The Lowercase Semantic Web:
Using Semantics on the Existing World Wide Web

Ben Munat – May, 2004
The Evergreen State College
munben25@evergreen.edu

Abstract: *Despite the enthusiasm of certain technological utopians, building the Semantic Web is a hard problem. Capturing true, incontrovertible meaning of natural language remains an elusive goal. Also, the widespread adoption of complex data modeling languages like RDF or OWL is unlikely to happen overnight. A more feasible approach would be to focus first on specific domains and use the simpler technologies. XML and XHTML are two relatively simple technologies already on hand and can be used to lend a little more semantic value to the Web's existing content.*

This paper gives an overview of the idea of a "Semantic Web," discusses some of the problems in creating a general worldwide web of meaning, and describes a few nascent XHTML and XML based technologies that help clarify their content. Since XHTML is merely html made to conform to the rules of XML, it is not much of a leap for Web site creators to adopt XHTML. And, as we shall see, there are some simple semantic additions to XHTML that one can start using right away. It is argued that this approach fits in well with a more reasonable and piecemeal approach to bringing meaning to the World Wide Web.

Keywords: *FOAF, GeoURL, GRDDL, Lowercase Semantic Web, Mangrove, RDF, Semantic Web, Telemakus, XHTML, XML, XFN*

I. Introduction

Back in the nineties, while the Web was still rocketing into society's collective consciousness, futuristic fantasists were already pondering its next phase. Having billions of documents at your beck and call twenty-four hours a day was nice, but finding anything in this data soup was (and still is) tricky. What if, they wondered, the Web could be smarter? What if it could bring "structure to the meaningful content of Web pages"? [Ber01] What if we could have a *Semantic* Web in which the content was not merely voluminous raw *data*, but was semantically labeled *information*?

Unfortunately, bringing such a dream to fruition is one of the "hard problems" that faces contemporary computer science. In much the same vein as the over-hyped and under-delivered artificial intelligence craze of the eighties, bringing about the sort of universal, multipurpose, readily

available Semantic Web that Tim Berners-Lee predicts in "Weaving the Web" is fraught with difficulties. Computers are fantastic at performing repetitive tasks very quickly and without variation. They cannot, however, make any associations between one chunk of bits and another without being explicitly told to. And, even if humans do the work of creating an "ontology" or concept mapping for the data contained in the computer, we still rely on the particular interpretation of the human or humans doing so.

Besides the inherent difficulty in clearly recording the meaning of content, the brief history of the Web has shown that we cannot always trust Web authors to accurately or honestly label their own content. [Bro04] The HTML "meta" tag, in which page authors could put keywords relevant to their page, has been made nearly irrelevant by attention-seeking hucksters filling these tags with inappropriate, misleading terms. Consequently, most search engines ignore the "meta" tag and instead use secret algorithms to analyze the entire content of the page.

The amount of work and expense to create semantic markup for general Web content has made for a lack of momentum outside of academic circles. Within academic circles, however, as well as the scientific and medical communities, researchers have been looking into using semantic markup to help categorize and synthesize the voluminous amount of research data these areas produce. The subject-specific and finite-terminology nature of these areas helps avoid the overwhelming task of categorizing the whole world.

2. Limited Focus Systems

Sherrilynne Fuller and associates at the University of Washington's School of Medicine have developed *Telemakus*,[1] a system for mining medical research findings. This software maintains a database of published medical articles, which are analyzed and processed into conceptual mappings. A search for a particular set of experimental criteria or outcomes can then be matched to documents that not only use the specific terms of the search, but to documents that use semantically related terms. The project is an encouraging first step towards a more concept-oriented basis for data search, however it clearly benefits from a pre-existing ontology developed for the medical field.[2] *Telemakus* also enjoys the luxury of being developed in an academic environment: there is not the sort of "make or break" urgency that a business would place on such a project.

