of the very virtuoso and well known pianists of the 18th century. The difference is that – as opposed to Joseph Joachim – the name Clara Schumann is a nearly unique name.

- In the case of "the virtuoso pianist Clara Schumann" we probably might rephrase the article author's intention as "By the way: did you already know that Clara was a virtuoso pianist?" The.
- In the case of "the violinist Joseph Joachim" we probably should rephrase the meaning as "Joseph Joachim (you know: the violinist, not the author)"

What we are actually discussing here is the *intentionality* of the author which linguistically is different to the more objective aspects of semantics. We are pushing here the boundaries of semantics where the interpretation of meaning has to rely just as much on pragmatics. We strike here the discussion on how far one is going to extend the concept of semantics in annotation. Not many ontologists tend to address this area of intentionality. And even expert linguists probably would simply tell us to stay away from this field because of it’s complexity. Anyway, we believe there is much potential in considering this topic. In fact we are discussing here approaches the type of discussion commonly found in literature research where one tries to interpret the symbolisms, subtleties and the intentions behind a novel, a poem etc. Surprisingly flavours of such can also be found in our highly fact oriented literature. And we need to

Joachim had given Brahms a letter of introduction ... Clearly "Joachim" refers here to Joseph Joachim (you remember: the violinist).

It is quite common to manage such a coreference automatically by means of standard coreferencing algorithms in NLP, so we do not have to rely on the syntactical pattern alone. The system’s co-ref suggestions could be produced as a pop-up menu for the person that annotates.

Some sophisticated IE systems already are able to propose concepts, themes, collocations and expressions with a high degree of contextual affinity that could constitute nominations/candidates for annotation. Combined with a Amazon.com recommendation approach (collab filtering type of approach) whereby already tagged documents are exploited we could foresee that nominations that are made can also be associated with former instances of "word/phrases - annotation combinations". Hence the work of the user will be reduced to mostly a selection process. Unlike folksonomies this will also assure continuity and control. In the beginning, until the system is initially seeded, the user effort will be pretty manual. But documents within a company and a well known business domain that remains very constant (as it is also the case in X-Media) we can achieve a high degree of annotation reuse very rapidly due to high contextual homogenity and persistent purpose related to the business objectives of the company itself. Thus it should not take long to speed up the process and obtain reuse.

While one main objective of annotating documents in X-Media is to provide exactly this type of information it is unclear to which extend a normal user during her ordinary work time (and work flow) is willing to comply her annotation efforts towards this objective.

From the authoring point of view the question is how to embed this additional information to the text in a way that doesn't bore the human reader with unnecessary explicitness, but anyhow gives the machine the reference to the respective object id of Joseph Joachim. One objective of this type of annotation is to encode the result of a label-to-ID classification (which in X-Media is called fusion). From a syntactical point of view there are well known syntactical patterns which combine a link with an alternative text, like e.g. `<a href="Joseph_Joachim">Joachim</a> had given` or `[[ Joseph Joachim | Joachim]] had given` . A minor challenge remains how to represent such double layered information in a GUI which does not expose wiki markup or html tags directly to the user.

Brahms participated with Schumann and Albert Dietrich in writing a sonata for Joachim; this is known as the F-A-E Sonata. While this sentence is not too difficult to parse for a sophisticated NLP algorithm, it points to a well known problem for semantic annotation: We do have here a typical n-ary relationship, which can't be formalized with one simple binary RDF triple.

If we want to represent this sentence in F-logic (or OWL or RDF), we have to make use of an reification. We could to introduce the class `Writing` and allocate e.g. an instance `writing-12345` with:

```
writing-12345
   a Writing,
   author Johannes_Brahms;
   author Robert_Schumann;
   author Albert_Dietrich;
   product F-A-E_Sonata;
   dedicatee Joseph_Joachim .
```

In the context of the task to support the user to annotate text the main question is: How can we present such reified relationships to the user based on the idea of semantic annotation? To which complexity of RDF graphs is the very concept of annotation compatible with authoring (and amendment) of RDF graphs? Looking back at the history of NLP we are not surprised if we encounter significant differences between the structure (i.e. grammar) of natural language expressions and the structures of underlying representations of the semantics of an expression.

First findings. In crafting a text annotation component we should not try to get trapped in trying to do things which are known to be overly complex. Instead we should strive at providing solutions which are easy to use and easy to understand.

There are more variation of the pragmatics and meaning even of simple annotations than we might think of at a first glance. We have to look more in detail onto a broad range of annotations in order to (a) understand the pragmatics and intentions of the writer who provides annotations, which is (b) a precondition to provide pragmatically a useful and manageable annotation platform for living (as opposed to archived) documents.

The main objective of this deliverable is to understand in more detail what a user does if she annotates living documents. Having such an understanding we are able to strive at annotation support for living documents which is pragmatically useful.

We continue the discussion by transforming the findings of the introductory example into a more analytically structured representation.

2 Living documents vs. archived documents

A living document is a document which is not in a final, stable or archived state. A living document is a document "@work", which means that there are people who still want to edit, amend or refine the document.

Normally annotating a living document is done in order to communicate actions to be done between several authors of a document (or at least one author during time). Such editor's notes then will be tackled in the next editing step. Finally most of the annotations will be removed before a document get's finalized, released and delivered. A first question immediately arises: What’s the purpose of annotations – and how do annotations look like – which are added to a living document in order to be kept alive even until after a living document is finalized?

Having this question in mind the task of giving annotation support for living document may serve two objectives:

- Support the user in managing intermediate annotations which are to be removed before finalizing a document.
- Support the user in authoring text annotations which are intended to be part of the final document.

Even from an uniformed and naive point of view a user has many reasons to annotate a document. If performed by a user 'annotating' a given document means to add or refine what is already written, e.g.

