UNIVERSITA' DEGLI STUDI DI NAPOLI FEDERICO II

Facoltà di Ingegneria
Corso di Studi in Ingegneria Informatica

Tesi di laurea

Tagging techniques for Search Engine Optimization

Anno Accademico 2012/2013

relatore
Ch.mo prof. Porfirio Tramontana

candidato
Russo Gianluca
matr. M63/90

Contents

List of Figures

Introduction

"*Find a physiotherapist with a rating of excellent or very good on trusted rating services who works in a clinic located in a range of maximum 20 miles from my location and whose services are covered by my medical insurance*" [1]. These are the kind of requests that we will be able to submit to the Web of the future. A Web thus conceived by its creator, Tim Berners Lee[1], who in 2001 laid the conceptual foundations of the Semantic Web.

Today, the World Wide Web is an ever growing information reservoir containing a substantial portion of human knowledge; it offers an open, decentralized (and, from some viewpoints, uncontrollable) environment in which anyone can publish all kinds of information, coupled with powerful search engines which find and rank relevant information. The result is that the current Web is an extremely useful space for the user to engage in research, learning, commerce, entertainment, communication and socializing.

For most users, the first and sometimes the only source that is used to answer a question, find an event or learn a new occurrence, is a quick search on the Web through a search engine such as Google. This way, the user expects to find the most significant and relevant documents related to the initial question. Moreover, the search is done by using keywords that the user associates with the context sought and summarize the basic and the most important concepts of the search that would have been expressed in natural language.

However, the obtained results are sometimes not in line with what is expected. The user may find pertinent information but this may not always be in line with what was initially required. Furthermore, the results are not direct and concrete answers, but instead a series of links to possible answers. The user must activate, therefore, a series of actions and filters in order to understand and extract the information that he considers most appropriate and relevant within those links.

[1]http://www.w3.org/People/Berners-Lee/Overview.html

The challenge for the natural transition from the current Web to the "*Web of the future*" is to convert existing and new Web content that can be understood by humans, into semantically enriched content that can be understood by machines. A Web where it is possible to search for information and content in natural language to obtain well defined and more relevant answers; the Semantic Web "*an extension of the current Web in which the information in given well-defined meaning, better enabling computers and people to work in cooperation*"[1]. More than 10 years after its conceptualization, the complete transition to Semantic Web has not yet been realized. Although many efforts have been made in this direction, much still remains to be done.

Most of the content on the Web has been and is still built only for human consumption (human-understandable), thus making it difficult to automate tasks by intelligent agents, such as automatic reasoning, inferential operations[2] and information retrieval in a quick and accurate manner. Most of the pages in the Web environment are written in HTML, which is oriented only to the visual formatting of information, without the use of Semantics , in other words, the attribution of meaning and explanation of the information described.

In this context, the Semantic Markup of Web documents is one of the real steps towards adapting Web content to the Semantic Web in a natural way, by keeping the content of the Web page visible and readable at the same time.

Over the years, several techniques and languages have been standardized in order to add Semantics to the content of Web pages and make them machine-understandable. Some of the earliest work not directly, but in a parallel way, built annotation structures stored in separate documents from HTML Web page itself. Often the result of this decoupling makes it difficult to integrate information and its consumption.

The more recent techniques for Semantic enrichment allow the published document to be enhanced with some special annotations embedded in the same document. This enables the description of entities that are found in the content, as well as the existing relationship.

Therefore, in this context, tools for describing and/or annotating documents play an important role. This is due to the fact that these tools make it possible to create the basis of shared knowledge that will enable machines to really understand the meaning of the content.

One of the various aspects related to the Semantic Web and directly connected to the machine-understandable content, is the fact that the search results related to these documents assume

[2]The Inference is the process under which a proposition accepted as true, switches to a second proposition whose truth is derived from the contents of the first. The Inference on the Semantic Web can be characterized by discovering new relationships.

greater importance and more meaning than traditional documents. In other words, since the machines - and in this case the search engines - have full knowledge of the information content of the document itself, they may provide more accurate results, with further descriptions and with better positioning in terms of ranking.

This task is implicitly dedicated to the practice of Search Engine Optimization (SEO). SEO describes the whole series of activities aimed to achieve better visibility and positioning of a Website thanks search engines being able to gather, analyse and read the content of the Website. However, the benefits of SEO can be observed from two points of view:

- user perspective - using the search engine that obtains among the top results more accurately and is of greater interest for the user himself;
- owner / creator of the content (Website) perspective - an increase in visits mainly due to a better visibility of the content itself, which results in many cases in a key business objective.

We can definitely say that the process of semantic annotation of Web pages is a important point for the Web, not only because it is at the heart of the transition from the current Web to the Web of the future, but also because it is key for those who use it daily. While it is a fundamental step for the construction of the Semantic Web, which lays the foundation for the shared knowledge understood by the machine, its benefits implicitly fall in the main objective of SEO techniques.

The operation of semantic annotation therefore, is among the many techniques that, operating directly on the page's content of the Website and defined as "on-page SEO", are able to achieve an improvement in the positioning of the contents in search results.

Objectives

This thesis aims to analyze two aspects of the semantic content life cycle on the Web. In particular the thesis focuses on the phase of semantic enrichment of Web documents, more precisely Web pages written in HTML. It aims to improve the quality of web content from the machine-understandable viewpoint, and to look at the impact that these documents currently have on search engines.

The main contribution of this thesis, regarding the first aspect, is the implementation of an integrated architecture to facilitate the creation of semantic enrichment of documents. This is done by adding annotations, so that the structured data can also be analyzed and exploited by information retrieval tools to obtain more accurate results. This is realised through the "*MicrodataSemantic*" tool developed in this thesis.

Aside from the above mentioned features, the tool allows a simplified way to perform validation of Web pages in order to verify the presence of semantic machine-understandable content-type. One of the goals is to evaluate the effectiveness of this tool by comparing, through the indexes of information retrieval, the result of semiautomatic semantic annotation generated by the tool, with the optimum entirely manually generated.

The second aspect examined is how the enriched documents with semantic annotations are processed to give more information to the user, once submitted to the search engines. This therefore adds value, not only regarding the ranking and then the positioning of the Web pages, but also in terms of information that can be displayed to the user in the search results.

Chapter organization

The present work, in addition to the introduction, is divided into five chapters each covering the following topics:

- Chapter 1 is dedicated to the basic concepts that will be used during this thesis and the topics involving the Semantic Web, such as its historical origin, architecture, principles, languages for semantic representation and the latest technologies for Web pages annotation;
- Chapter 2 is focused on the problem and the solution of Web documents semantic enrichment, analyzing the issue in detail and the methodology adopted in the thesis;
- Chapter 3 presents the tool developed for the semi-automatic annotation of Web content, MicrodataSemantic, illustrating through the methodologies of software engineering, the architecture and the main features. The reader can also find a description of the basic characteristics of the technology used;
- Chapter 4 provides a description of the case study, presenting the business environment in which this work was born and developed, describing which requirements to meet and

the results achieved through the use of classical and widely used metrics in the field of information retrieval. It also analyzed the SEO results achieved;

- Chapter 5 considers the work and the research developed, focusing on results and recommendations for future work.

Chapter 1

Context and background

This chapter describes the basic concepts used in the thesis by providing an overview of the technologies and languages that are the foundation of the Semantic Web. The chapter ends with the general description of the Search Engine Optimization techniques.

1.1 From a Web of documents to a Web of Knowledge

1.1.1 The Internet

Even though 34% of world population use the internet every day[1], only a few know about the origin and the historic development of this ever-present technology.

First, it has to be said that the Internet has its roots in the so called ARPANET, a project commissioned by the Advanced Research Projects Agency (ARPA) to study country-wide data communication. In 1969, ARPANET only consisted of four computers (called hosts) in different cities, that were connected by a network. ARPANET grew over the years and electronic mail became one of the first early applications. In 1973, ARPA introduced the "Internetworking" program with the goal of developing an open architecture network. Within this network environment, the different networks may have differing architectures but could interwork through a meta-level – "Internetworking Architecture".

[1]http://www.internetworldstats.com/stats.htm

A new network protocol was needed to support this architecture, which led to the creation of the Transmission Control Protocol (TCP) and Internet Protocol (IP), jointly known as (TCP/IP). TCP/IP is the low-level protocol used for most of the traffic on the Internet today. High level protocols such as the File Transport Protocol (FTP), TELNET, the Simple Mail Transfer Protocol (SMTP), and the Hypertext Transfer Protocol (HTTP), all rely on the foundations set by TCP/IP in order to transfer files, perform remote logins, transfer electronic mail, and exchange Web documents using the Internet[2].

1.1.2 World Wide Web

The World Wide Web (abbreviated simply as Web in the following) was the result of a radical new way of thinking about sharing information and data.

Since its birth and during its consolidation, the Web has completely revolutionized the way in which information and data are exchanged.

The basic idea of the web, according to his inventor Tim Berners-Lee was "*a common information space in which we communicate by sharing information*"[3]. Essentially, before the development of the web, there were other ways to communicate, with others standards like e-mail or ftp just to name some. But the difficulties consisted in having to create ad hoc channels to up transfer information through various computer applications, the non-interoperability of the information exchanged, the non-universality of the data used by the applications. This had resulted in the closure of communications, which were essentially bidirectional or directed to a restricted group of users. Obviously the internet and its use before the spread of Web, was mainly directed to government agencies and scientists, whom used this new paradigm of communication to exchange information and data.

The birth of the Web, or with the publication of the first Web page, dated 1990[4], meant implicitly to defining different standards on which the today's Web is still based[3]:

- A standard presentation of the data, the Hyper Text Markup Language HTML, a markup language that web browsers use to interpret and compose text, images and other content into visual or audible web pages;

[2]The birth of the Internet http://www.lk.cs.ucla.edu/personal_history.html
[3]http://www.w3.org/People/Berners-Lee/ShortHistory.html
[4]http://www.w3.org/Consortium/facts.html#history

- A standard protocol for transferring hypertext, the Hyper Text Transfer Protocol (HTTP), an application-level protocol with the lightness and speed necessary for distributed, collaborative, hypermedia information systems;
- A unique identifier for documents written in HTML, which is able of uniquely referential the information, Uniform Resource Locator URL.

In addition to these standards, Tim Berners-Lee developed the first web browser "WorldWideWeb", a point and click hypertext editor[5] and the first Web server that was used to deliver Web pages on the request to clients. After the introduction of the Web, there was a rapid explosion of Web server, which contains information in the form of html pages, and the introduction of new Web browser software with additional features. This, in conjunction with the spread of the personal computer, more and more accessible to the mass market, and the development of access technologies to Internet, has made possible to the Web to enter more deeply into everyday life.

To understand the dimensions of the Web, in 2008 the largest search engine Google had indexed more of 1 trillion (as in 1,000,000,000,000) unique URLs on the web[6]. Nowadays, it can be easily understood that the Web presents an ever growing reservoir containing a substantial portion of the human knowledge. All this is allowed by the easiness of content creation, publication and distribution.

This large amount of information has decreed what is now called the Web of Knowledge, the Web where all kinds of information, each data type, is available on the Web. However, the sheer scale of the Web, together with its decentralised, highly redundant and largely inaccurate nature, makes using the knowledge within rather cumbersome. This problem is often referred to as “information overload".

To some extent, the problem has been tackled by advanced technologies as the information retrieval, which power the nowadays Web search engines and make finding of resources relatively easy. Moreover, it is still not possible for applications to interoperate with other applications without some pre-existing, human created, and outside-of-the-web agreements as to the meaning of the information being transferred.

[5]http://www.w3.org/People/Berners-Lee/ShortHistory.html
[6]http://googleblog.blogspot.cz/2008/07/we-knew-web-was-big.html

1.1.3 Hyper Text Markup Language - HTML

HTML (Hyper Text Markup Language) is the standard language in which Web pages are written. HTML, in turn, was derived from SGML (Standard Generalized Markup Language), an international standard (ISO 8879) for the definition of device - and system - independent methods of representing information. In the Web context, standards are set by the W3C (World Wide Web Consortium); they are called recommendations, in acknowledgment of the fact that in a distributed environment without central authority, standards cannot be enforced.

HTML was primarily designed as a language for semantically describing scientific documents, although its general design and adaptations over the years have enabled it to be used to describe a number of other types of documents. HTML is essentially a text stream with special codes embedded. These codes, called tags, are identified by having angle-brackets that surround them.

An example HTML document is given in below:

Listing 1.1: HTML example

```
<HTML>
    <HEAD>
        <TITLE>Hello world!</TITLE>
    </HEAD>
    <BODY>
        <H1>Hello world test page</H1>
        <P>Welcome to the world</P>
        <DIV>
            <SPAN>AUTOHR:</SPAN>
            <SPAN>GIANLUCA <SPAN>
            <SPAN>RUSSO<SPAN>
            <P><A HREF="www.example.com">WebSite link</A></P>
        </DIV>
    </BODY>
</HTML>
```

One of the most important tag in HTML is the anchor tag, indicated by `<A>`. With the `<A HREF=...>` form, anchor tags create a hypertext link to another document by specifying a URL. These tags indicate that a Web browser should retrieve the document represented by the URL if the link is activated. It is one of the most important tag because realize the first basic idea of the Web, the linking between documents. Although HTML's tags are mostly presentation oriented, some tags were added to provide weak semantic information. HTML 2.0 introduced the `META` element and the `REL` attribute. The `META` element specifies metadata in the form of a name/value pair. A popular use for `META` was to indicate keywords, for example `<META name=''keywords'' content=''Semantic Web''>`, that could help search engines index the site. However, many sites began abusing the keywords by including popular keywords that did not accurately describe the site (this is known as keyword spamming). As a result, many search engines now ignore this tag.