Mangrove,[3] another Semantic-Web-related project at the University of Washington, focuses on simplifying the creation of metadata and providing immediate gratification to the user as an incentive to enter the data. For example, it offers a fairly simple user interface into which one enters her personal data (contact info, office location, interests, etc.). This data is then published, marked up with an XML vocabulary developed by the project, in an HTML page. The special markup allows the data to be collected by "semantic services", such as a department "who's who" page and event calendars. As with *Telemakus*, *Mangrove* benefits from limiting its scope to a small set of data. However, *Mangrove* does not rely on an existing ontology, but instead encourages users to provide metadata with the reward of seeing their information in a public forum.

---

[1] Readers can try out *Telemakus* at: http://www.telemakus.net/
[2] The UMLS Metathesarus, which includes 900,551 concepts and 2.5 million concept names. [Fuller 2003]
[3] There are test applications of *Mangrove* at: http://www.cs.washington.edu/research/semweb/

Both *Telemakus* and *Mangrove* succeed because they have finite goals. Rather than working to create a single, overarching web of knowledge, these projects take a limited topic area and create a tool that helps people do a specific task within that area. Also, both offer specific and immediate rewards: *Telemakus* gives researchers a way to find previous research data without spending weeks digging through false-positive search results, while *Mangrove* facilitates the collection of personal data, coordinates meeting schedules and maintains a departmental calendar by distributing the responsibility to all involved and making it a simple task to do so.

The metadata that *Mangrove* collects through its annotation user interface is encoded in the extensible markup language (XML) according to a schema created by the project members. This data can then be placed in a regular HTML Web page. One of the core rules for Web browsers since their earliest incarnation has been that they should simply ignore any markup they do not understand. Therefore, when tags such as the following are embedded in a Web page:

```
<html xmlns:uw="http://www.cs.washington.edu/research/semweb/vocab#v1_0">
<uw:event>
    <uw:topic><i>New Grad</i> Orientation Lunch </uw:topic>
    <uw:date>Sep 26, 2002</uw:date> at
    <uw:time>1:00-2:30 p.m.</uw:time> on the
    <uw:location>HUB Lawn</uw:location>.
</uw:event>
</html>
```

the text within the tags will appear in the page just fine, even though no Web browser in the world understands the tag "`<uw:event>`". However, the "semantic services" programs running on the UW computers *do* understand this tag, and all the others in the UW semantic services XML schema. In other words, the content displayed in the browser window looks the same to us, but we have succeeded in telling any interested computer programs the intended meaning of that content.

Mangrove offers a clear example of how simple it is to leverage existing Web technologies to offer immediate and useful tools. And these need not be overly complicated technologies. As mentioned already, these tags *Mangrove* uses are XML tags. And, while some of the sub-languages or "vocabularies" created from XML can get very complex, the basic concept behind XML is stunningly simple. To mark up your data with XML you simply make up a tag that identifies each piece of your data and put the opening version (without the "/") before the data and the closing version (with the "/") after the data. For example:

```
<person>Ben Munat</person>
```

There are, however, a few rules to follow:

- Tag names are case sensitive; convention dictates using lowercase.
- If you nest one tag within another, the inner tag must end before the outer tag.
- *Elements*, as an opening/closing pair of tags (and any data within them) are called, may also have *attributes*. These are indicated with name/value pairs included inside the angle brackets of the opening tag.
- The value of an attribute must be enclosed in quotation marks.

Another useful concept offered by XML, and one that *Mangrove* uses to good effect is that of namespaces. This is a way of preventing the confusion resulting from mixing an XML grammar that uses an "event" tag with UW's vocabulary, which has an "event" tag as well. A namespace is declared, as in the attribute of the "html" tag above, by associating some unique identifier with a short label, like "`uw`". This label is then used as a prefix on all tags that are part of that namespace, and insure that machines (and humans, for that matter) reading the document know that it is UW's event tag and not someone else's. This means that different XML vocabularies can be combined in the same document… like mixing one's own markup about people, places and events at the University of Washington into an HTML document.