- to add or rephrase text which is missing i.e. in order to clarify or avoid misunderstandings or errors
- to comment what is already written e.g. express being uncomfortable with a specific wording
- to add a short summary
- to tag regions of text in order to have fast acces to it

In a traditional print document (be it a paper copy or an electronic screen version of an office document) there are prima facie two typical places where a user may add annotations: On the margin a user can write more wordy and elaborate annotations. And within the lines of the text itself a user rather would work with symbolic annotations like (possibly coloured or otherwise attributed) highlighting or underlining.

If speaking of "annotating living docs" a core question is: Why should an author want to annotate a text if it is still possible to simply edit the respective text itself? While we have to do some work in order to answer the question the very question itself suggests a first finding: The very understanding of the semantics of the term "annotation" same as the process of annotation is dependent on whether you annotate an archived or a living document. While annotating an archived document means to attach information to a document because it cannot be changed annotating a living document is attaching information to a document as a genuine part of the authoring process itself.

This understanding is also refelcted in legal aspects. Most government legislations in the Western world observe that an archived document is self contained and self sustained. Adding anything to it will automatically imply the creation of a new document. A government memo (having been published) with a pencil marking is per definition a different document from the copy without the pencil markings.

This does also create a whole lot of trouble with e-mails being used for business orders, bookings, delas etc. Since it is common to copy the original request with a response and a response to a response etc. Which document is then the ruling document - each addition may twist the original semantics. All of this has become increasingly important to resolve and may imply some practical problems in our context here.

3 A short phenomenology of text annotations

In this section we analyze and describe in very detail what is done if a user annotates interactively a text document.

3.1 Glossary on the word field "document"

The subject of this deliverable is the topic “living documents”. To start the discussion we’d like to communicate our understanding of some core terms – i.e. the word field "document" itself – in advance. In order to simplify presenting rough ideas we give here some very rough an informal definitions of some concepts used within this deliverable from a very pragmatic point of view.

One first challenge in defining concepts like "page", "title" etc. is that the understanding is different whether you refer to

- a paper copy of an article
- a pdf article on your screen
- a sophisticated html version of an article generated by a sophisticated tool like LaTeX2html or OntoDoc.

Because giving a cross representation (paper copy, pdf on the screen, html web site) definition raises sophisticated ontology modelling issues, our starting point of understanding is a wiki, i.e. a set of interlinked pages which are rendered in html and viewed by a common web browser.

page In the web the content (the rendering of an html serialized char stream) of one browser tab (or the suurounding window, if there are no tabs). For simplicity we don't think of frames etc. here.

subject The cognitive concept (e.g. "The living of Ferdinand Porsche") which a user has in mind when perceiving a page. Tradfitionally such a cognitive concept is communicated in the title of a page. In the times of Wikipedia this subject is also communicated in the very ID of a wikipage.

wiki title In Wikipedia the ID of a wiki page serves as a unique short title of a subject. From a social (and very pragmatic) point of view the Wikipedia title becomes itself a unique ID of a subject.

document Traditionally the term "document" stems from earlier times when communication was paper oriented. Traditional office products like MSWord or the OpenOfficeWriter fully focus on supporting the user in authoring multisection multipage documents. Tools like LaTeX2html or OntoDoc typically cut such a traditional document into a set of interlinked and browsable web pages. If we speak of a document when looking at such an sliced web representation the term "document" refers to a whole set of narrowly related web pages which could be seen a (probably virtual) result of such a printdocument to html slicing step. In other words: A set of html pages comprise one document it there is (at least in pinciple) a description or policy known how to compiling these pages into one document.

formal section heading A heading within a document which describes rather the formal function of a section than it's content. Typical formal section headings are Contents, List of Figures, Abstract, Introduction, Summary, Bibliography or Annex . A "semantically aware" doc2html transformer like OntoDoc doesn't translate formal sections into own pages.

material section heading A heading within a document which gives a very short summary of the section's content. In order to be short and concise a material section heading assumes that the reader is aware of the context, i.e. the general subject under consideration and the headings of the parent sections. Because of this conciseness assumption a doc2html transformer – even if it is told to translate each material section into one single page.-- cannot assume the section headings to be unambiqe.

header, body The header of a text chunk contains the chunk's metadata, the body it's data. If dealing with a structurally sound text structure representation the title and subtitle of a text chunk are in fact metadata. However, in poor text formats like html the title of a section is identified with a data chunk which is tagged with tags like `h1, h2 .. h6` , `th` or `caption` etc.

title, subtitle A title is part of the header of a text chunk.

atomic chunk An atomic chunk is a sequence of chars which belongs together. Typically we identify one atomic chunk with one or some few words ("bike", "invasive Haemophilus influenzae type f infection") in a text.

chunk A non atomic chunk is a chunk which embeds chunks recursively. You can represent a chunked text in XML. From a pragmatic point of view a text chunk is simply an xml element which contains some text (either directly – then it is an atomar chunk – or indirectly, because it contains at least an atomar chunk.). Note that this understanding of chunk restricts chunking of a text into a tree structure without overlappings. This means that chunking a text only reveals shallow syntactic text structures.

3.2 Subtasks of interactive annotation

Selecting a segment within a text and annotating it interactively typically comprises three analytically distinct tasks.

3.2.1 (1) Define a region to be a new chunk of information which will be the subject of an annotation

Often text chunks are already given, e.g. sections, paragraphs, hyperlinks, already highlighted phrases. But often the reader wants to define a new text chunk which she wants to annotate. This calls for defining a new region being a new text chunk, e.g. by interactive highlighting or by introducing a new heading. In fact there is a whole bunch of good old fashioned annotation (GOOFA) which perform exactly this structuring task:

- A heading formulated in terms of content is short and concise description of a section – hence it is an annotation!
- A formal heading (a heading in terms of structure) like "introduction", "summary" indicates a relationship of the sections's content to the parent section
- "Semantic" paragraph styles like "exercise" (in text books), "warning" often can be understood as text commenting, amending, discussing the previous text.
- Tree ordered discussion threads in social web platforms often are typical (recursive) annotations of annotations of annotations.
- In discussing an inline citation an author uses her text itself as an annotation of another text.