Other `META` element that is useful for search engines and help them to understand what's the main content of the web page is `<META name=''description'' content=''This page speak about Semantic Web''>`. The `REL` attribute of the anchor (`<A>`) and link (`<LINK>`) elements names a relationship from the enclosing document to the document pointed to by a hyperlink; the `<REV>` attribute names the relationship in the reverse direction.

HTML 3.0 added the CLASS attribute, which could be used within almost any tag to create semantic subclasses of that element. Unfortunately, the semantic markup elements of HTML are rarely used, but even if they were, the semantics they provide is limited. To address the semantic limitations of HTML, Dobson and Burrill[4] attempted to reconcile it with the Entity-Relationship (ER) database model. This is done by supplementing HTML with a simple set of tags that define "entities" within documents, labelling sections of the body text as "attributes" of these entities, and defining relationships from an entity to outside entities. This was the first attempt to formally add structured data to web pages and thus presented an approach to a problem that was also a significant motivation behind the design of XML.

During 2004, Mozilla Foundation, Apple, and Opera Software in response to the slow development of W3C web standards and W3C's decision to abandon HTML in favour of XML-based technologies decided to form a community of people interested in evolving HTML. Their work produced the last revision of HTML, HTML5, adopted as working draft also from W3C[7]. This new version of HTML, that will be released as "Recommendation" in 2016 as planned from W3C, include many new features for markup the web pages, introducing new elements and attributes for multimedia content for example `<video>` and `<audio>` tag. Another new introduction of HTML5 is something that was missing in the previous versions of HTML, the adding of semantic to the content natively with explicit tags. This is done in HTML5 by using Microdata tags which will be explained in the next chapter.

[7]http://www.w3.org/TR/html5/

1.1.4 Extensible Markup Language - XML

Despite its popularity, HTML suffered from two problems. First, whenever someone felt that HTML was insufficient for their needs, they would simply add additional tags to their documents, resulting in a number of non-standard variants. Second, because HTML was mostly designed for presentation to humans, it was difficult for machines to extract content and perform automated processing on the documents. To solve these problems, the W3C developed the Extensible Markup Language (XML)[8]. XML allows angle-bracketed tags to be embedded in a text data stream and these tags provide additional information about the text. However, unlike HTML, XML does not provide any meaning for these tags. Thus, the tag `<P>` may mean paragraph, but it may mean part instead. An XML example is given below:

Listing 1.2: XML example

```
<?xml version="1.0" encoding="UTF-8"?>
        <users>
                <user>
                        <name>Gianluca</name>
                        <surname>Russo</surname>
                        <city>Prague</city>
                </user>
</users>
```

There are three kinds of tags in XML: start tags, end tags, and empty-element tags.

A start tag consists of a name and a set of optional attributes, surrounded by angle-brackets. Each attribute is a name/value pair, separated by an equal sign. An end tag consists of the name from a previous start tag, but preceded by a slash ("/") and cannot have any attributes. Every start tag must have exactly one matching end tag. Empty-element tags are like start tags, but don't have a matching end tag. Instead, an empty element is indicated by a slash just before the closing bracket. For example, `<IMG SRC=''photo.jpg''/>` would be an empty-element tag. The data from a start tag to an end tag comprises an element. An element can contain other elements, free text or a combination of the two between its start and end tags. A well-formed XML document contains exactly one top-level element, but can have an arbitrary nesting of elements within that element.

Although XML's flexibility makes it easy for authors to describe arbitrary content quickly and easily, this flexibility can be problematic for machine processing. Since XML cannot express

[8]http://www.w3.org/TR/WD-xml-961114.html

the meaning of tags, most processing applications require tag sets, whose meanings have been agreed by some standard or convention. To help with machine processing, XML allows grammars to be defined for XML tags. This information is contained in a document type definition (DTD)[9] that specifies valid elements, the contents of these elements, and which attributes may modify an element. DTD essentially define a context free grammar. An XML document that has an associated DTD and conforms to the rules defined in it, is said to be valid or a well-formed XML document.

Although a DTD provides syntax for an XML document, the semantics of a DTD are implicit. That is, the meaning of an element in a DTD is either inferred by a human due to the name assigned to it, is described in a natural-language comment within the DTD, or is described in a document separate from the DTD. Humans can then build these semantics into tools that are used to interpret or translate the XML documents, but software tools cannot acquire these semantics independently. Thus, an exchange of XML documents works well if the parties involved have agreed to a DTD beforehand, but becomes problematic when one wants to search across a set of DTDs or to spontaneously integrate information from multiple sources.

One of the hardest problems in any integration effort is mapping between different representations of the same concepts – the problem of integrating DTDs is no different. One difficulty is identifying and mapping differences in naming conventions. As with natural language, XML DTDs have the problems of polysemy and synonymy. For example, the elements `<PERSON>` and `<INDIVIDUAL>` might be synonymous. Similarly, an element such as `<SPIDER>` might be polysemous: in one document it could mean a piece of software that crawls the World Wide Web while in another it is referred to an arachnid.

An even more difficult problem is identifying and mapping differences in structure. XML's flexibility gives DTD authors a number of choices. Designers attempting to describe the same concepts may choose to do so in many different ways. In Listing 1.3, three possible representations of a person's name are shown. One choice involves whether the name is a string or is an element with structure of its own. Another choice is whether the name is an attribute or an element. One of the

Listing 1.3: Structural differences in XML representation

```
<!-- The NAME is a subelement with character content -->
       <PERSON>
              <NAME>Gianluca Russo</NAME>
       </PERSON>
```

[9] http://www.w3.org/TR/REC-xml/#dt-doctype

```
<!-- The NAME is a subelement with element content -->
       <PERSON>
              <NAME>
                     <FNAME>Gianluca</FNAME>
                     <LNAME>Russo</LNAME>
              </NAME>
       </PERSON>

<!-- The NAME is an attribute of PERSON -->
<PERSON NAME="Gianluca␣Russo">
```

reasons for these problems is the lack of semantics in XML. There is no special meaning associated with attributes or content elements. Element content might be used to describe properties of an object or group related items, while attributes might be used to specify supplemental information or single-valued properties. Once humans have identified the appropriate mappings between two DTDs, it is possible to write XSL Transformations (XSLT) stylesheets[5] that can be used to automatically translate one document into the format of another. Although this is a good solution to the integration problem when only a few DTDs are relevant, it is unsatisfactory when there are many DTDs; if there are n DTDs, then there would need to be O(n2) different stylesheets to allow automatic transformation between any pair of them.

Of course, the problems of mapping DTDs would go away if everyone could agree on a single universal DTD, but even at the scale of a single corporation, data standardization can be difficult and time consuming, data standardization on a worldwide scale would be impossible. Even if a comprehensive, universal DTD was possible, it would be so unimaginably large that it would be unusable, and the size of the standards committee that managed it would preclude the possibility of extension and revision at the pace required for modern data processing needs. In 2001 the W3C has released as "Recommendation" an alternative to DTDs called XML Schema[10]. XML Schemas provide greater flexibility in the definition of an XML application, even allowing the definition of complex data types. Furthermore, XML Schemas use the same syntactic style as other XML documents.

However, XML Schema only gives to XML an advanced grammar specification and datatyping capability, and still suffers from the same semantic drawbacks as DTDs. Often there is the need to use elements defined differently from DTD. It is possible that the name of an element

[10]http://www.w3.org/XML/Schema

present in a DTD is the same as an element of another DTD. This is the classic situation of collision between names of elements and attributes. This issue can be solved through the use of Namespace XML[11] by a URI, used in the XML document, when we refer to a certain element, eliminating the risk of ambiguity.

1.2 Introduction to Semantic Web

The term Semantic Web was proposed by Tim Berners Lee in 2001[1], as part of an article published by the magazine "Scientific American", in which the Web creator's ambition was to generate a sort of Internet "thinking", the most advanced hypothesis of collective intelligence.

The basic idea of Tim Berners-Lee and the W3C (of which he is the founder) is to evolve the current Web, not to replace it, by translating the concept of the Web from "machine readable" to "machine understandable", in other words to make the information understandable and actionable directly to the machines. The creation of a semantic network, as well as imagined by Lee, is designed to ensure that the computer will become able to handle the information automatically and "learn" to realize a series of processes in a precise, continuous and repeated way.

The goal is to create an environment in which the information is created in such a way as "*to develop effective cooperation between computers and people*"[1]. In Tim Berners Lee's conception, the term Semantic Web is associated with the idea of a Web in which intelligent agents interact each other. In other words, applications are able to understand the meaning of the on-line contents and therefore are able to guide the user directly to the information being sought, as well as replacing the user to perform certain tasks. An intelligent agent should be able to:

- Understand the meaning of documents and information resources on the network;
- Create routes, based on information requested by the user, then guiding them towards it (in some cases it can even replace the user);
- Move from site to site by connecting different elements of the information requested based on semantic and logical relations.

[11]http://www.w3.org/TR/1999/REC-xml-names-19990114/

In the definition of Berners Lee, the concept of semantic technology assumes a central importance since it allows to understand the real meaning of Web content considering the context in which they are inserted. The expression “semantic information”, in fact, indicates information that has a meaning, or that is inserted and referred to a well-defined context. Therefore, the development of semantic information online, is emerging as a key step in the direction of the Semantic Web and in the natural evolution of the World Wide Web, so that the machines are able to interpret, interact and understand each other’s.

The meaning that the machines will be able to process is obviously something different from what is put into a normal communication between humans; machines are certainly not capable to really understand what will be asked to them, but they must be able to recognize that information which are important for users, thanks to the software that should be implemented for this purpose. In this regard, the most important thing will be the way in which the information is structured.

Add meaning to the Web does not mean replacing the old Web with something new, but to create an infrastructure that is put on top of the current Web, an environment that is a basis platform for the search engines, for brokers of information and, finally, for intelligent agents. In general, the Web is configured as an environment for the information exchange, intended to be used by human beings; all documents published on the Web are designed to have the human being as an end user.

The content of a document is a text enriched with illustrations, images, and audio-video, made by the author of the document to convey some of his knowledge to the reader. The document thus, contains the author’s knowledge, information and data formalized through the use of formal grammar in a natural language. It means that documents assume a semantic value when they are interpreted by users. The activities of selection, contextualization and interpretation of unstructured content cannot be achieved by any computer.

In other words, the information is machine-readable, but not machine-understandable. In order for the "meaning" of the data transmitted on the Web is "understandable" by the computer, it is necessary to associate meta-information to them that describe the "semantic" content in an understandable way to computers. In this sense, metadata are information, structured to be processed by a machine, which describes a Web resource. Especially, the metadata are written with a syntax that makes them understandable to the machine.

In this context, then, semantic content means that content processable by computers, "machine-understandable": this does not refer to the semantics of natural language or artificial intelligence techniques, but the semantics of data, which consist in useful information, to ensure that while the machine receiving the data, using them in the correct manner.

In the article that defines the Semantic Web[1], the authors propose a first example that highlights special features and characteristics of the new vision of the Web: the baseline scenario is that two boys who use a software agents for searching a good physiotherapist on the Web.

The demands of the two young men, however, are complex: the therapist must be known for his professionalism and reliability, its outpatient clinic must be located near their home, its performance must covered by medical insurance. Software agents that use Semantic Web technologies are not only able to complete the task of searching for a physiotherapist according to the criteria given above, but are also able to refine the search itself to put at the disposal of two users, different alternatives, such as to compare the availability of the specialist for an appointment with their own personal agenda or to motivate the selection of the results produced by the research.

Through the current version of the Web, which is machine-readable, to achieve the same efficiency, the two users of the previous example, should manually activate different search operations, with the risk of not being able to find all the possible alternatives.

The current Web is indeed an amazing universe of information, in which the selection, manually or supported by search engines, requires a considerable effort from the user. As shown in the example above, then, the Semantic Web is intended to enable machine processing of information through the use of logic and semantics.

The challenge of the Semantic Web, in the end, is to find a logical language suitable for expressing both the data and the rules for the automated reasoning on data information. This language, must of course be able to coexist with other knowledge representation systems and must be sufficiently expressive to enable reasoning on the Web scale.

1.3 Semantic Web Architecture

In 1998, when Tim Berners Lee presented his strategy for the Semantic Web development, he also presented a proposal for a layered architecture, highlighting the necessary technologies (existing and to be developed) for its realization. An infrastructure such as the Semantic Web, which needs years to be fully defined, involves a gradual build, layer by layer.

Each layer must be built on the other in such a way that the upper level is interoperable with the lower levels. It is important, in fact, that at each step an agreement is reached between the various research groups, in order to have the same starting point in the construction of the successive layers. In building one layer of the Semantic Web on top of another, two principles should be followed:

- Downward compatibility. Agents, fully aware of a layer, should also be able to interpret and use information written at lower levels.
- Upward partial understanding. The agents, should be able to take at least partial advantage of information at higher levels. Of course, there is no requirement for all tools to provide this functionality; the point is that this option should be enabled.

While these ideas are theoretically appealing and have been used as guiding principles for the development of the Semantic Web, their realization in practice turned out to be difficult, and some compromises needed to be made. Fig.1.1 shows the "layer cake" of the Semantic Web in the first idealized version by Tim Berners Lee which describes the main layers of the Semantic Web design and vision.