3. The Extensible Hypertext Markup Language

Ironically, the examples on Mangrove's Web site are demonstrated in pages written in an older version of HTML. This is ironic because not only is HTML a dead technology,[4] but also because the language that has replaced HTML—the Extensible Hypertext Markup Language or XHTML—was created specifically to allow the sort of tricks that Mangrove undertakes.

There has been a fair amount of confusion surrounding the transition to XHTML, causing its adoption to flounder at a maddeningly slow pace. It is maddening because the purpose of changing to XHTML is deceptively simple. Just make HTML conform to the rules of XML: close and properly nest all tags and use lowercase for tag names. There has also been some narrowing of the restrictions on where certain tags can go in a document and even what tags can be used, but it is not hard or mysterious to write perfectly robust, cross-browser compatible, standards based Web pages in XHTML.

The confusion about XHTML may have been due to another modern, standards-based language: Cascading Style Sheets. In the early, sudden explosion of the Web, browser writers took it upon themselves to add proprietary tags to HTML in order to create certain rendering effects. This created a disastrous Tower-of-Babel effect in which certain tags only worked in certain browsers, and completely ignored the fact that HTML was not intended to be presentation markup. So, it is true that valid, correct XHTML does not offer some tags that early Web developers came to rely on. And, because the same people that were promoting the transition to XHTML generally pushed CSS as well, the difficulties with switching to CSS for presentation (incomplete and inconsistent implementation in browsers and occasionally frustrating terminology choices) likely led people to fear XHTML as well.

In fact, this is of particular interest to the notion of encoding semantics in a markup language. It is deeply ironic that Tim Berners-Lee kicked off the Web with only HTML. If he had created CSS at the same time, the presentation/content dilemma may not have occurred to the degree it has. If he had created XML first, and introduced HTML as handy pre-defined subset of XML for marking up documents, the difference would have been even more clearly pronounced. Had history played out this way, we would have had more of a semantic foundation to the Web from the start. The meaning of the tags has always been there, but people, generally thinking visually rather than conceptually, have missed this nuance. If things had happened in this more deliberate manner, browsers would not

[4] The development of HTML ended in 1999 with version 4.01. It was replace by the XHTML family of languages.

impose *any* presentation decisions; stylesheets would do that… and the XML/HTML tags would simply indicate the meaning or purpose of the words within.

So, we have seen that XHTML may have been unfairly associated with the difficulties CSS experienced in its infancy. We have also seen that XHTML does not change HTML much more than requiring your markup to conform to the rules of XML. However, we have also seen that the Mangrove project embeds its semantic markup in a non-XHTML file and their system is able to find and process the required data just fine. So, why is XHTML important? First, it bears repeating that HTML is a dead language. XHTML is the current standard and, as we move into an era with more and more programmatic applications that require predictable, parseable, standards-compliant markup, using XHTML will become more and more important. In fact, the W3C intended XHTML as a transitional language for browsers to rely on until XML becomes the lingua franca of the Web.

But also, switching to the XML-conforming XHTML opens the whole world of XML-based languages to the author. For example, with Extensible Stylesheet Language Transformations (XSLT) one can specify replacements for tags or groups of tags in a source document, allowing for automated reformatting of documents. In the realm of dynamically generated Web content, having the power of the various XML manipulation libraries at your disposal is a powerful tool. And, though one might argue that browsers and Google are able to figure out what to do with broken HTML, search-engine optimizers actually recommend standards-compliant XHTML to help get better Google ratings. And, as it becomes more and more trouble to support the various bastardizations of HTML, browsers will gradually work their way around to being less forgiving about the quality of the markup. The browser argument is also profoundly shortsighted, since the basis for this "just-good-enough" attitude is generally the fact that it looks a certain way in a browser window. What appearance your words have in a browser, however, is completely irrelevant to whether they can be used by the upcoming wave of automated, Semantic-Web-oriented applications.