In discussing how to annotate text in detail it is frequently missed that commenting sections with headings and structuring sections with paragraphs etc. is a typical and first class mode of annotating text.

However, when having structural elements like sections, headings etc. it cannot be taken for granted that the object of an annotation is an atomic or even compact segment. We shall call an atomic region a region which doesn't have subsegments, e.g. a simple paragraph, a list item, an emphasized, a linked phrase etc. In a complex region there are subsegments, typically a section, a table or other (list of) atomic regions. (Complex regions sometimes are not even compact. An example for a non-compact region would be a single voice within a music score or (in html) a column of a table. In fact it is not always trivial to select (and to display) a non-continuous region interactively in common WYSIWYG systems. This opposes a serious constraint on displaying non-continuous annotations (probably gathered from more sophisticated NLP methods like e.g. relation detection) to the user.) If dealing with non-atomic regions we in general have to decide whether annotating a node annotates only the node or also it's whole subtree.

If we are looking for other interaction paradigms how to identify a region of interest being the subject of an annotation there are several different paths of interaction:

- Interaction with user like WYSIWYG highlighting (Note: WYSIWYG segmentation typically is rather fine grained. Interactive segmentation with "highlighting" typically is done within one or two paragraphs.)
- Typing in of syntactic shortcuts, e.g. `[[link]]` or `*emphasized text*`
- Anticipative segmentation by information extraction: IE algorithms "slice" more complex ODF documents into a (adequate) number of slices. Such a slicing doesn't have to be necessarily exhaustive. Heuristics and/or metrics for "adequate" slices are wanted.

3.2.2 (2) Select a text chunk

Selecting a certain chunk (either already given by text structure or temporarily defined by user interaction) can be done by several different interaction paradigms like

- move the cursor into the chunk
- move the cursor into a representation (i.e. a title) of the respective chunk
- select the chunk in an outline view of the text highlight a region

Selecting a chunk is a non trivial task if the chunk itself is recursively structured. Note that while the result of an *interactive* fine grained segmentation most often is a compact segment of text, non-atomic (and even non-compact) chunks may make up the vast majority in a text which was chunked by more sophisticated IE or NLP algorithms.

3.2.3 (3) Associate the selected text chunk with a tag

As a tag we want to call a label used for annotating. Controlled term tags are words or phrases which stem from a list (or more complex structure) of terms with an explicit, more or less explicit or formalized meaning. (Free term tags don't have a controlled vocabulary in the background. They are mainly used in folksonomies.) More sophisticated commercial text annotation tools already can suggest concepts in the text to be tagged according to a list of controlled terms.

Controlled term tags typically are gathered from an external structure like:

- hard character formatting like bold, italics, underline. Note: Many users communicate text structure only by means of formatting. There is an astonishingly persistent understanding that a line formatted e.g. "24pt bold" is a heading
- soft formatting with OOo stylist (F11)
    - paragraph styles like heading1, text body, list item etc.
    - character styles
- distinct object
    - publicly defined concept or named entity from wikipedia
- XML (i.e. xhtml or DocBook) element
- ontology element

An important related question is (a) whether to allow the user and (b) how to support the user to extend the controlled vocabulary during the annotation process. To integrate a text annotation component *narrowly* with a sophisticated terminology management component is one of the most challenging (and as of today not sophisticatedly solved) challenges in X-Media.

3.3 On the granularity of an annotation

In the introductory example we annotated exclusively words or short phrases. While having words as the thing which is annotated seems to be pretty intuitive it is not self-evident at all. Quite the opposite is true: In order to understand the pragmatics of annotating we have to discuss at which levels of granularity annotations occur – and, in advance, how the user is supported in selecting the object she want's to annotate: Before you can annotate a segment you have to identify a segment as a distinct segment!

Typical segments a human user might want to annotate are e.g.:

- A word or short phrase: This is the level of annotating wiki links semantically. `[[author::Johannes_Brahms]]` Semantic wiki links are phrase annotations.
- A sentence
- A region within one paragraph
- One or more subsequent whole paragraphs within one section (this is the level we suggest to use in the annotation tool, see below)
- A whole section: This is the annotation level which the SemanticMediaWiki assumes to be reasonable by default
- Document

The typical garanularity which typical NLP/IE algorithms segment and annotate on (i.e. take as a subject of annotation) are slightly different:

- stemming / lemmatization: mainly words
- named entity recognition: word up to phrase level
- POS tagging: sentences
- wiki markup parser
  - coarse grained granularity w.r.t. par, list item or heading
  - fine grained w.r.t. links

As easily can be seen IE/NLP and a human user have different views onto a text. While NLP/IE algorithms rather focus on recognizing known objects (or at least anonymous objects as instances of certain classes), the user often is interested in annotating larger segments of text.

Pondering about these findings we bvelieve there is no fast or overall solution to the challenges stated. Consequentially we suggest to start from the simple end. The issue is not to get trapped in very accurate, domain specific annotations that use cues that only external expert readers easily can understand (respective that can be accessed solely with rather complex knowledge based systems). Instead we strive for a simple and manageable annotation component to

- leverage the quality of work, making the user responsible for the work
- make sure that the result of annotating is part of a continuum, i.e. that it hinges onto other annotations so that they can be traced and found more easily.

A generic approach that could lead to a generic and extendable solution is to aggregate fine grained annotations from IE to a more coarse grained user interaction level.

We need a policy how to represent fine grained annotations which mostly occur on word level to larger contexts, i.e. to paragraph or section level.

While this approach addresses many problems of fine grained vs. course grained annotations, it does not solve another core question: What is meant with highlighting text?