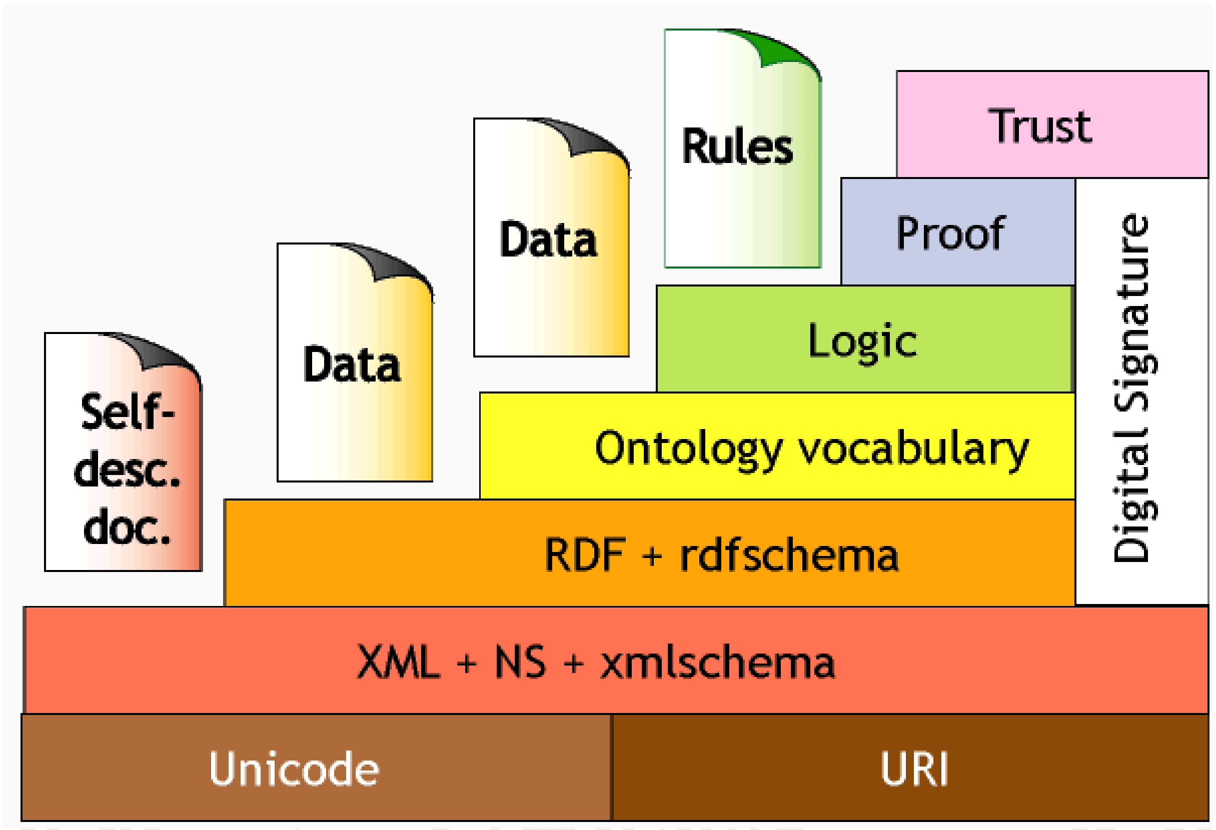


Figure 1.1: Semantic Web architecture

- Unicode – URI : at the core of the Semantic Web there is URI (Uniform Resource Identifiers) - the standard that is used to uniquely define resources on the web, such as documents, files and web page addresses - as well as the Unicode standard, a system of encoding that associates a number (or rather, a combination of bits) to each symbol, sign or character regardless from the program, the platform or the language used;

- XML - NS – XmlSchema: the upper level is the XML already described in section 1.1.4, which plays a primary role in the architecture of Semantic Web, along with the NS (namespaces) and XML Schema. XML is the language that brings with it some information on the semantics of objects, while the NS - identified by the URI - ensure interoperability between metadata dictionaries;

- RDF – RDF Schema[12] : Resource Description Framework and RDF Schema are the language for describing online resources and their types for semantic interoperability of data and information to enable intelligent agents to make logical inferences. The RDF

[12]http://www.w3.org/RDF/ - http://www.w3.org/TR/rdf-schema/

data model does not rely on XML, but RDF has an XML-based syntax. RDF Schema provides modelling primitives for organizing Web objects into hierarchies. RDF Schema is based on RDF and can be viewed as a primitive language for writing ontologies . But there is a need for more powerful ontology languages that expand RDF Schema and allow the representations of more complex relationships between Web objects;

- Ontology Vocabulary[13] : built on top of RDF, ontologies are metadata systems, corresponding to specific vocabularies for describing the relationships between the resources of the Web, allowing intelligent agents to interpret and understand the data.

The levels described above are currently already been standardized by the W3C. The layers that are based on these levels, however, are still under development and standardization which in the coming years could indicate the way forward for the representation of knowledge online in an orderly and efficient manner.

- Logic Level: through a universal language of heuristics, - i.e. procedures that will predict the results available on the web - assertions can be used to derive new knowledge. However it is a component still in development, since such a universal language does not yet exist. The Logic layer is used to enhance the ontology language and to allow the writing of declarative knowledge application-specific.
- Digital Signature: the authentication system of digital documents, is a significant component in the Semantic Web architecture. The digital signature may help to establish the origin of ontologies and deductions as well as data on the Web. The digital signature, in practice, certifies that a particular person wrote a particular document or statement. In this way, users who use such documents or instructions may be sure of their authenticity.
- Proof: The Proof layer involves the actual deductive process as well as the representation of proofs in Web languages (from lower levels) and proof validation, i.e. by proving the inferential logic that underlies the reasoning of the agent.
- Trust layer: The Semantic Web can be seen as a huge database of information where the user can access and extract data and new knowledge; the most important thing is that the user himself is able to distinguish reliable data from those unreliable. The information about the level of confidence or trust on data, can be somewhat explicit (this information

[13]http://www.w3.org/standards/semanticweb/ontology

comes from a source x and x is part of the trusted sources list), or can be derived automatically through intelligent agents based on the technologies of the Semantic Web (this data is from the source x, x is not part of the sources list that I consider reliable, but is in the list of trusted sources of the person y that I consider trustworthy). This deduction, or inference, can be more or less complex and may not be predictable beforehand. In any case, user could also understand how the level of trust of an online information resource has been calculated by the intelligent agents through the level of Proof. The Trust layer will emerge through the use of digital signatures and other kinds of knowledge, based on recommendations by trusted agents or on rating and certification agencies and consumer bodies. Being located at the top of the pyramid, trust is a high-level and crucial concept: the Web will only achieve its full potential when users have trust in its operations (security) and in the quality of information provided.

This is the classical layer of Semantic Web, as well as the first conception. An alternative architecture for the Semantic Web was proposed by several groups at the W3C Workshop in 2004/05 on Rule Languages for Interoperability and presented in the talk "Web for real people"[14] by Tim Berners-Lee . They argue that while a single-stack architecture would hold aesthetic appeal and simplify interoperability, such architecture is unrealistic and unsustainable. For one thing, it is presumptuous to assume that any technology will preserve its advantages forever and to require that any new development must be compatible with the old .

1.4 The principles of the Semantic Web

Before analyze the technologies and languages at the base of the Semantic Web, we focus the principles that characterize this newly revised version of the Web, taking into account the study carried out by Marja Ritta Koivunen and Eric Miller[15]. In particular, they found six principles that underlie the architecture of the Semantic Web and describe the essential characteristics:

Principle 1: Everything can be identified by URI's

As mentioned previously, this is the idea behind the Semantic Web, according to which, people, places, and things in the physical world should be uniquely identified with an unambiguous

[14]http://www.w3.org/2005/Talks/0511-keynote-tbl/
[15]http://www.w3.org/2001/12/semweb-fin/w3csw

name in the Semantic Web, through a well defined identifier. This identifier can be the URL of a web page, as well as the email address. The URIs therefore, represents the starting components of the Semantic Web infrastructure, to allow the machines and software to describe the semantics of resources available online and therefore to explain the content and information on the web.

Principle 2: Resources and links can have types

In the current web, the one in machine-readable version, the resources (i.e. content, information and documents on the network) and its links are the backbone of the online information area. However, resources and links are usually made to be understood by users and do not normally contain metadata that explain their content and their relationships with other web pages.

Users are able to interpret, understand and associate a specific meaning to the relations, to resources connected each other; without metadata, the nature of a resource, the logical and semantic relations with other resources are not understandable by machines. Computers therefore, require that on the resource is added a meta-information that would allow to establish "richer" relationships: “*A more relationship information could be for example*, `<depends on>`, `<is a version of>`, `<has subject >`, `<authors>`".

With the Semantic Web, resources and links acquire a "type", or a meta-information that associates a concept to each resource and each link; concept that can allow computers to understand the relationships between the logical and semantic resources, through explicit link. Fig.1.2 a) represent the current Web consists of resources and links, without meaning associated to each link or relation between documents or general resources on the web. Fig.1.2 b) shows instead the Semantic Web concept with meaning associated to each relationship between Web resources.

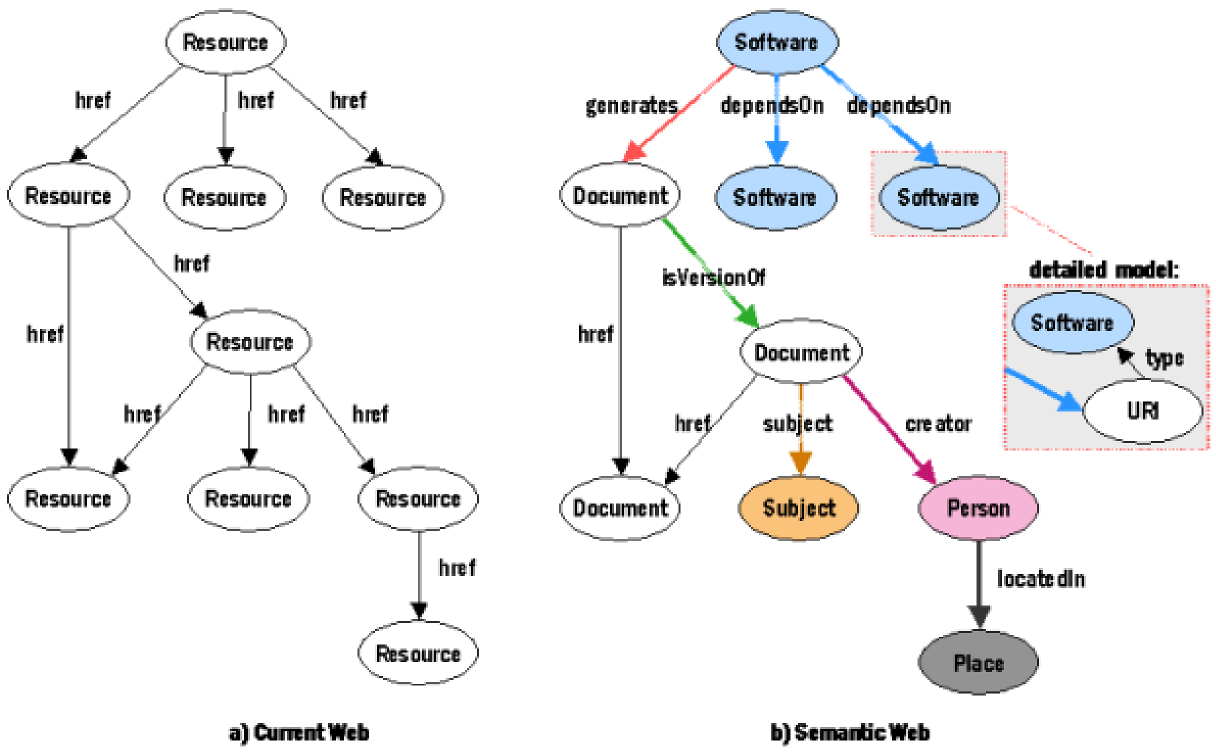


Figure 1.2: Current Web and Semantic Web

It is actually interesting to note that this conception of the Semantic Web is in fact already present since the first vision and proposal of the Web by Tim Berners lee in 1989, as shown in the first publication of the principles of web[16] and shown in Fig.1.3.