4. Enter the "lowercase semantic web"

Within cutting-edge Web circles, work is already underway to start creating some of these semantic-aware applications. One group of such "technorati" has dubbed this the "lowercase semantic web." As defined by a presentation given by Tantek Çelik and Kevin Marks at the O'Reilly Emerging Technology conference in February of this year, the lowercase web is [CM04]:

- simple semantics with microformats
  - don't try to "define the world"
  - small pieces loosely joined
- evolutionary not revolutionary
  - add semantics to today's Web rather than create a future Web
- user centric design
  - humans first, machines second
  - "people are helping to create metadata" (David Sifry)

In other words, rather than attempt to assemble the all-knowing, all-seeing, uppercase Semantic Web, somehow getting lots of people to create and use sophisticated ontologies, why not use little aspects of the existing Web languages to offer new somewhat semantic features?

To start with, the very names of the tags we use to mark up our content convey some semantics. This is a somewhat controversial issue, however, as what meaning is conveyed is still subject to misinterpretation. Also, have we really "taught" the machine the meaning of the word by labeling it? Uschold [Usc01] refers to a "semantic continuum", in which he refers to the identifier used for a mark up tag as "informal explicit semantics". To be certain, the tags offered by XHTML do not indicate the real-world semantics of their content, but declare the structural nature of the content; i.e. is this piece of content a heading (`<h1>`), a paragraph (`<p>`), etc. This does not mean, however, that there is no semantic indication to be gained from these tags. They are merely only useful in understanding the structure of a document. If the task at hand were to gather information about document structure, this markup would be indispensable.

Gaining insight into the structure of a document may be helpful for rendering that document, but it doesn't give any further insight into the meaning of the document's content. This is clearly a shortcoming of the structure-oriented nature of XHTML and a strong motivating factor for moving to XML for all Web documents. With XML, one can use more descriptive tags and can control the valid usage of these tags with a schema or document type declaration. Until Web browsers become more conversant in all the necessary XML nuances [5] however, we can use some features of XHTML to start including semantic markup.

A simple place to begin is by identifying the purpose of a hyperlink. This can by achieved with the use of the "rel", or relationship, attribute of the XHTML "a", or anchor, tag with which one makes a hyperlink. For example:

```
<a rel="license" href="http://creativecommons.org/licenses/publicdomain/">
```

Here we have a link to a Creative Commons license,[6] which identifies the property rights asserted for the document. By including this attribute on the link to the license, we open the door for aggregators to scan the Web for documents adhering to a particular license. This would mean that the program would not have to maintain a list of "hrefs" that point to licenses, but could just scan all "a" tags for the "`rel="license"`" attribute.

Vote Links are another use of the "rel" attribute, in which one characterizes one's feelings about the page at the other end of the link. According to the Vote Links specification,[7] three values are allowed: "vote-for", "vote-abstain", and "vote-against". In much the same way as the Creative Commons license indicator above, software applications can scan pages for only those links with a "`rel="vote-for"`" attribute in order to compile a list of links to documents with which the author agrees. The creators' documentation explains that "indexing and tracking applications treat all links as endorsements" and so they created vote links to specifically declare their intentions. Since this technique is very new, its applications are still limited, although there will likely be new uses to come.

---

[5] Current versions of most browsers will parse and present XML documents, and the content can be styled with CSS. However, browsers have not yet implemented the hyperlink and form technologies offered by XML (XLink and XForm).
[6] http://creativecommons.org/
[7] http://developers.technorati.com/wiki/VoteLinks

Another example of link classification is the XHTML Friends Network (XFN).[8] It occurred to the creators of XFN that if a link was to the website of a friend or acquaintance, the "rel" attribute could be used to let others know just how this person is related. In good Web standards form, the folks behind XFN took the time to define an accepted list of terms (e.g. "acquaintance", "friend", "co-worker", etc.) to avoid a term explosion similar to the effects on HTML of the browser wars.