3.4 On the semantics of highlighting a region

What could you mean by highlighting a "short" region within an html page? (A "short" region has a length of one word at minimum up to some words. A whole sentence would exceed a "short" highlighting.)

3.4.1 Sth. which a page refers to

The highlighted region may be sth. (i.e. a subject or the label of a subject) which the page refers to. In this case the URL of the page "represents" something else, i.e. it's content. For our annotation simply adds additional *data to* (as opposed to *data about*) the pages content we call this annotation data annotation, page annotation or simply "annotation".

To give an example: (Say that) the page *wikipedia.org/FerdinandPorsche* represents (signifies) the content (here: the historical living, i.e the person) Ferdinand Porsche. If we highlight the region "Stuttgart" within the sentence "In April 1931 Porsche founded his consulting firm, Dr. Ing. h.c. F. Porsche GmbH, Konstruktionen und Beratungen für Motoren und Fahrzeugbau, in Stuttgart" this means:

- Something which is mentioned in the current context *refers* to some other thing (semantically spoken: e.g. to an instance e.g. of owl:thing or skos:concept) which has the label "Stuttgart".

We could formalize this fact as

```
(ThisPage refersTo X) and (X hasLabel "Stuttgart")
```

In our wired marker example ontology (see below, section about Wired Marker) this is the yellowish tree with the top concept data: "(this_page's)_topic refers_to topic_XYZ"

3.4.2 Label of a page's content itself

The highlighted region may be a label of the page's content itself.

To start with an example: (Say that) the page *wikipedia.org/PorscheCompany1931* represents (signifies) the content (here: the car company) which is referred normally to simply by the name "Porsche". Actually the full name of the company is pretty verbose. If we highlight the region "Dr. Ing. h.c. F. Porsche GmbH, Konstruktionen und Beratungen für Motoren und Fahrzeugbau" within the sentence above this means:

- The subject of our page (here: wikipedia.org/PorscheCompany1931, i.e. the company "Porsche" in year 1931) has the *label* (semantically spoken: a written representation; in RDFS we would speak of a string literal) "Dr. Ing. h.c. F. Porsche GmbH, Konstruktionen und Beratungen für Motoren und Fahrzeugbau".

We could formalize this fact as

```
(ThisPage hasLabel “ Dr. Ing. h.c. F. Porsche GmbH …  ”)
```

Note that it is the (content of the) page itself which has an attribute (here: the string valued attribute “hasLabel”), as opposed to the former example, where the (content of the) page *refers* to sth. which has the highlighted string as a label.

For our annotation adds data *about* (as opposed to “*to*”) the pages content we call this annotation meta data annotation. In our wired marker example ontology this is the purple tree with the top concept meta data: "This topic has property P"

3.4.3 Description of the information object itself

The highlighted region may be data which describes the information object itself

To start with an example: The page *http://www.w3.org/MarkUp/2008/ED-rdfa-syntax-20080125* is a typical W3C spec draft. As always the W3C are very explicit on giving authorship and versioning information to the reader. If we highlight the name "Mark Birbeck" within this document this means:

- One of the authors of the document which can be retrieved under http://www.w3.org/MarkUp/2008/ED-rdfa-syntax-20080125 has the label Mark Birbeck.

We have to deal here with the subtle difference between sign and reference: In a triple like

```
rdfa-syntax-20080125 hasAuthor MarkBirbeck
```

it is stated that the mental concept which is referenced by the given URI `http://www.w3.org/MarkUp/2008/ED-rdfa-syntax-20080125` (here: the RDFa syntax) has an author, i.e. has a person who has invented it. But as easily can be geussed it was not only this single person who invented RDFa alone. What we more probably would like to express is the fact that Mike was the author of the document which describes the RDFa syntax.

We are not able to formalize this fact in standard RDF. Instead we have to make use of refinements like RDF graphs, e.g.

```
rdfa-syntax-20080125-citeId
   {rdfa-syntax-20080125
        hasSubtitle
             “A collection of attributes … to support RDF”
   }.
```

```
rdfa-syntax-20080125-citeId hasAuthor MarkBirbeck .
```

For our annotation adds data not w.r.t. the document's content but to the very document itself we call this annotation a meta metadata annotation. We give metadata to a document, i.e. a collection of data and metadata. In our wired marker example ontology (wired-marker-seeding) this is the green tree with the top concept meta meta data: "This information object has property Q" (media annotation)

3.4.4 The highlighting indicates sth. which is important

Highlighting a region may simply indicate that sth. is important, annoying, interesting etc. Such conservative annotations are far away from everything we are thinking of when speaking of annotations in X-Media. However, simply stating "we do not consider such annotations" would be a pretty silly answer from a pragmatic point of view: In fact it is this type of annotation which is most important when working with text as a learner. We have a vast variety of informal meanings with such a highlighting. If you use different colours in order to distinguish semantics like "important!", "ask my professor" or "keep in mind" you perfectly adhere to one of the intended use case of wired marker.

3.4.5 You want to archive the highlighted region into your PIM

Often you simply mark a region of text in order to use it later. Then you probably want to copy the text into your personal information manager. You will drop such a snippet to a folder, i.e. one of the wired marker folders. In marking a more extended region of interest you *define* a self containing chunk (segment) of text which intentionally become the subject of an annotation.

In this case the semantics of annotating text is exactly the other way round than above. Instead of annotating a whole page (represented by an URI) whereby the highlighted region is meant to be (the label of) a related topic or a label of the page’s topic itself, you now annotate the highlighted region being a subject whereby the according related topic is selected by dragging the highlighting into a wired marker folder.

Note that this sort of highlighting is *not* an annotation in the X-Media sense. Instead it is the act of defining a self containing text chunks as a sub-document within a more complex document. Typically such stand alone chunks are intended to being used as a quotation within another text. An adequate handling of such a highlighting would be to extract the selection and allocate an URI of its own right for it. Doing such things is the dedicated task of a text processor. In general we cannot handle this case it in an annotation tool.