[16]http://www.w3.org/History/1989/proposal.html

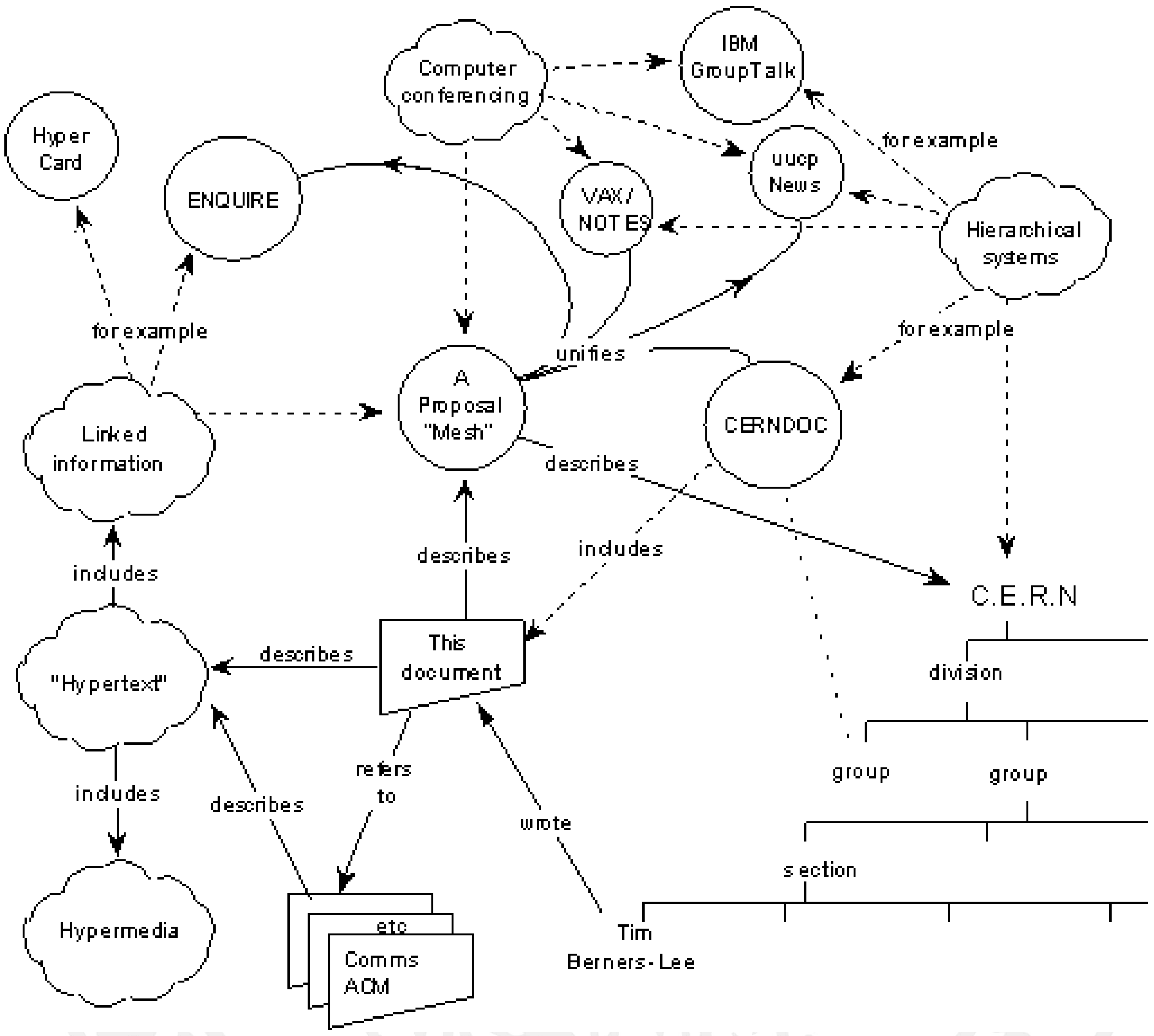


Figure 1.3: Original World Wide Web Proposal

Principle 3: Partial information is tolerated

The current web is democratic and decentralized, in the sense that any user can easily set up all the links that it considers appropriate in its Web resources, that is, content, information, and documents. However, often these hyperlinks are only "partial" and incomplete, that are ineffective and not valid from the semantic viewpoint (for example a link that connects a document dedicated to the Semantic Web with an image that represents the Web 1.0), as well as in some cases not more active. These links, therefore, likely to create ambiguity and semantic difficulties relevant to users. The Semantic Web will be democratic and decentralized, since all internet users will be able to create new documents and create links between resources. The information structure, then, will still be fragmented and "partial". The tools and technologies that will be developed for the effective realization of the Semantic Web, will be able to tolerate bias and incomplete information online and able therefore to understand the meaning of the relationship even if incomplete and inaccurate from the logic and semantics viewpoint.

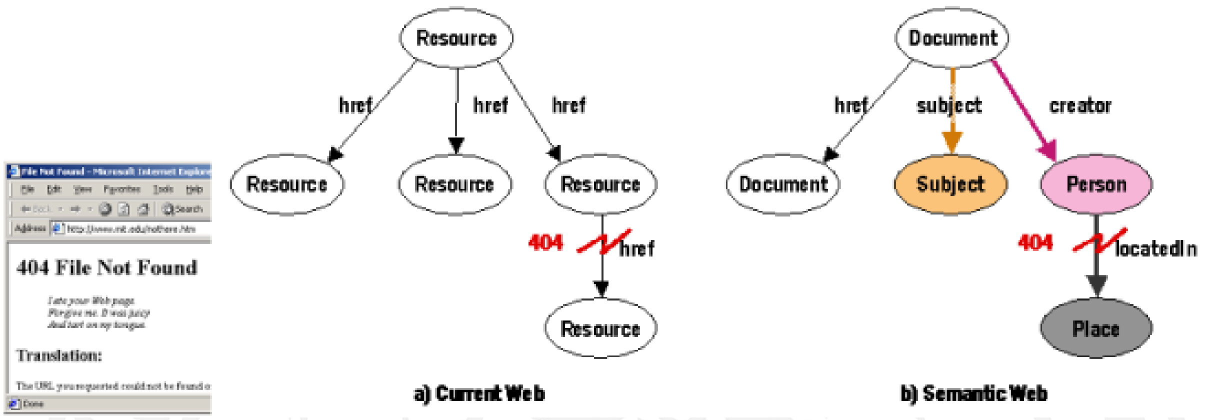


Figure 1.4: Both the current Web and the Semantic Web handle partial information

Principle 4: There is no need for absolute truth

The veracity of the information available on the Semantic Web will not be demonstrable at all, as is already the case for the current web. However, the Semantic Web applications will be based on potentially application that will be able to create a “trust machine” as defined by Berners Lee concept. Computers and software, therefore, will be able to check automatically, without user intervention, the reliability of documents and online content through interpretations and inferences.

Principle 5: Evolution is supported

As with the Web in machine-readable version, even in the Semantic Web documents, information and content will always be subject to evolutions and changes of meaning since the web is embedded within a social context. In addition, since the web is decentralized - and it will be also in machine-understandable version- for the same resource there can be several meanings; some may prevail over others, but in the absence of a central authority to standardize definitions and meanings for each resource, the Semantic Web should be able to handle all the different vocabularies put online by the different communities that can use them.

Ambiguity and uncertainty about the meanings described for various resources, must be managed in the best way by the technologies of the Semantic Web. The infrastructure of the web, defined as 3.0, therefore, must be able to manage the process of evolution and updating of information in the network and its meanings, not by replacing the new ones to the old ones, but by integrating and coordinating them.

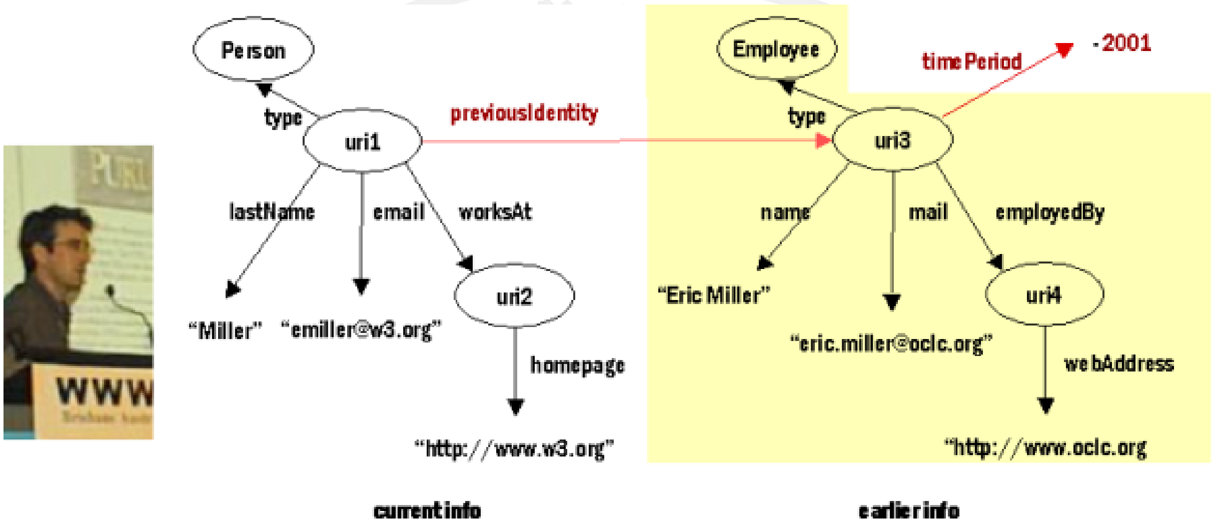


Figure 1.5: Combining new information with old when the old information cannot be changed

Principle 6: Minimalist design

The aim of the W3C activity is to standardize no more than is necessary. In the technical definition of the Semantic Web, the W3C aimed to propose infrastructure less standardized as possible of the Semantic Web, with the aim of facilitating the development of an open space to innovation and new developments online. As already pointed out, such an infrastructure refers to four levels (data, diagram, logical and ontological) that draw minimalist architecture of the web.

1.5 Semantic Web Languages

1.5.1 Markup concept

The term markup indicates a set of information regarding a text or individual elements that compose it. In general, the technique of composing text using markers or labels (tags) requires the definition of a number of conventions, in other terms a markup language for the documents. It is a language that describes the mechanisms of the text representation (such as colour, font and structure) which, through standardized conventions, can be used across multiple media and make explicit interpretations of the text itself.

The markup is separated from the content and can optionally contain itself. Markup languages can be different and are distinguished by different criteria, such as the representation of the text, the basic structure, the meaning of the elements that compose it and viewing or formatting. In the Semantic Web development, the ones most commonly used are descriptive markup languages, such as SGML, HTML and XML. These markup languages allow to separate the text (formatting) from the content (marking).

Generally what is produced with a descriptive markup language is not the document itself, but a source code file, which must then be interpreted by a software application (in case of HTML, the browser).

1.5.2 Metadata concept

The whole family of XML languages has the purpose of generating data that are not only intended for reading, but to be used by software and applications. XML meets the needs of syntactic standardization that can be used by different applications and platforms such as language interchange. However, the data that must be processed by computers and software, must contain information about the data, which specify the context and to describe the content. Such information, take the name of "metadata".

The meta-information in fact, allow users to specify information about the documents created or used, which are not only readable, but also interpretable by applications and by search engines. The term metadata literally means "data about other data". This definition is the most basic, but certainly the most explanatory of the concept: the metadata are in fact support information of primary information. Tim Berners-Lee formally defines metadata as "information about web resources understandable to computers."

Therefore, metadata play a fundamental role in the processes of obtaining and processing information. Furthermore, metadata may also be present and managed in three ways: in the documents themselves which they are associated; in different documents, which are immediately associated with the object to which they refer; in different documents that can be processed separately. Metadata, therefore, are potentially able to improve and optimize the content and resource on Internet.

Moreover, from a Web Semantic perspective, adding metadata on contents will give logical sense to them and would give to the machine the opportunity to reason about the data and interpret them. The most obvious benefit would be the ability to have more precise and effective search, no longer simply based on keywords, and a real and more deep sharing of available information between different Web applications.

However, the proposal represents a narrow definition of the metadata purposes, which conversely can be applied to different contexts and meet different needs. In fact, in order to facilitate and make more targeted search, metadata can and should be used to provide information on the resource, in order to ensure the usability in time.

1.5.3 Resource Description Framework - RDF

Metadata makes a crucial contribution to improving access to information. The effective use of metadata, however, requires to set the conventions for syntax, semantics and the structure of documents and on-line resources. Syntax, namely the systematic organization of data for automatic processing, facilitates the exchange and use of metadata between different applications.

The structure, however, can be seen as a constraint on the formal syntax for a consistent representation of semantics. In this regard, as already analyzed, XML provides semantic information using a mechanism for defining the structure of a document. Such meta-language defines a tree structure for documents, in which each node detects a well defined tag, whereby it is possible to interpret the information contained. The semantics of an XML document, however, is not explicitly specified, is "embedded" in tag names.

The basic tool for the encoding, exchange and reuse structured metadata - and thus to introduce the semantics of the documents within the Web – is RDF - Resource Description Framework, a child language of XML. It is designed with the aim of adding meta-information to web documents and generally to online resources. In this direction, RDF, allows the addition of semantic to the content and documents on the web, without making any assumption on the structure.

The main feature of RDF is to enable semantic interoperability between those applications that exchange information on the web. RDF allows therefore, the construction of the semantic structure within documents, overcoming the major limitation of XML, that is, highlight the meaning of documents and contents. In other terms, while XML provides a basic syntax of a document but does not refer to the semantics, RDF adds meta information to the documents themselves, providing information about the meaning and content.

In summary, XML supports syntactic interoperability, while RDF aims at semantic interoperability. In general, RDF is a framework that defines how the information can be represented on the web. "*RDF is a foundation for processing metadata; it provides interoperability between applications that exchange machine-processable information on the Web. Basically, RDF defines a data model for describing machine-processable semantics in data*"[17] .

The development of RDF within the Semantic Web was led to allow the following uses[18]:

- Web metadata: providing information about Web resources and the systems that use them (e.g. content rating, capability descriptions, privacy preferences, etc.)
- Applications that require open rather than constrained information models (e.g. scheduling activities, describing organizational processes, annotation of Web resources, etc.)
- To do, for machine processable information (application data) what the World Wide Web has done for hypertext: to allow data to be processed outside the particular environment in which it was created, in a way that can work at Internet scale.
- Interworking among applications: combining data from several applications to arrive at new information.
- Automated processing of Web information by software agents: the Web is moving from having just human-readable information to being a world-wide network of cooperating processes. RDF provides a world-wide standardized language for these processes.

Therefore, the design of RDF has aimed to achieve some objectives like establish a very simple data model, easy to process and manipulate by the applications; establish a formal semantics and inferences verifiable.