Using some advanced CSS,[9] these links can be given a unique presentation based on the type of relationship. But, more importantly, Web aggregators like http://rubhub.com can be used to create views of the XFN relationships advertised by a site. Impressively, it doesn't just scan a page for "rel" attributes, it doesn't even simply scan a whole site for the XFN attributes, it crawls the Web for XFN tags that link back to the site for which you performed the search. Future applications will likely tie XFN into your digital address book, creating webs of relationships rather than just being a simple name and address database.

The Friend of a Friend project (FOAF)[10] has a similar intent to XFN. However, the similarities end with the intent. Rather than being based on a single tag attribute, FOAF is an entire RDF vocabulary. RDF (Resource Description Framework) is, as its name would suggest, a framework for describing resources. It uses the concept of a triple (subject, object, predicate) to indicate facts about the resource. For example, in "the author of this paper is Ben Munat," the subject is "the paper," the predicate is "author," and the object is "Ben Munat". FOAF uses RDF-XML, the XML syntax for describing the RDF data model, to allow users to describe themselves and their relationships with others. As it defines dozens of terms—"classes" and "properties" in RDF terminology—FOAF is clearly more robust than XFN. However, the power comes with a cost: you must understand RDF and XML concepts in order to write FOAF by hand. However, there are a variety of tools to make the creation of FOAF code painless. The comparative simplicity of XFN will likely lead to its more rapid adoption, although if some of the more popular content management systems incorporate FOAF into a suite of similar semantic markup tools, it could become quite popular.

It is interesting that both of the previous projects center on social networking concepts, which is currently a popular Web trend. The success of Friendster,[11] for example, prompted Google to create the invitation-only Orkut.[12] The popularity of these kinds of human interaction tools should not really come as a surprise. Outside of the business, scientific, and academic worlds, one could argue that social interaction (e-mail, instant messaging, etc.) is the most popular use for computers. It is only natural that the first stirrings of somewhat semantics-savvy Web applications would involve social interaction.

One of the most explosive Web-based social phenomena in recent years has been Web logs or "blogs". Though blogs may not specifically be a semantic activity, there are some blogging trends that are of interest here. Those who maintain their own blog and are actively interested in the blog community often feature a "blogroll" and/or syndication on their site. A blogroll is collection of links

---

[8] http://gmpg.org/xfn

[9] Identifying an element by an attribute other than "id" or "class" requires using the syntax [*attr-name*~="*value*"] and is supported in recent versions of Mozilla and Opera, but not the Web's most popular browser, Internet Explorer.

[10] http://www.foaf-project.org/

[11] http://www.friendster.com/

[12] http://www.orkut.com/

to other weblogs, which, when combined with many other inter-linked such lists of links, forms a network of online personalities. This is clearly another offshoot of the social networking phenomenon and can also make use of XFN and/or FOAF to further "semanticize" the nature of the links.

Syndication in the Web world most often takes the form of RSS, which stands for "Rich Site Summary," "Really Simple Syndication," or "RDF Site Summary," depending on who you ask.[13] RSS exploded onto the Web scene around the same time as weblogs and they've often gone hand in hand, although RSS and a few competing forms of syndication have become popular with news sites as well. To use RSS, a site author merely publishes a simple XML file with entries for various news stories on her site. There are many RSS aggregating applications freely available, and feed publishers can inform these aggregators about their offerings. There are also applications that will automatically create and maintain the RSS files, and many content management systems offer syndication features. Though it may not dip too deeply into the semantics of the information to which the feed refers, RSS is classic first cut semantic markup: it offers grouping by topic and identifies the key elements of the article.

Blogchalking,[14] another outgrowth of the blog phenomenon, involves using specific keywords on the XHTML "meta" tag to indicate information about the owner of the blog. It is a kind of "hobo language for bloggers." To use it, one would put a "meta" tag in the head of her page that would look something like this:

```
<meta name="blogchalk" content="country, state, city,
      neighborhood, language, name,
      gender, age, internet connection,
      hobbies" />
```

The italicized terms within the content attribute's value would be replaced with the author's personal information. Other bloggers can then search for blogs by authors from a particular area, with a particular interest, of a certain age, etc.