However, there is a special case which could be handled in principle within wired marker: If the segmentation the user want's to point at is already given in the document we can use the respective fragment id! From a UI point of view a pre-segmented region can be selected i.e. by a linked table of contents. The URI incl. fragment id would then point to a segment, and the user can use the highlighting according to one of the supported semantics. (A shortcoming of wired marker is: Even if you annotate an URI with a fragment id, wired marker ignorse the fragment id for annotation. This is because hyperanchor uses the #hyperachor1.2 fragment notion and overrides the anchor / the page fragment which the user already selected).

3.5 On the semantics of the chosen tag in a controlled vocabulary

Of course the very meaning of a tag itself which is used for annotation affects the meaning of an annotation. (In order to make authoring of the examples easier we use here an inline XML representation. Needless to say that we could think also of a nice WYSIWYG view onto this XML structure which also would be suitable to visualize external X-Media annotations.) Compare the following examples:

Peggy is fond of …

- `<car>Cinquecento</car>`
- `<span about="[myonto:170CF53A]" >Cinquecento</span>`
- `[[ likes :: Fiat_Cinquecento ]]`

In the first case we simply state that Peggy's favourite is a car (as opposed i.e. to Cinquecento as the Italian Renaissance of the sixteenth century). In the second sentence we refer to an object with a certain id in our ontology. (In fact we refer again to a class of objects, i.e. the class of Cinquecento cars with the type constructor number 170CF53A.) And in the third sentence we use the extended Semantic Media Wiki link syntax in order not only to refer to the Fiat_Cinquecento, but also to tell more about the very relationship between Peggy and this car: Being fond of something is modelled by the property "likes" in our ontology.

The question arises: Wouldn't it be nice to derive semantic meaning from an arbitrary XML structure?

Look at the thought experiment developed in X-Media WP5[1]:

```
<sentence>
  In
  <year>2005</year>
  <presented subj='Fiat' object='Punto'>
    Fiat presented the new Punto
   </presented>
  at the
  <exhibition>IAA</exhibition>
  in Frankfurt.
</sentence>
```

What is meant by the various XML tags, e.g. with `<exhibition>`? As already stated in the introductory example we have to decide here whether the tag "exhibition" adds information (here: type information) to an already uniquely identified object (here: the object with the ID "IAA"), or whether the tag adds information to the text which al-

[1] Published in: Sebastian Blohm, Jürgen Umbrich, Philipp Cimiano, York Sure: Iterative Learning of Relation Patterns for Market Analysis with UIMA. In UIMA Workshop at GLDV Frühjahrstagung. April 2007. Download: http://www.aifb.uni-karlsruhe.de/Publikationen/showPublikation?publ_id=1472

lows the user to identify an object uniquely from it's string (here: the hypothetical unique object IAA-Frankfurt-2005 which has the non unique skos:altLabel "IAA"). Sticking to the former interpretation we could paraphrase the tag as: "The well known object with the id IAA is of type exhibition." But more probably the intended meaning reads as: "The string IAA is a reference to an object of the type exhibition"

While this understanding looks to be straight forward it is more difficult how to read the tag "presented" or the tag "sentence". Prima facie the tag <presented subj='Fiat' object='Punto'> seems to represent an RDF triple, serialized in a RDFa-like manner. But what is then the meaning of the included text – and what would be the meaning of XML tags if they would be present?

We face here the general problem of how to interpret an XML structure semantically. It is well known that XML itself brings no semantics at all with it. There is no way to derive a semantic model from a given XML Schema or XML DTD: The XML world is purely syntactical. We have to accept that there is no generic or well defined path which allows for interpreting an arbitrary XML structure in a semantic matter.

While we have discussed here an example in XML style, our findings are perfectly true also for GUIs which associate highlighted regions with tags from a list or a tree. (This is of course the case with i.e. with wired marker or using OpenOfficeOrg char styles as semantic markup.) Our findings are also true for GUIs which allow for an interactive selection of regions which are complex enough to exceed the expressivity of XML (and which would call for more sophisticated technologies like SGML).

3.6 Intermediate summary

Annotating a text comprises three steps:

- Define text structure, i.e. chunk a text into segments which are worth to be annotated.
- Select a structural element, i.e. a chunk.
- Tag the element, i.e. associate the selected chunk with an element from a controlled term or free term vocabulary.

Dependent on the granularity of the selected chunk there are different possibilites of what an annotation is intended to mean. Annotating a single phrase simply supplies additional data to the phrase – data which either helps the reader to disambiguate the phrase or which adds information to an already identified object. If the user annotates a more medium grained text chunk (e.g. a whole paragraph, a table) she might add data, metadata or meta meta data to it – very similar to the possibilities if she'd annotate a document as a whole. In fact annotating a medium grained text chunk might be interpreted as the act of declaring that the respective text chunk could be interpreted as a stand alone information object (describing a distinct subject) which is worth being annotated.

The most relevant variable however is the modelling of the objects we use for annotating within the ontology itself. In more simple cases the annotation tag is a class or an instance. In a more complex case the annotation tag refers to a property in the ontology. The ultimate and most expressive case is that the user want's to annotate a chunk of text with a more or less RDF graph itself.

Because we cannot please this ultimate generic approach we have to look for a much more simple solution.

4 Cycling of iterative annotations between man and machine

Suppose that the user has written a new piece of text (say some paragraphs) and sends this newly written text to an X-Media component for annotation. Suppose further that the NLP component does segmentation and annotation and writes back the annotations to the piece of text. The user then looks at segments and it’s annotations and amends some of them. Some selected annotations are explicitly positively or negatively evaluated by the user as (a) being important, (b) being suprising, but correct (such an annotation would yield the most information gain!) or (c) more or less wrong (i.e. to be deleted).