[17](Broekstra et al., 2002).
[18]G. Klyne, J. Carrol, Rdf concepts and abstract syntax, W3C Recommendation 10 February 2004 http://www.w3.org/TR/rdf-concepts/

1.5.4 Ontology

"*Ontologies are an essential backbone technology because they provide an important feature: they interweave formal semantics understandable to a computer with real world semantic understandable to humans*"[7].

Ontology is a term borrowed from philosophy that refers to the science of describing the kinds of entities in the world and how they are related. With particular attention to the computer, ontology is a description of the concepts and relationships between them. In this case, it is a document that indicates in a formal way, the meaning and the links between distinct terms. More precisely, introduced in a specific situation, it identify the aspects that are considered relevant in that context and those that can be ignored. It specifies the concepts and how they relate to each other, the properties and how these properties are connected by using inference and logic.

In other terms, the ontology allows to work with a structured set of concepts where relationships are clear and, above all, significant at the semantic level. Ontologies are probably the most important layer towards the construction of the Semantic Web. It is only with the creation of ontologies, designed as a structured set of definitions of vocabulary, may be possible the junction between the data layer and the top ones. Ontologies are the links that will allow to connect the current Web to knowledge representation structures based on meaning.

The importance of ontologies assume makes their development essential to the emergence of the Semantic Web since without them, the web miss the key element that should allow machines to process the formal meaning of the information. The actual creation of ontologies, and their implementation in real applications, will require the adoption of design methodologies, such as to consider the changing needs of evolving ontologies and their use in concrete situations.

The most urgent problem currently about ontologies (also in reference to the needs of the Semantic Web), is to ensure the evolution of ontologies when these become part of concrete applications. In this perspective, in fact, in order to make ontologies efficient in Semantic Web, they must inevitably undergo a process of changing and continuous updating, since the transformation of the shared knowledge of any concept is continues.

1.6 Search Engine Optimization - SEO

The term SEO, Search Engine Optimization, refers to the whole series of activities aimed at obtaining an improvement in the visibility of a website or a web page in a search engine's search results (typically defined as SERP Search Engine Results Page). These optimization techniques can be merged into a wider range of activities related to web marketing and therefore having very specific business purposes.

The birth of these techniques is relatively recent, mainly due to the exponential growth of web documents on the web and the difficulty for a content to be easily discovered by the users. Search is a global phenomenon. As of June 2012, the worldwide population of Internet users numbered over 2.4 billion[19], and more than 158 billion searches are performed worldwide each month as of May 2013[20], approximately 5.2 billion web searches are performed every day.

Search engines are at the centre of this disruptive event, and having a business's website rank well in the search engines when people are looking for the service, product, or resource it provides is critical to the survival of that business. The benefits directly related to the results that SEO techniques reach, are implicitly the higher rank in the SERPs, which involves a series of indirect benefits as:

- Increase in Traffic: which then results in one of the most common business goals to increase sales and in general the market share.
- Make web-pages accessible: one of the big benefits of search engine optimization is ensuring that the pages are accessible by the search engines and from the users.
- Better ROI than normal ADS: the best placement in the SERPs leads to the pages of the website, which are pre-selected by the search engines, those users who are really interested in that contents.

A 2006 study[8] answers the question of why being in the top positions leads to increased traffic and hence the benefits listed above. They conducted heat-map testing with search engine that produced interesting results related to what users see and focus on when engaged in search activity. As shown in Fig.1.6, the graphic indicates that users spent the most amount of time focusing their eyes in the top-left area, where shading is the darkest.

[19]http://www.internetworldstats.com/stats.htm
[20]http://www.comscore.com

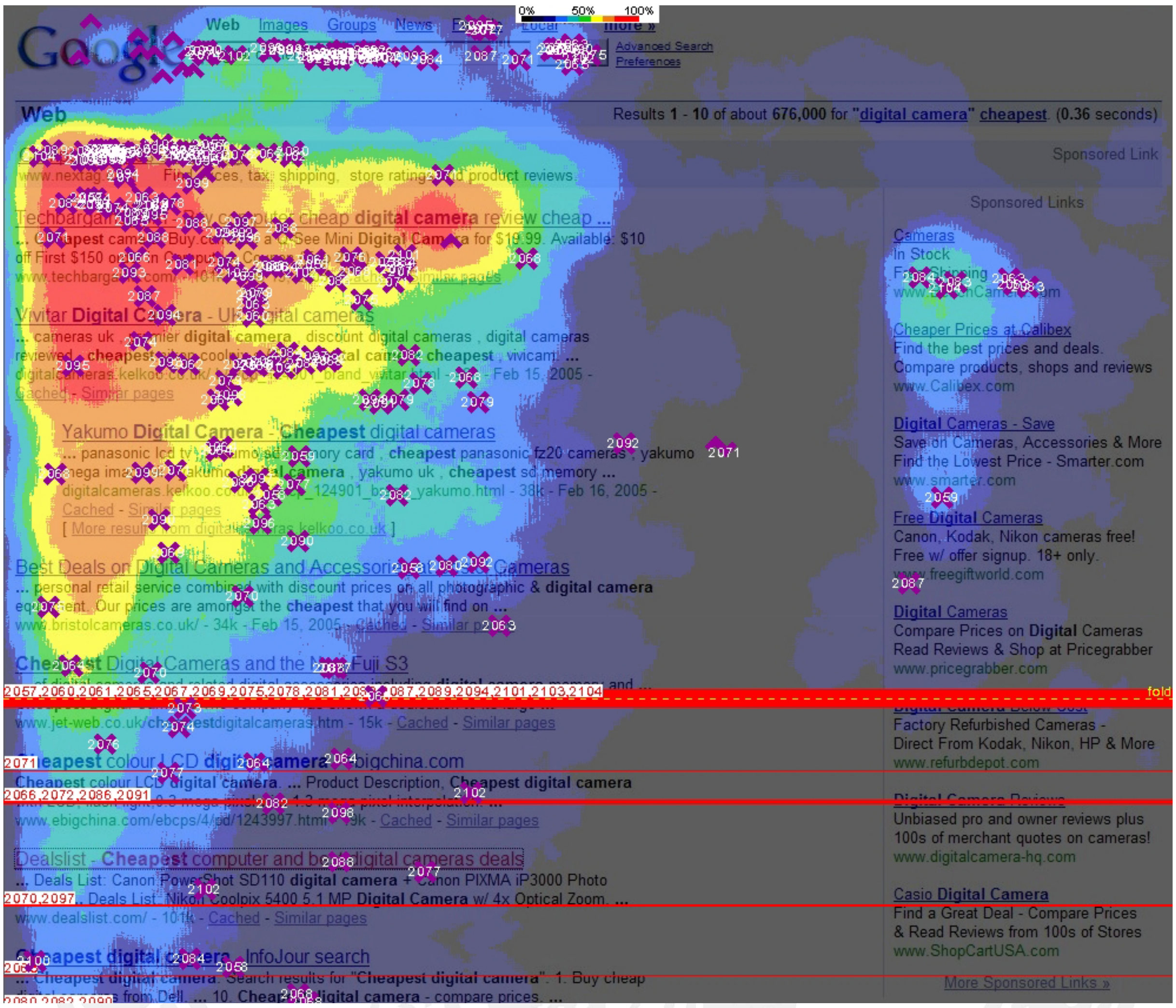
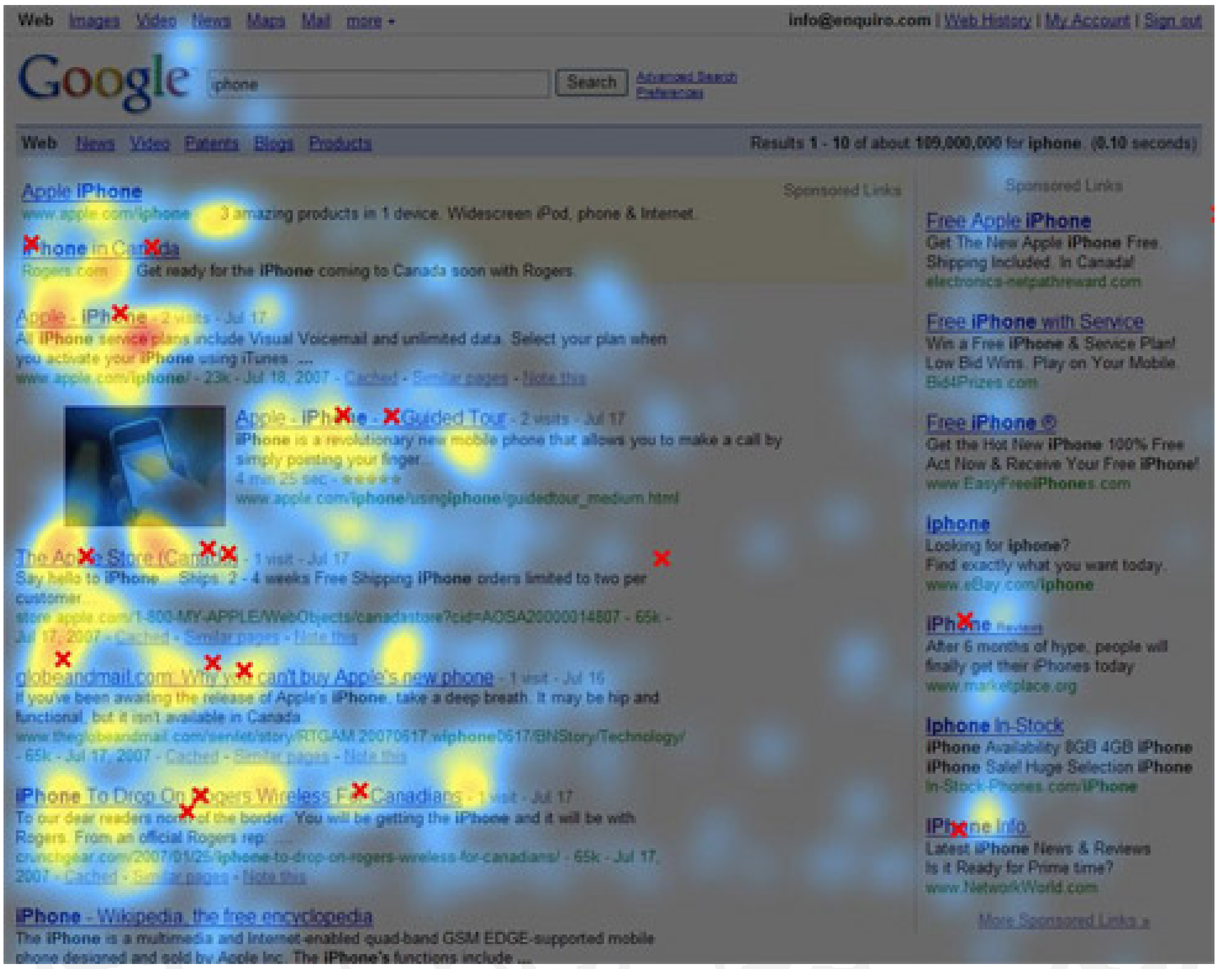


Figure 1.6: Search Engine Results Page heatmap

Moreover, in addition to focus the attention on the first results provided by the search engines, the study also has analyzed the impact of the results with more content and more descriptions, defined as Rich Snippets. The result of this study, shown in Fig1.7, creates more a chunking effect, where the chunks are around the various rich media objects, such as images , video or richer descriptions. Understandably, users focus on the image first. Then they look at the text beside it to see whether it corresponds to the image or video thumbnail.


Web Images Video News Maps Mail more ▾ info@enquiro.com | Web History | My Account | Sign out

Google iphone Search Advanced Search Preferences

Web News Video Patents Blogs Products Results 1 - 10 of about 109,000,000 for iphone. (0.10 seconds)

Sponsored Links

Apple iPhone
www.apple.com/iphone 3 amazing products in 1 device. Widescreen iPod, phone & Internet.

iPhone in Canada
Rogers.com Get ready for the iPhone coming to Canada soon with Rogers.

Apple - iPhone - 2 visits - Jul 17
All iPhone service plans include Visual Voicemail and unlimited data. Select your plan when you activate your iPhone using iTunes. ...
www.apple.com/iphone/ - 23k - Jul 18, 2007 - Cached - Similar pages - Note this

Apple - iPhone - Guided Tour - 2 visits - Jul 17
iPhone is a revolutionary new mobile phone that allows you to make a call by simply pointing your finger ...
4 min 25 sec - ★★★★★
www.apple.com/iphone/usingiphone/guidedtour_medium.html

The Apple Store (Canada) - 1 visit - Jul 17
Say hello to iPhone. ... Ships: 2 - 4 weeks Free Shipping iPhone orders limited to two per customer.
store.apple.com/1-800-MY-APPLE/WebObjects/canadastore?cid=AOSA20000014807 - 65k - Jul 17, 2007 - Cached - Similar pages - Note this

globeandmail.com: Why you can't buy Apple's new phone - 1 visit - Jul 16
If you've been awaiting the release of Apple's iPhone, take a deep breath. It may be hip and functional, but it isn't available in Canada
www.theglobeandmail.com/servlet/story/RTGAM.20070617.wiphone0617/BNStory/Technology/ - 65k - Jul 17, 2007 - Cached - Similar pages - Note this

iPhone To Drop On Rogers Wireless For Canadians - 1 visit - Jul 17
To our dear readers north of the border: You will be getting the iPhone and it will be with Rogers. From an official Rogers rep: ...
crunchgear.com/2007/01/25/iphone-to-drop-on-rogers-wireless-for-canadians/ - 65k - Jul 17, 2007 - Cached - Similar pages - Note this

iPhone - Wikipedia, the free encyclopedia
The iPhone is a multimedia and Internet-enabled quad-band GSM EDGE-supported mobile phone designed and sold by Apple Inc. The iPhone's functions include ...