GeoURL[15] is another, similar new use for the "meta" tag. However, the only personal data about the author it offers is a latitude and longitude. For example:

```
<meta name="ICBM" content="46.6667, -64.8333">
```

The "ICBM" name of this tag is indeed a reference to the type of missile, coming from the black humor of old hacker lingo. The GeoURL site and other Web crawler tools can then make a map of people using this technique, allowing Web authors to see who is near them. Interestingly enough, this begins to break down the traditional illusion of anonymity inherent in the Web, albeit voluntarily.

---

[13] The RSS community has fractured into competing specifications and ideologies. See http://news.com.com/2009-1032_3-5059006.html for some background.
[14] http://www.blogchalking.tk/
[15] http://geourl.org/

Both blogchalking and GeoURL are examples of new uses of the controversial HTML "meta" tag. "Controversial" because these tags quickly became abused during the Web explosion and search engines started ignoring them as unreliable. But though search engines may be important for finding general information from the Web at large, it is a mistake to write off all use of "meta" tags. Searching for content is only one possible Web activity. Falling into the notion that a human searching for and reading content is the be-all of the Web is short sighted. Brooks [2003] acknowledges that metadata is still of value on the "closed Web". The author would argue that the various groups using the techniques described in this paper create de facto "closed Web communities". And, for the purpose of their activities in these groups, they use specific tools that *do* pay attention to *certain* tags.

There is, in fact, a proposal that could help rekindle interest in the "meta" tag. XHTML Meta Data Profile (XMDP) "seeks to define a metadata profile format using principles of simplicity, reuse, and minimalism."[16] It works by declaring a link to a profile as an attribute of the "head" element. Tools that wish to use the document's metadata would read the linked profile. This essentially provides a schema for the metadata.[17] Such a proposal is an important first step toward imbuing the "meta" tag with some "trustability" and allows for the writing of software agents that can use the contained metadata.

The original attempt at creating a "meta" tag specification is called the Dublin Core, after Dublin, Ohio, the city in which the specification was created. Originally formulated in 1995, the Dublin Core Metadata Initiative[18] identified fifteen acceptable values for the "name" attribute of the "meta" tag. It is a metadata "profile" that can be specified in the "profile" attribute of the "html" tag of an (X)HTML document, like the unofficial XMDP profiles above. In use, its "meta" tags look something like this:

```
<meta name="dc:title" content="Ben Munat" />
<meta name="dc:date" content="2004-05-14" />
```

As the search engines started ignoring "meta" tags, the Dublin Core floundered. However, the standard has not simply disappeared because it has seen little use. The tags are still available and have started showing up in other metadata frameworks. In fact, Dublin Core metadata tags are often used in RDF/XML files to label the essential pieces of data. Actually, it is ironic that search engines did not start using Dublin Core metadata, because it was the "keywords" meta attribute that spammers generally abused. This tag allowed authors to put in whatever terms their heart desired. The fifteen Dublin Core tag classifications label *specific* pieces of information that are arguably harder, or at least are more pointless, to spoof. Since the tags specify more accurately what piece of information its value identifies, it is of little use to put in inaccurate terms. It's not foolproof, but it's not anarchy either.

The final technology covered here has the awkward moniker, Gleaning Resource Descriptions from Dialects of Languages (GRDDL).[19] This W3C project proposes a "mechanism for encoding RDF

---

[16] http://gmpg.org/xmdp/description

[17] The XMDP proposal, however, seeks to keep all its elements in standard XHTML, and therefore uses an XHTML definition list (`<dl>`) element rather than an XML Schema.