Complementary there is a (major) part of machine made annotations which are not considered (and thus not processed) at all by the user. This may be simply because a user often has not enough time for doing boring busy tasks. Another issue could be that a user only want's to know annotations which are interesting. As a result we get a piece of text where some annotations are explicitly processed (i.e. confirmed, amended, rejected) by the user. Suppose that this user amended paragraph is sent again to the X-Media annotation stack.

Then questions arise like:

- How do the automatic annotation components deal with annotations which are already there, i.e. interactively confirmed annotations, interactively rejected annotations and annotations which the user didn't look at ?
- How do user given and machine extracted segmenting work together, i.e.w.r.t. subsequent cycles of selecting a segment prior to annotating it?
- How can we refer to annotations of the X-Media stack without having to use an online connected and X-Media specific annotation tool?

While these challenges in principle could be solved on a technical basis the main issue with living documents remains: If annotating a living document is an act of authoring: Do we want an automatic annotation component to act as a co-author of a document?

5 Suggested approach

5.1 Utilizing the Firefox Wired Marker plugin

Wired marker is a free firefox plugin. You can get if from http://www.wired-marker.org/en/index.html or simply ask firefox to search for wired marker (Tools > Add-ons > Get Add-ons) Wired marker basically allows for highlighting regions within a rendered html browser page and dragging the highlighted region into a folder structure. If we interpret the wired marker folder tree as an "semantic" object we have a prototype of an interactive semantic annotation tool at hand. The question of importance however is how to interpret the folder structure in detail – i.e. when knowing how different the semantics of highlighting text may be. Some restricting characteristics of wired marker (as compared to a more sophisticated annotation tool) are:

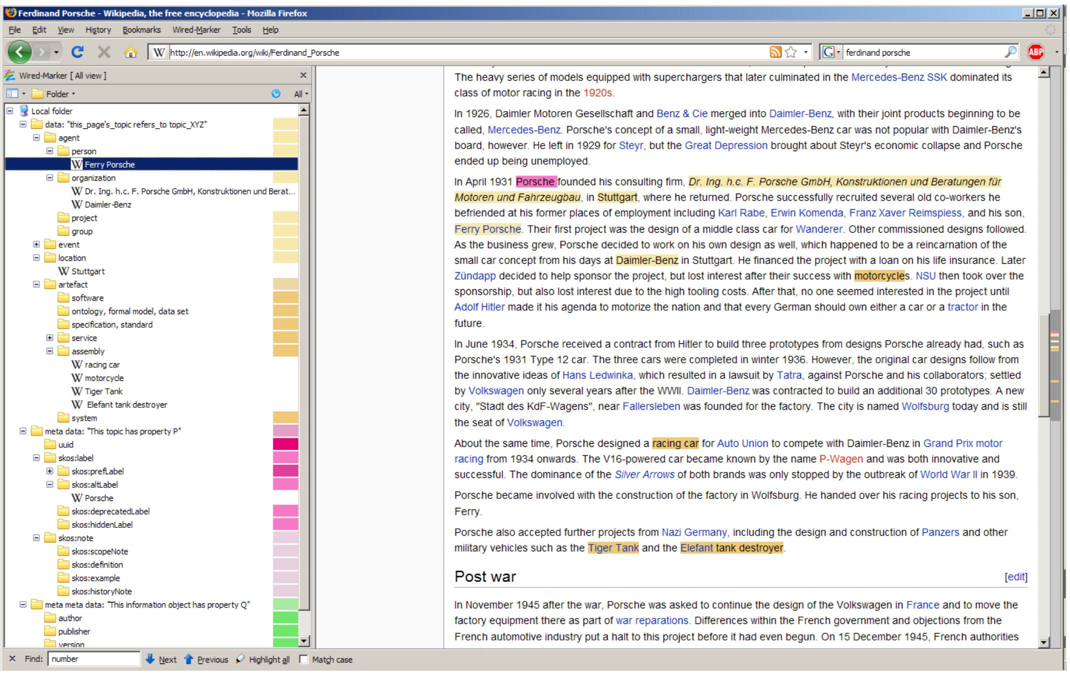
- We only have page annotations. Even if you explicitly browse to URL#fragmentID, it is only the URL and not the URL plus fragmentID which will be annotated. (However: Because we have the exact path which is annotated we can turn page annotations also to region annotations.)
- There will be always all bookmarked regions be highlighted. Suggestion what to make better: Only highlight folders which are shown also in the wired marker folder view.
- If the same region is draged and dropped to multiple folders the coloring of the region should be done according to the currently selected folder. (And folders not selected / not expanded / not focused) should not result in a highlighting at all.

Even if wired marker is pretty lighweight and generic it is perfectly fine to serve as a basis for prototyping interaction patterns of a html (and thus browser) based annotation tool. We i.e. can tackle even HCI related questiones like:

- How complex is an annotation ontology exptected / allowed to be?
- How many annotations should be shown in total?
- Do we really want to have a colorful patchwork rug?
- How can we support the user the many different meanings of annotations?

We built a Wired Marker folder tree which is intended to reflect the findings of our little phenomenology of annotations above.

A typical data annotation might looks like this:



The heavy series of models equipped with superchargers that later culminated in the Mercedes-Benz SSK dominated its class of motor racing in the 1920s.

In 1926, Daimler Motoren Gesellschaft and Benz & Cie merged into Daimler-Benz, with their joint products beginning to be called, Mercedes-Benz. Porsche's concept of a small, light-weight Mercedes-Benz car was not popular with Daimler-Benz's board, however. He left in 1929 for Steyr, but the Great Depression brought about Steyr's economic collapse and Porsche ended up being unemployed.