Sponsored Links

Free Apple iPhone
Get The New Apple iPhone Free. Shipping Included. In Canada!
electronics-netpathreward.com

Free iPhone with Service
Win a Free iPhone & Service Plan! Low Bid Wins. Play on Your Mobile.
Bid4Prizes.com

Free iPhone ®
Get the Hot New iPhone 100% Free Act Now & Receive Your Free iPhone!
www.EasyFreeiPhones.com

iphone
Looking for iphone? Find exactly what you want today.
www.eBay.com/iphone

iPhone Reviews
After 6 months of hype, people will finally get their iPhones today
www.marketplace.org

Iphone In-Stock
iPhone Availability 8GB 4GB iPhone iPhone Sale! Huge Selection iPhone
In-Stock-Phones.com/iPhone

iPhone Info
Latest iPhone News & Reviews Is it Ready for Prime time?
www.NetworkWorld.com

More Sponsored Links »

Figure 1.7: Search Engine Results Page chunked heatmap

Also motivated by this study, SEO techniques studying either the characteristics of the search engines or analyzing the improvements on web pages, try to increase the ranking and the information displayed for Web pages in the SERP. SEO techniques can be divided into two categories[21]:

- On-Page SEO: covers all the operations and the changes made to the content of the web page or website in general.
- Off-Page SEO: covers the operations performed externally to the content of the web page or website, then without changing the content but indirectly contribute to the SEO objective.

[21]http://www.camic.cz/a950-il-tuo-sito-web-e-indicizzato/news.tab.it.aspx

Within SEO techniques, has to be considered that “Given the user’s query, over 200 signals (including the analysis of the site’s content and inbound links) are applied to return the most relevant results to the user”[22].

Since the final purpose of search engines is to provide more accurate results that best fit the needs of users, SEO techniques provide a good help to achieve this goal. A goal which is actually one of the main benefits of the Semantic Web. Then adopting the SEO techniques, on the one hand involves direct improvement for search results, on the other indirectly contributes to the construction of the Semantic Web.

[22]http://googlewebmastercentral.blogspot.it/2008/10/good-times-with-inbound-links.html

Chapter 2

Semantic annotation of Web pages

This chapter discusses the problem related to semantic annotation of web documents which constitutes the base point for the creation of shared and understood knowledge by the machines in the Semantic Web. First, the reader can find an in depth analysis of the aspects that are directly related to the semantic enrichment of web pages. Therefore, different types of annotation will be examined, as well as methods of markup and finally the associated limits and difficulties. In the following, there is an overview of the possible solutions for this problem, justifying the choice of one solution as the adoption of the thesis

2.1 The basic concepts and related problem

The dictionary definition of annotation is "*A comment (usually added to a text); the act of adding notes*". To annotate, then, is "*to supply critical or explanatory notes*"[1]. In the computer and information retrieval context, annotation consists of assigning a note to a specific portion of text. More specifically, in Semantic Web context, annotation is the process of inserting tags in documents to assign semantics to text fragments, allowing to create the document processable not only by humans but also by automated agents.

Semantic annotation of web documents is the only way to make the Semantic web vision a reality. It can be considered as the basic process towards the Semantic Web, without which its realization would not be possible. In SEO context, annotation helps search engines understand the information on web pages and provide richer search results.

[1]http://dictionary.reference.com/browse/annotation

Semantic annotations can be coarsely classified as being formal or informal. Formal semantic annotations, unlike informal semantic annotations, follow representation mechanisms, drawing on conceptual models using well-defined knowledge representation languages. Such machine processable formal annotations on web resources, can facilitate a number of important aspects of information management:

- For search and retrieval - enriching documents with semantic representations helps to create more efficient and effective search interfaces[2]. Ultimately, users are empowered to counter the increasing information overload and gain better access to relevant documents and answers related to their information needs. The result achieved from this benefit, falls within the SEO objectives.
- In information presentation - semantically enriched documents can be used to create more sophisticated ways of flexibly visualizing information.
- To realize semantic maps and links between objects and entities that can not only improve and enhance search capability, but provide new ways and approaches to extract new information[3].
- The ability to create social maps based on the relationship of contents[4].
- For information integration - semantically enriched documents can be used to provide unified views on heterogeneous data stored in different applications by creating composite applications.
- For reusability and interoperability - enriching documents with semantic representations facilitates exchanging content between different systems.

Considering therefore an unstructured web document, the starting point for semantic annotation requires a number of prerequisites:

- A taxonomy defining the entity classes. It should be possible to refer to those classes;
- Entity identifiers which allow them to be distinguished and linked to their semantic descriptions;

[2]https://support.google.com/webmasters
[3]http://www.google.com/insidesearch/features/search/knowledge.html
[4]http://ogp.me/

- Knowledge base with entity descriptions.

A more accurate distinction can be made according to the storage location of the annotations and their level of automation. Based on the storage location it is possible to identify two categories of procedure for making annotations, Internal-Embedded or External-Linked. There are a number of arguments regarding this, for the choice of one or the other technique and there are different viewpoints that highlight the problems and limitations that each choice can lead to.

2.1.1 Internal-Embedded annotation

In this kind of Semantic annotation, the process of creation and addition of Semantic to the content is located within the same Web Page that is elaborated. This means that an information that describes the meaning to the web page content is directly added, through the use of metadata.

One of the negative aspects that come up from the adoption of this solution, requires the user to have writing access to the original page to be able to annotate it. This is hardly a realistic requirement on the Web, because most of the time the content is inserted through some integrated management systems, such as the CMS.

According to [2] it is possible to list other points against this choice:

1. Embedded encoding of information will increase the complexity of authoring Web content;
2. An increase of maintenance costs when keeping Web pages, with high amount of semantic content, up-to-date;
3. Bulky annotations will increase download times for all applications, even those that do not need to (or cannot) process the semantic annotations. The operation of inserting metadata within the content of the web page, implies the changing of the web page structure, which is built according to the formatting rules of HTML, and then could lead in some cases to visualization issues.

The operation of inserting metadata within the content of the web page, implies the changing of the web page structure, which is built according to the formatting rules of HTML, and then could lead in some cases to visualization issues.

2.1.2 External-Linked annotation

Unlike the previous method, semantic markup of web documents is carried out separately from the document itself. This means that the annotations reside in a separate file, but directly linked to the original document.

This approach has the substantial advantage of not requiring any modification of existing Web documents that are already published as HTML. Furthermore, the treatment of the information described within the external document can be thought also for other types of applications.

This technique could facilitate the addition of semantics to such web pages as they do not require access for editing. Automated tools then, can add semantics independently and then allow the connection with the original documents.

Although this solution has several advantages, it brings different aspects that limit, on several occasions, its use. First of all, it requires a detailed knowledge of how to use and create this type of documents. This means that anyone who creates a new web page, must have the requirement of creating this new semantic document; then connect this to the original document and upload it to an infrastructure that allows its consumption. Secondly there is a need for an infrastructure on the server to handle requests for that document.

These limitations may be overcome, but there are some obstacles more difficult to deal with. For example, the difficulties and the requirement of keeping semantic descriptions aligned for those web pages that are modified or updated. An example is the situation where there is an upgrade of a large amount of web pages, for which a manual updating operation of external documents is unsustainable.

The real limitation is the creation of these external documents for those web pages whose contents are dynamically created from the server.

Another strong aspect that limits this choice is that the search engines can only process the semantics added contextually to the content of the document. This means that nowadays only particular applications, especially designed for the treatment of this type of documents, are able to access and extract the semantic content.

Basing the distinction of the annotation type related to the level of automation, we can identify two categories: Direct or manual annotation and Automated annotation .

2.1.3 Direct or Manual annotation

The user makes annotation directly on given content by using his knowledge. The degree of automation is zero; the user is responsible for analyzing the documents, trying to identify semantic entities and relations between them.

This approach requires a thorough knowledge of the scope, of the language and technologies that the user must use to add semantics to the document. Obviously, the main limitation that involves the choice of manually enriching web pages is due to the fact that this procedure is highly complex and laborious.

An example is the manual annotation of a huge number of documents or for those documents that have a high number of entities overlapped inside them. This adoption in addition to being time-consuming, also leads to the possibility of errors without the regular use of verification and semantic validation tools.

From the study conducted by [9]"The results of our user study indicated that not every domain expert is a good annotator due to difficulties in the understanding of the conceptual model of semantic annotations". Therefore, it is unlikely that, the manual semantic annotation of web pages, carried out by conventional creators/authors of web content, can have the result of a complete semantic enrichment without creating disambiguation and errors

2.1.4 Automated annotation

An automatic or semiautomatic process generates the annotations. By some means or other, the process identifies semantic entities and relations in the source content and makes correspondences with ontologies.

An approach based on a fully automatic annotation, with almost non-existent human intervention, would make the transformation and enrichment of web pages contents a solved problem from the scratch. Obviously, though this approach could be fully achieved in the coming years, nowadays it is a hardly practicable technique.

Current practices, that attempt to address this problem in a fully automatic way, rely on different sectors of information retrieval, such as text mining or natural language processing.

Some tools such as SemTag[10] OntoNEO[11] SCORE[12], trying to reach this goal, which in some cases can be considered satisfactory, but used extensively, fails to provide a complete support to semantic enrichment. Although these automated tools, are able to work well for limited concepts or for those which refer to predetermined structures, fail to explore and extract the concepts and relationships that are expressed in natural language.

2.2 Technological solutions

In the Semantic Web vision, as said before, the goal is to add meaning into the current Web, so machines can understand its contents. Based on the Semantic Web Stack, RDF can be used to express the meaning of a web document in a machine-understandable way.

More specifically, for a given Web Document, it is possible to create a set of RDF triples to describe its meaning and then publish this on the web. More deeply, to finish a markup process, it is a necessary to first create a collection of RDF statements to describe the meaning of a web document, then put them into a separate file and finally link somehow the original Web document to this RDF file.

The use of RDF therefore, falls into the category of semantic annotations made through the use of External-Linked techniques. Again, this technique, as previously stated, does not allow search engines to extract semantic information of the content and then make this choice unsuitable for the purposes of the thesis.

The more recent techniques for Semantic enrichment adopt the use of the Internal annotation. Although it has several limitations, the Internal annotation is able to overcome the difficulties that are encountered when using the other technique.

Several languages and techniques, which fall into the this category have been presented over the years. They somehow are able to limit, and in some way to eliminate the related restriction presented in the previous paragraph.

The languages that are currently standardized by the W3C for adding meta-information to the descriptive content of web pages are Microformats, RDFa and Microdata. They perform the same task using different approaches.

The Microformats are simple predefined formats to add meta information to elements and attributes into the standard markup languages.

RDFa (Resource Description Framework in Attributes), as the name implies, provides a set of attributes to include RDF metadata directly in HTML, XHTML, and in different types of XML-based documents.

Microdata is a WHATWG HTML specification used to nest semantics within existing content on web pages. Microdata has become part of the HTML 5 specification to support the semantic annotation of web pages in a natural way and without changing the rendering of the pages themselves.

2.2.1 Microformats

Designed for humans first and machines secondly, Microformats are a set of simple, open data formats built upon existing and widely adopted standards[5]. Microformats are community-driven vocabulary agreements for providing semantic markup on Web pages, to describe a specific type of information —for example, a review, an event, a product, a business, or a person.

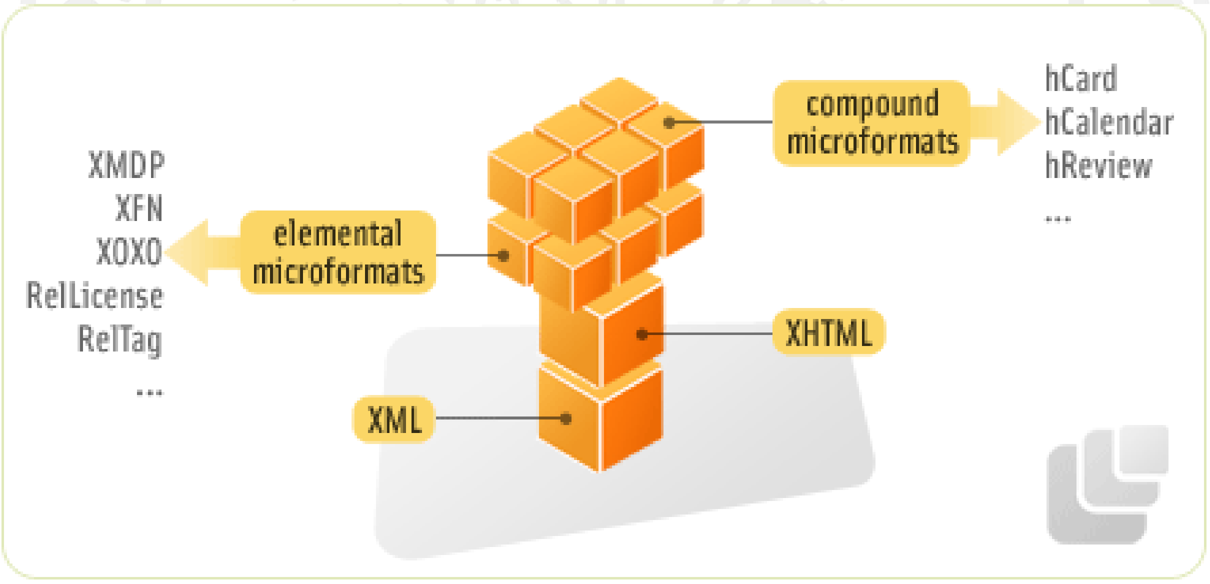


Figure 2.1: Microformats

Microformats are not a formal specification of the Semantic Web. But because they embrace the idea of information for both humans and machines, they support the Semantic Web goal of human and machine consumption. Yet unlike Semantic Web, which advocates separate machine

[5]http://microformats.org/about

versions of content, Microformats encourages making human-readable content more machine-readable[14].