[18] http://dublincore.org/

[19] http://www.w3.org/2004/01/rdxh/spec

statements in XHTML and XML to be extracted by programs such as XSLT transformations." It is a metadata profile—which is, as with previous examples, linked to with the "html" tag's "profile" attribute—that defines a way to link the XHTML document to XSLT transformations that will extract the metadata out of the document and render it as RDF. This allows a page author to simply include appropriate metadata and links to transformations with which agents can convert the metadata into RDF for further processing. Note that this in not limited to just Dublin Core or one particular type of metadata markup. Its feasibility has been demonstrated for the Creative Commons licenses, GeoURL, RSS, FOAF, and XFN. Although one could just as easily link to an RDF document that describes the content of the XHTML page, it is entirely feasible that one could glean the RDF from documents of which one was not the author. This could be a useful way to extract semantics from legacy documents, though there would obviously need to be some metadata in the document to extract, and the document would also need to be well-formed XHTML in order to be parsed for the transformation.

5. Conclusion

This last point—that a document must be well-formed XHTML in order to be parsed for a transformation—is ultimately one of the central issues of this paper. This is because, when we attempt to "bring structure to the meaningful content of the Web", when we try to extract a modicum of useful information from the mountainous flow of raw data, we need the help of our machines. And the machines can't help unless they can read the data we hand them. So, the struggle to build a Semantic Web is not so much about bringing meaning to the Web as it is about we humans agreeing on the methods and codes we use to label our intellectual artifacts and then programming our machines to analyze and synthesize this content.

It is certainly a valuable and noble pursuit to create ontologies or databases of terms and concepts. It is reasonable to believe that such research can lead to quite powerful analysis engines and the sort of all-knowing, mechanical oracle that has been pondered in science fiction. However, it is important to deal with the simple steps as well. There are billions of pages on the Web and the vast majority of them are mere chunks of text suitable only for display to humans. It was the intent of this paper to show how adding some useful semantic notation to a basic Web page need not be hard and can begin with the simplest of additions. This was demonstrated by a number of cutting edge techniques already being used by Web aficionados.

Attempting to create an all-encompassing worldwide web of semantic meaning is a monumental task, and, as with any difficult task, it is best approached in pieces. Each domain or field of interest can concentrate on creating standardized ontologies and creating tools to use them. But also, we can work to add simple semantic indicators wherever possible. This notion of a "lowercase semantic web" should not replace or distract from the more elaborate and sophisticated "official" Semantic Web, but should augment it. As Uschold explains in [Usc01], there is a semantic continuum for Web content. The techniques loosely grouped together under this "lowercase semantic web" label merely help push the less explicit towards the more explicit. And every little bit helps.

References:

Ber99 – Tim Berners-Lee. *Weaving the Web* Harper, San Francisco, 1999.

Ber01 – Tim Berners-Lee, James Hendler and Ora Lassila**.** *The Semantic Web* available at: http://www.sciam.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21

Bro02 – Terrence A. Brooks. *The Semantic Web, universalist ambition and some lessons from librarianship* available at: http://informationr.net/ir/7-4/paper136.html

Bro03 – Terrence A. Brooks. *Web search: how the Web has changed information retrieval* available at: http://informationr.net/ir/8-3/paper154.html

Bro04 – Terrence A. Brooks. *The Nature of Meaning in the Age of Google* available at: http://informationr.net/ir/9-3/paper180.html

CM04 – Tantek Çelik and Kevin Marks. *Real World Semantics* available at: http://tantek.com/presentations/2004etech/realworldsemanticspres.html

Ful03 – Sherrilynne Fuller, Debra Revere, Paul Bugnia, George M. Martin. *Telemakus: A Schema-based Information System to Promote Scientific Discovery*

McB02 – Brian McBride. *Four Steps Towards the Widespread Adoption of a Semantic Web* available at: http://www.hpl.hp.com/semweb/publications/Brian's%20sardinia%20paper.pdf

Usc01 – Michael Uschold. *Where Are the Semantics in the Semantic Web?* available at: http://lsdis.cs.uga.edu/SemWebCourse_files/WhereAreSemantics-AI-Mag-FinalSubmittedVersion2.pdf