In April 1931 Porsche founded his consulting firm, *Dr. Ing. h.c. F. Porsche GmbH, Konstruktionen und Beratungen für Motoren und Fahrzeugbau*, in Stuttgart, where he returned. Porsche successfully recruited several old co-workers he befriended at his former places of employment including Karl Rabe, Erwin Komenda, Franz Xaver Reimspiess, and his son, Ferry Porsche. Their first project was the design of a middle class car for Wanderer. Other commissioned designs followed. As the business grew, Porsche decided to work on his own design as well, which happened to be a reincarnation of the small car concept from his days at Daimler-Benz in Stuttgart. He financed the project with a loan on his life insurance. Later Zündapp decided to help sponsor the project, but lost interest after their success with motorcycles. NSU then took over the sponsorship, but also lost interest due to the high tooling costs. After that, no one seemed interested in the project until Adolf Hitler made it his agenda to motorize the nation and that every German should own either a car or a tractor in the future.

In June 1934, Porsche received a contract from Hitler to build three prototypes from designs Porsche already had, such as Porsche's 1931 Type 12 car. The three cars were completed in winter 1936. However, the original car designs follow from the innovative ideas of Hans Ledwinka, which resulted in a lawsuit by Tatra, against Porsche and his collaborators; settled by Volkswagen only several years after the WWII. Daimler-Benz was contracted to build an additional 30 prototypes. A new city, "Stadt des KdF-Wagens", near Fallersleben was founded for the factory. The city is named Wolfsburg today and is still the seat of Volkswagen.

About the same time, Porsche designed a racing car for Auto Union to compete with Daimler-Benz in Grand Prix motor racing from 1934 onwards. The V16-powered car became known by the name P-Wagen and was both innovative and successful. The dominance of the *Silver Arrows* of both brands was only stopped by the outbreak of World War II in 1939.

Porsche became involved with the construction of the factory in Wolfsburg. He handed over his racing projects to his son, Ferry.

Porsche also accepted further projects from Nazi Germany, including the design and construction of Panzers and other military vehicles such as the Tiger Tank and the Elefant tank destroyer.

Post war [edit]

In November 1945 after the war, Porsche was asked to continue the design of the Volkswagen in France and to move the factory equipment there as part of war reparations. Differences within the French government and objections from the French automotive industry put a halt to this project before it had even begun. On 15 December 1945, French authorities

We describe the complete wired marker folder tree in the next sections with the titles

- data
- meta data
- meta meta data

5.1.1 data: "(this_page's)_topic refers_to topic_XYZ"

Meaning: If the user has marked, dragged and dropped a marked string into wired marker folder XYZ the semantic is: "The current page refers to an instance X in the ontology where (1) X is a narrower term of XYZ and (2) X has our highlighted string as one of it's labels". If the user dragged and dropped the URL directly to wired marker the semantic is: "The current page is about a topic which is pretty similar to a concept in the ontology which is a narrower concept of XYZ."

This annotation will produce a triple like

```
URL skos:subject X, with (X skos:broader XYZ) and (X skos:label high-
lighted_string)

URL myont:related X, with (X rdf:type XYZ) and (X rdf:label high-
lighted_string)
```

in words: highlighted_string is interpreted being a `skos:label` (and thus also being an `skos:preflabel, skos:altLabel` etc.) of an instance of a `skos:concept`.

Note: Because we work with SKOS all folders are instances of `skos:concept` (as opposed to classes compared to other ontology engineering tools).

- agent sth which can be recognized being a responsible (i.e. liable) actor
    - person a natural person
    - group
        - project
        - company
        - group
- event sth. that happens in time
    - meeting
        - conference
        - workshop
    - breakdown This is sth. what we are interested in especially in the bike brake failure show case ontology
        - bike brake failure
            - brake overheat
            - worn rim
- location
- artefact An artefact is sth. that may have parts, modules, subsystems etc.
    - software
        - ontology, formal model, data set
    - specification, standard
    - service
        - search engine
        - shop
    - assembly sth. you can touch; most often it is made of / has other assemblies. Example: a bike brake, made of brake pads, a brake lever etc.
        - bike brake
            - bicycle disc brake
            - bicycle rim brake
    - system sth. abstract that works and interacts together. It often has sub systems, and it alwys involves one or more assemblies. Ex.: A back pedal brake system (and i.e. it's sub system "brake actuation") involves the assembly bike chain.

5.1.2 meta data: "This topic has property P"

Meaning: The bookmarked URL (without hyperanchor1.2 part) will become an instance of a `skos:subject`. (Actually ever newly added (instance of a) concept should immediately be copied into the "instances of `skos:concept`" tree.)

This annotation will produce a triple like

```
URL skos:label highlighted_string
URL rdf:label highlighted_string
```

In words: highlighted_string is interpreted being a skos:label (and thus also an skos:preflabel, skos:altLabel etc.) of an instance of a skos:concept.

- uuid
- skos:label The marked string on this page will become the respective skos:label (prefLabel etc). If there is no marked string (i.e. when dragging directly the URL) we will take the title of the page as the label.
    - skos:prefLabel
        - prefLabel@en
        - prefLabel@de
    - skos:altLabel
    - skos:deprecatedLabel
    - skos:hiddenLabel
- skos:note The marked string on this page will become the respective skos documentary note
    - skos:scopeNote
    - skos:definition
    - skos:example
    - skos:historyNote

5.1.3 meta meta data: "This information object has property Q" (media annotation)

Meta meta data are describing the page itself (as opposed to it's content).
This annotation will produce a triple like

```
(URL2 skos:version highlighted_string) and (URL2 comm:??? describes
URL)
```

Typical meta meta data are:

- author
- publisher
- version
- last publishing date

6 Conclusions and Outlook

We have built a little tool which exports a folder tree like above from a (view onto a) given ontology. From a technical point of view we are already able to configure wired marker being a fully fledged semantic annotation tool for XHTML files. However, in trying to use this tool for everyday work (our task was to annotate a bunch of web pages w.r.t. the domain of semantic terms like OWL, F-logic or triple etc.) we learned:

- Many users pretty rarely did spend enough time to perform appropriate annotating.
- Some users who didn't had the whole taxonomy in mind (in fact the vast majority of the users) where annoyed when they had to perform lengthy searches for the correct annotation term. They additionally missed term disambiguation and scope-of-use information when dealing with more complex terminologies.
- Some users where annoyed if they didn't find their most specific annotation term (as opposed to generic upper level concepts like "assembly" or "software") within the folder tree. They argued that the information gain from an annotation is not worth the annotation effort if the annotation class is too generic.