Today there are seven stable standard Microformats specifications:

- hCalendar for events
- hCard for people, places and organizations
- rel-license for licensed content • rel-nofollow for limiting user-agent link on hyperlinks
- rel-tag for indicating a link destination is to a page about a keyword (tag) for the current page
- XFN, social relationship in links
- XOXO (Extensible Open XHTML Outlines) for indicating outlines and blogrolls.

Moreover, there are several drafts which are still making their way through the process to become specification:

- adr - address location information
- geo - latitude & longitude location (WGS84 geographic coordinates)
- hAtom - blog posts and other date-stamped content
- hListing - listings for products or services
- hMedia - media info about images, video, audio
- hNews - news articles, extension of hAtom
- hProduct - products
- hRecipe - cooking+baking recipes
- hResume - individual resumes and CVs
- hReview - individual reviews and ratings
- hReview-aggregate - aggregate reviews and ratings

- rel-author - link to the author's home page (from an article)
- rel-home - link to the homepage of a site
- rel-payment - link to a payment mechanism

In general, Microformats use the class or rel attribute in HTML tags (often <*span*> or <*div*>) to assign brief and descriptive names to entities and their properties. Here's an example of a short HTML block showing basic contact information for Gianluca Russo.

Listing 2.1: Simple HTML

```
<div>
       <img src="www.example.com/gianlucarusso.jpg" />
       <strong>Gianluca Russo</strong>
       Computer Engineering student at Federico II
       0000 Piazzale Tecchio
       Napoli , AZ 12345
</div>
```

Here is the same HTML marked up with the hCard[6] (Person) Microformats:

Listing 2.2: HTML marked up with Microformats

```
<div class="vcard">
       <img class="photo" src="www.example.com/␣gianlucarusso.jpg" />
       <strong class="fn">Gianluca Russo</strong>
       <span class="title"> Computer Engineering student</span> at <span class="
          org"> Federico II </span>
       <span class="adr">
              <span class="street-address">0000 Piazzale Tecchio </span>
              <span class="locality"> Napoli </span>,
              <span class="region">AZ</span>
              <span class="postal-code">12345</span>
       </span>
</div>
```

Microformats opened the way to make the content of web pages more understandable from machines (search engine for example), but it does of course have its weaknesses.

[6]http://microformats.org/wiki/hCard

The first, is that they reuse many of the HTML attributes (for example class, title) to carry semantic information, in such a way that they can interfere with the normal use of the attribute. The second weakness is that mixing vocabularies starts to get messy; because each Microformat is a combination of vocabulary and syntax, then it is actually quite specific, and they can therefore interfere with each other. The third weakness is that, because each vocabulary also requires 'syntax', then even if a perfectly usable vocabulary exists, it has to be 'converted' to be a Microformats[13].

2.2.2 RDFa

The power of RDF can be exploited through external files written in rather complex syntax. To embed, however, in a more simple and natural way metadata within Web documents, W3C has standardized the RDFa[7] .

RDFa (or Resource Description Framework – in – attributes) is a W3C Recommendation that embeds RDF triples in HTML documents. The RDF data is not embedded in comments within the HTML document, as was the case with some early attempts to mix RDF and HTML, but it is interwoven within the HTML Document Object Model (DOM)[15].

This means that existing content within the page can be marked up with RDFa by modifying HTML code, thereby exposing structured data to the Web. A detailed introduction into RDFa is given in the W3C RDFa Primer.

The actual version by W3C is RDFa 1.1, which has reached recommendation status in June 2012[8]. It differs from RDFa 1.0 in that it no longer relies on the XML-specific namespace mechanism. Therefore, it is possible to use RDFa 1.1 with non-XML document types such as HTML 4 or HTML 5. RDFa shares some of the same goals with Microformats.

Whereas Microformats specify both syntax for embedding structured data into HTML documents and a vocabulary of specific terms for each Microformats, RDFa specifies only a syntax and relies on independent specification of terms (often called vocabularies or taxonomies) by others. RDFa allows terms from multiple independently-developed vocabularies to be freely intermixed and is designed so that the language can be parsed without knowledge of the specific vocabulary being used[9].

[7] http://www.w3.org/TR/xhtml-rdfa-primer/
[8] http://www.w3.org/TR/2012/REC-rdfa-core-20120607/
[9] http://www.w3.org/TR/rdfa-core/

RDFa 1.1 specification defines several standard attributes :

- about – a URI or CURIE specifying the resource the metadata is about
- rel and rev – specifying a relationship and reverse-relationship with another resource, respectively
- src, href and resource – specifying the partner resource
- property – specifying a property for the content of an element or the partner resource
- content – optional attribute that overrides the content of the element when using the property attribute
- datatype – optional attribute that specifies the datatype of text specified for use with the property attribute
- typeof – optional attribute that specifies the RDF type(s) of the subject or the partner resource (the resource that the metadata is about).

Consequently, adding a semantic markup to web content means to describe all the information inside (or most of it). These information types are called entities or items. Each entity has a number of properties. For example, a Person has the properties: name, address, job title, company, and email address.

In general, RDFa uses simple attributes in XHTML tags (often <span> or <div>) to assign brief and descriptive names to entities and properties. Here's an example of a short HTML block showing basic contact information for Gianluca Russo.

Listing 2.3: Simple HTML

```
<div>
My name is Gianluca Russo but people call me Lucas. Here is my home page: <a
   href="http://www.example.com">www.example.com</a>. I live in Italy, and I am
    an engineer student at Federico II University.
</div>
```

Here is the same HTML marked up with RDFa.

Listing 2.4: HTML marked up with RDFa

```
<div xmlns:v="http://rdf.data-vocabulary.org/#" typeof="v:Person">
       My name is <span property="v:name"> Gianluca Russo </span>,
       but people call me <span property="v:nickname"> Lucas </span>.
       Here is my homepage: <a href="http://www.example.com" rel="v:url">www.
           example.com</a>.
       I live in Italy, and I am an <span property="v:title">engineer student </
           span>
       at <span property="v:affiliation"> Federico II University </span>.
</div>
```

The example begins with a namespace declaration using xmlns. This indicates the namespace where the vocabulary (a list of entities and their components) is specified. It is possible to extend the description environment by using more than one vocabulary.

Moreover, RDFa is designed and is officially specified to work in a variety of different languages including HTML5, XHTML1, HTML4, SVG, ePub and OpenOffice Document Format. Having a structured data syntax, which supports as many Web document formats as possible, is good for the web because it reduces the tooling necessary to support structured data on the Web.

On the basis of RDFa 1.1, a light version has been developed that is leaner than RDFa which has been standardized by W3C: RDFa 1.1 Lite[10]

"RDFa Lite is a minimal subset of RDFa, the Resource Description Framework in attributes, consisting of a few attributes that may be used to express machine-readable data in Web documents like HTML, SVG, and XML. While it is not a complete solution for advanced data markup tasks, it does work for most day-to-day needs and can be learned by most Web authors in a day. The full RDFa syntax provides a number of basic and advanced features that enable authors to express fairly complex structured data, such as relationships among people, places, and events in an HTML or XML document. Some of these advanced features may make it difficult for authors, who may not be experts in structured data, to use RDFa. This lighter version of RDFa is a gentler introduction to the world of structured data, intended for authors that want to express fairly simple data in their web pages. The goal is to provide a minimal subset that is easy to learn and will work for 80% of authors doing simple data markup."

[10]http://www.w3.org/TR/rdfa-lite/

An example of RDFa Lite 1.1 is given below:

Listing 2.5: RDFa Lite 1.1 example

```
<div vocab="http://schema.org/" typeof="Person">
       <a property="image" href="http://example.com/images.png">
       <span property="name">Gianluca Russo</span></a>,
       <span property="jobTitle">Founder/CEO</span>
       <div>
              Phone: <span property="telephone">123456789</span>
       </div>
       <div>
              E-mail:<a property="email" href="mailto:mail@email.com">
                  mail@email.com </a>
       </div>
       <div>
              Links: <a property="url" href="http://example.com␣/">Example.com</
                  a>
       </div>
</div>
```

2.2.3 Microdata

As briefly mentioned in the HTML paragraph, with HTML 5 standardization effort in 2009 supported by WHATWG, Microdata standard was introduced as an alternative proposal for embedding structured data into Web pages. Microdata is an attempt to provide a simpler approach than RDFa and Microformats for annotating HTML elements with machine readable tags and with an unambiguous parsing model.

It allows the use of any vocabulary, similarly to RDFa and instead of using the attribute from HTML as Microformats, it defines five new HTML attributes[11] :

- itemscope: Creates the Item and indicates that descendants of this element contain information about it.
- itemtype: A valid URL of a vocabulary that describes the item and its properties context.

[11]http://www.whatwg.org/specs/web-apps/current-work/multipage/microdata.html#microdata

- itemid: Indicates a unique identifier of the item.
- itemprop: Indicates that its containing tag holds the value of the specified item property.The properties name and value context are described by the items vocabulary. Properties values usually consist of string values, but can also use URLs using the a element and its href attribute, the img element and its src attribute, or other elements that link to or embed external resources.
- itemref: Properties that are not descendants of the element with the itemscope attribute can be associated with the item using this attribute. It provides a list of elements ids (not itemids) with additional properties elsewhere in the document.

Microdata also comes at the cost of brevity - Microdata-enriched markup can bloat up a bit more than Microformats-enriched markup[16]. Microdata is also a bit more structured than Microformats. For example, it's common to have one entity nested inside another. Microdata uses these simple attributes in HTML tags to assign brief and descriptive names to items and properties. Here's an example of a short HTML block showing basic contact information for Gianluca Russo as listed in the RDFa example:

Listing 2.6: Simple HTML

```
<div>
My name is Gianluca Russo but people call me Lucas. Here is my home page: <a
   href="http://www.example.com">www.example.com</a>.
I live in Italy, and I am an engineer student at Federico II University.
</div>
```

Here is the same HTML marked up with Microdata:

Listing 2.7: HTML marked up with Microdata

```
<div itemscope itemtype="http://schema.org/Person">
My name is <span itemprop="name">Gianluca</span> <span itemprop="family␣name">
   Russo</span> but people call me <span itemprop="nickname">Lucas</span>.
Here is my home page: <a href="http://www.example.com" itemprop="url">www.
   example.com</a>.
I live in Italy, and I am an <span itemprop="jobtitle">engineer student </span>
   at <span itemprop=affiliation itemscope itemtype= "http://schema.org/
   Organization><span␣itempop=name>␣Federico␣II␣University␣</span>.</span>
</div>
```

This Microdata proposal, has gained substantial attention since the announcement of collaboration by Google, Microsoft and Yahoo! on Schema.org, a collection of vocabularies for making up the content on Web pages.

2.2.4 Schema.org

Once defined the languages that allow adding semantics to the content of Web pages, we must explain how they define the entities and the relationships between them. To support this problem ontologies are defined, as already introduced in the first chapter.

An ontology provides a shared vocabulary, which can be used to model a domain, that is, the type of objects and/or concepts that exist, and their properties and relations[17]. The current vocabularies standardized by the W3C[12], and defined as Good Ontologies, are listed below:

- The Dublin Core (DC) ontology[13] : a light weight vocabulary for describing generic metadata
- The Friend Of A Friend (FOAF) ontology[14]: used to describe people and social relationship on the Web
- Socially Interconnected Online Communities (SIOC) ontology[15]: used to describe online communities such as forums, blogs, mailing lists, wikis
- Good Relations[16]: used to describe products sold online
- The Music Ontology[17]: used to describe information related to the music industry

It can be noted that, even covering important domains of interest, such vocabularies cannot express a generalization of concepts and different domains. Obviously, being vocabularies tailored to express the concepts related to a specific and well-defined domain of interest, they lack the flexibility and interconnection between different semantic contexts.

[12]http://www.w3.org/wiki/Good_Ontologies
[13]http://dublincore.org/specifications/
[14]http://xmlns.com/foaf/spec/
[15]http://rdfs.org/sioc/spec/
[16]http://www.heppnetz.de/ontologies/goodrelations/v1
[17]http://musicontology.com/

There is therefore, the need for a unified vocabulary that expresses somehow concepts and relationships on a larger scale, which is widely chosen and shared by the community. This would facilitate the understanding operation of the semantics, because it refers to a vast environment of well established knowledge.

In this direction, Schema.org[18] is an initiative launched on 2 June 2011 by Bing, Google and Yahoo! to "*create and support a common set of schemas for structured data markup on web pages*". Schema.org is a shared markup vocabulary that makes it easier for webmasters to decide on a markup schema and get the maximum benefit for their efforts.

The introduction of Schema.org, with its support from the major search engines, accomplishes two important things. It normalizes the structured markup supported by the search engines, and it extends the topical domains of presently supported Microformats and structured vocabularies.