We conclude from this finding that neither striving for completeness nor providing only top level concepts pleases the user. We have to do two things:

- We have to investigate further in how to provide manageable domain and task specific annotation ontologies to the users if we really want to support them doing their task.
- We have to decide at which level of granularity we mainly want to support the user in annotating.

6.1 A plea for a lossy annotation aggregation

On the one side it is clear that the user does not has the time to amend annotations on sentence (resp. a table element, part of a figure) level. On the other side we have evidence that a user might be willing to annotate larger areas of text like i.e. sections. If we consider all these things there is no need to tackle the challenging problems of amending complex, fine grained annotations of a text.

In order to allow for interactive annotation we suggest to aggregate annotations to an intermediate level of text granularity like a section, a paragraph, a table, a figure and so on. (It has to be discussed whether it is the interactive text annotation component itself which has to do this aggregation or whether there should be a separate aggregation peer. Suppose that the user amends aggregated annotations: How do these amendments on the aggregated level loop back to its origins, i.e. the fine grained annotations which are produced by the X-Media annotation components?)

Annotation aggregation per definition is a lossy transformation. However, the losses are not necessarily bad. Throwing away information can be understood as an act of information consolidation, which is i.e. true if it is the user which amends the aggregation.

It is perfectly fine to have aggregated annotations and fine grained annotations in place at the same time. This is even true if they are not consistent, since consistency of annotations never was requested being a precondition. Quite the opposite is true: As long as we cannot assume that automatic and fine grained IE results are consistent, we want to involve the user to aggregate them to hopefully more consistent annotations on a more coarse-grained level.

Letting the user amend only aggregated annotations also goes with GOOFA text structuring. Giving something a heading (a caption, a title) identifies it being a text portion worth being the subject of an annotation.

Next steps: Having defined sections as a first class focus for annotation aggregation we also have a rule of thumb at which granularity we want to roundtrip complex xhtml or ODF documents to other annotation components. We i.e. will outline how to use a Semantic Wiki for annotating sliced ODF documents, which is perfectly adequate from the user interaction point of view. Wikis which are able by design to store even very fine grained RDFa annotations (e.g. IkeWiki) are to roundtrip both the aggregated and the fine grained annotations. Wikis which does not allow for fine grained RDFa annotations but instead support only page annotations (like e.g. the SemanticMediaWiki) are to roundtrip merely aggregated data,

6.2 Discussion

Most of our discussion circulates around the issue of semantics. The more complex aspects that we highlight are more related to intentional logic (typically referred to as linguistic pragmatism) than pure semantics. Intentional logic can be highly subjective. For many years linguists have shunned the topic, as syntax (grammar structure) and pure semantics have constituted their main focus.

Intentionality is an issue with respect to communication in all aspects. It is clearly demonstrated when people tend to use single channel communication i.e. telephone, e-mail etc. What the person intends to convey gets lost in translation at the other side very quickly, even with well formed sentences and literate people. Sentiments conveyed in e-mails have proven to be over amplified as a written word have greater power than spoken, often because voice tone, body language etc. are absent an unable to moderate the spoken word and thus the interpretation. Adjectives and negative/positive nouns are cases in point - positive adjectives or nouns tend to create a sense of euphoria at the other end while the opposite is likely to start a fight, in spite of moderate ambitions of the writer. Our example "the virtuoso pianist Clara Schumann" is clearly associated with this. The ignorant may eliminate the adjective altogether, while a music lover may become entirely fixed to this.

The issue we have discussed is also very much exploited by marketing. The French cosmetics industry have been shrewd here using semiotics as an instrument for stimulating sentiments related to "need to have" (as compared to need to annotate). By using special wording they know that a wanted demographic group will catch on, while the rest will be ignorant to it.

As we have pointed out there is an inherent complexity and fundamental issue of tagging. When looking for more generic answers to this one would have to dive deeply into semiotics. However, this is a complex area. We cannot do here more than bringing about the necessary respect for the matter which we otherwise find rather absent in discussions on semantic tagging. For the difficulties are not so much triggered by the pure semantics, but the subjectivity related to author's intention compared to the reader's assumed intention. Would the writer have written "virtuoso pianist" if he did not like the – female composer (sic!) – Clare Schumann or music? Sports journalists are notorious in this respect, often using pompous expressions for things they like and enjoy personally while being neutral or negative for other things.

This problem is greatly amplified by the problem of "single channel - single cue" communication. It is really an art to convey complex meaning in a short and narrow form. Implicitely we claim the same by addressing the interpretation of the XML code. Such a code is not even a full sentence. We have to keep in mind that most communication needs have to be very redundant using very many co-references to be well understood. The degree of redundancy needs to increase if there is a contextual discrepancy between the speaker and the listener - which is basically the case we dicussed above. The XML code and even the Wikepedia note is really not sufficient in itself to give you the answers that we seek. The answers must be found in the air - suggesting that general knowledge about the subject or domain ultimately drives the interpretation process. Since personal experience differs greatly our emphasis of the different words will also vary and due to lack of redundancy and channels it is hard to bridge the gap between individual interpretations and emphasis.

Literature

In our argumentation we did not make use of specific argumentations or proofs stemming from single and identifiable publications of other authors: From a scientific point of view there is nothing we have to cite. We also hope that our discourse contribution will be valuable to the reader without having to point to affirmative 3rd party voices.