Schema.org supports hundreds of tags for entity description, some example pf entities are listed below. Each entity contains several properties:

Creative_Works Article Comment Movie Photograph Sculpture Blog Diet Music Recipe Software Book Map Painting Review TV

Event Business Dance Food Sale Theatre Children Education Literary Social Interaction Comedy Festival Music Sports Visual Arts

Intangible Audience Job Post Offer Rating Enumeration Language Quantity Structured Data

Medical Anatomy Device Intangible Symptom Therapy Causes Guideline Procedure Study Condition Indication Risk Test

Organization Corporation Government NGO Team Educational Local Group

Person Name Affiliation Colleagues Email Location Address Birthday Contact Family Title

Place Area Landmark Residence Civic Business Tourist Attraction Product Brand Manufacturer Model Offers Reviews

A feature behind the creation of Schema.org is to facilitate the semantic enrichment of web pages in a direct way, aligned with the content. In fact, this project started as a language support for Microdata, then extended for use with RDFa. Furthermore, it can be used on web pages

[18]http://schema.org

written in any language. This way, who uses Schema.org just has to learn one thing rather than having to understand different, often overlapping, vocabularies.

The most part of the vocabulary on Schema.org was inspired by earlier formats such as Microformats, FOAF, GoodRelations and OpenCyc[19]. In addition to these vocabularies listed above, an original approach to Schema.org was Data-vocabulary.org. Its limitations are essentially poverty of entities and properties representable through it, reason for which it was quickly put aside after the introduction of Schema.org.

New entities and properties can be proposed to the community behind Schema.org, for future adoption and implementation. In fact, recently there was the standardization of medical and biomedical domain, proposed and supported by the community[20].

2.3 Technological choices

So far the semantic markup languages for web pages has been illustrated. All three languages accomplish the same goal, to make the contents of web pages machine-understandable, but they perform with different approaches. The actual presence of three different methods that respond to the same question, inevitably raises doubts and uncertainties about which is preferable to choose.

Certainly, there is the possibility to integrate all three languages and make them interact to increase the semantic description, but this is a disadvantage in the easiness of development and in the efforts of producing it. Still, there are automated translation tools that are able to exchange web pages between different formats, with approximately acceptable results.

Consider that using a language instead of another, may depend on the context of application. For example on the basis of what use will be made of the content, what type of content we want to express, or on the simplicity of use.

A first reasoning that may lead to the decision of a language instead of another, can be which consumers will read the data within web pages, and which formats they support. These may include :

- scripting libraries

[19]http://schema.org/docs/faq.html#0
[20]http://lists.w3.org/Archives/Public/public-vocabs/2012May/0057.html

- browsers and browser extensions
- general-purpose search engines
- vertical or domain-specific search engines
- data reuses with whom you have agreements

The second consideration may be the current state of the tooling to support a particular format. For example:

- Is it possible to publish using HTML5? If the publishing method is using a content-management system CMS that doesn't support adding new attributes such as @itemprop or @typeof then, the choice to markup the document will be constrained to using Microformats.
- Are there development tools available? Because it is not visible within a web page, it can be hard to tell whether HTML data has been written correctly. Consumers should provide validators that enable to check that the data has been correctly detected and interpreted.

The limitation in the use of Microformats, although in the long used and to date the most implemented in web pages, consists in not being able to have actually structured data. In other terms, there is not a simple way to obtain an entity that is able to have more structured entities inside or overlapped.

Moreover, Microformats were never standardized and due to its design, the development of its vocabularies (hCarc, vCard, etc) was centralized and limited to one organization[18].

Considering RDFa, with version RDFa Lite 1.1 and Microdata, and carefully observing their syntax, we can easily say that they have a quite similar approach. Microdata was initially designed as a simple subset of RDFa and Microformats, primarily focusing on the core features of RDFa. Unfortunately, when this was done, the choice was made to break compatibility with RDFa and effectively fork the specification.

Conversely, RDFa Lite highlights the subset of RDFa that Microdata did, but does it in a way that does not break backwards compatibility with RDFa. This was done on purpose, so that Web developers wouldn't have a hard decision in front of them[rif].RDFa Lite contains all of the simplicity of Microdata coupled with the extensibility of and compatibility with RDFa.

The next table shows the markup attributes for Microdata and the functionally equivalent ones provided by RDFa Lite :

MICRODATA 1.0	RDFA LITE 1.1	PURPOSE
Itemid	resource	Used to identify the exact thing that is being described using a URL, s
itemprop	property	Used to identify a property of the thing being described, such as a nan
itemscope	not needed	Used to signal that a new thing is being described.
itemtype	typeof	Used to identify the type of thing being described, such as a person, e
itemref	not needed	Used to copy-paste a piece of data and associate it with multiple thing
not supported	vocab	Used to specify a default vocabulary that contains terms that are used
not supported	prefix	Used to mix different vocabularies in the same document, like ones pr

Table 2.1: Comparison between Microdata 1.0 and RDFa Lite 1.1

By comparing the data in the table, we can say that there are slightly different ways of doing almost the same thing. One difference that can be highlighted is that with RDFa Lite it is possible to mix vocabularies in the same description context. RDFa, as the schema.org FAQ acknowledges[21], is much more extensible than either microdata or microformats, "but the substantial complexity of the language has contributed to slower adoption."

Looking into standardization level, the two languages are going to be on the same stage. RDFa Lite is a W3C recommendation from 7 June 2012[22] , while HTML Microdata is currently a working draft[23] . A proposal for Microdata to transition to Candidate Recommendation is advanced although with small doubts and objections from a few members of the W3C.

Following in the thesis, HTML Microdata markup language coupled with Schema.org vocabulary are chosen for semantic tagging of web content and as a basis to develop the web application MicrodataSemantic[24] . The choice of Microdata is due to several factors listed below:

1. It is an emerging technique that facilitates the semantic expressiveness of content and the characterization of the relationships between the different entities.

2. It is basically simple to adopt and understand, no special skills or levels of understanding are required. It is minimally invasive in terms of verbosity and number of elements.

3. It is a language supported and recommended by most major search engines (Google, Bing, Yahoo!) as a semantic markup language.

[21]http://schema.org/docs/faq.html#14
[22]http://www.w3.org/TR/rdfa-lite/
[23]http://www.w3.org/TR/microdata/
[24]http://microdatasemantic.appspot.com

The last reason is decisive because it influences, even if in small proportion, the ranking of the documents so it has a value also for the SEO objectives.

The choice of Schema.org, dictated in part by the adoption of Microdata, is also due to the support that this vocabulary has. Supported by all the major search engines, makes it easier for them to understand the information on web pages and provides richer search results in order to make it easier for users to find relevant information on the web.

Chapter 3

MicrodataSemantic tool

This chapter describes the solution developed for the semantic annotation problem. Starting from the reasons that led to the development of MicrodataSemantic tool, the reader can find an overview of the implementation process. This chapter describes, by using the models of software engineering, the various steps in the development of the tool. Requirements, Design and Development show, even with the aid of diagrams, the different choices and decisions for the development of the software.

3.1 Motivation and Overview

There are different motivations behind the development of the MicrodataSemantic tool. A more abstract goal, is to achieve an improvement of search results related to the content published on the Web. To accomplish this goal, it is necessary to semantically enrich web contents.

The operation of semantic annotation, as seen in the second chapter, can be done either manually, or automatically. Hence, the second and more specific goal, is to create a tool able to semi-automatically add semantic to the content of the document to be published.

In literature there are different solutions for this type of problem, some of which perform this task using proprietary systems and not standardized languages [ref]. Others do not provide an integrated environment for evaluation/modification of semantic content [ref] [ref] [ref]. Other solutions instead, propose an external type annotation for the content[1] . This last aspect, as

[1]https://code.google.com/p/ehost/

highlighted in the second chapter, does not allow search engines to extract the semantic content of the document, thus making it impossible to better understand the content in terms of entity and relationship.

Hence, the need to create an integrated tool to support different aspects of the creation phase of semantic annotations. Firstly, this tool must be able to validate the content of a web document, that is, the verification of well-structured and well-composed semantic annotations. Second, it must allow the creation of semantic content in semi-automatic mode. In this phase, the creation of semantics should be set following default template. In this way, the content can be more structured.

Finally, the tool should allow the final verification of semantic content created, clearly identifying the entities and the relationships between them. The creation of such a tool, as well as answering the question of creating an "assisted" system for semantic annotation, allows the user that creates/publishes online content, avoiding the laborious task of adding manual verification of semantics.

3.2 Requirements

The reasons outlined in the previous section led to formulate the requirements for the semantic annotation system MicrodataSemantic.

These requirements, listed below, are grouped into functional and non-functional requirements, described respectively in terms of user and system requirements.

3.2.1 Functional requirements:

Below are described the user functional requirements that must be supplied by the system.

1. Creation and semantic enrichment of documents: the user must be able to create content through the system that have well-defined semantics. The semantic enrichment must be automatic, that is, the user must not perform any manual operation of adding tags. The creation of semantic content must follow a predefined template.

2. Semantic verification for web pages: the user must be able to verify the presence of content and semantic elements within existing web pages accessible via url. It must also provide the relationships between the different entities recognized.

3. Search engines interpretations: in addition to verifying the presence of semantic information, the system must show the user how this content is interpreted by search engines when a search for this content is performed.

4. Semantic verification for local documents: the user can verify from a local web document, that is not created by the tool, the presence of semantic content in terms of entities and relationships between them as point 2 .

5. Validation of created content: the user, once created the content, can perform a validation of semantic elements described, before using the content in a web page. The user can check whether the extracted entities are valid and match them with what was entered in the content.

3.2.2 Non-functional requirements:

Applied to the whole system and not to individual features they include:

1. Language independent: The system should provide a tagging and adding semantics procedure that are independent from the language used . Hence, the semantic enrichment of content is independent from the language used, making the tool suitable across multiple languages.

2. Data representation: by the considerations expressed in the second chapter, the system must add, internally to the source code created, the tags that allow semantic enrichment. These tags must underlie the Microdata Specifications.

3. Vocabulary used: to describe the entities, relationships between them and the different properties associated, the system must make explicit use of a single vocabulary. Any reference to entities and properties must then be connected to the schemes presented by this vocabulary. This vocabulary is Schema.org

3.2.3 Use Case Diagram

From requirements analysis, UML Use Case Diagram notation is used for defining the user requirements and the interaction between the user and the system, highlighting the capabilities of the system as perceived by the user.

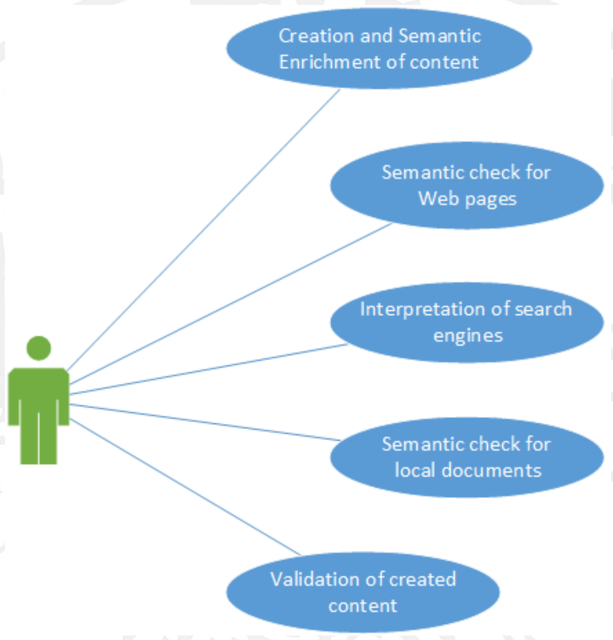

Creation and Semantic Enrichment of content
Semantic check for Web pages
Interpretation of search engines
Semantic check for local documents
Validation of created content

Figure 3.1: Use Case diagram

3.3 Architectural Design

The architecture used for the development of MicrodataSemantic tool is based on the client/server model. More specifically, a RIA web application (Rich Internet Applications) that support the process of semantic annotation is developed. The choice of this type of software architecture is influenced by some intrinsic features that this tool should have, which may be expressed as non-functional requirements as follows:

1. Performance: The tool should be able to respond in an efficient and effective way in terms of performance, to user requests.
2. Availability: The system must be distributed and accessed directly, without requiring the installation of software or plug-in by the user.

3. Maintainability: The software should be easily upgradeable and expandable, given the flexibility of the context.

The client/server approach, and more specifically with the adoption of RIA web application, responds directly to the performance and availability requirements.

First, being a web application, it is performed in a safe environment, the web browser. Therefore, it does not require the installation of the software by the user. Second, the basic characteristic of RIA web application is its strong interactivity, it can be compared to traditional desktop applications.

Relying on the client/server paradigm, the system software also becomes easily maintainable from a functional viewpoint. In fact, since only one version of the software runs and resides on servers, updating and distribution are minor problems. Another benefit is that the engine which processes the data, resides on the server level and provides a ready answer to the user interface, with considerable lightening for the computer user.

From the specification of the requirements, the integrated system for the semantic enrichment can be decomposed into two subsystems which provide two types of services. The first module is responsible for the process of verification and validation of semantic contents. The second module is responsible instead, of the content creation process by adding semantic tags.

3.4 Deployment

Having defined the general architecture of the MicrodataSemantic tool, the next step is the development of the application. Even the choices made at this stage and the technological solutions adopted are the result of the architectural requirements defined above. The first technology solution chosen to better respond to the architectural requirements, is the Vaadin framework.

Vaadin is an open source web application framework for rich Internet applications completely written in Java. In contrast to Javascript libraries and browser-plugin based solutions, it features a server-side architecture, which means that the majority of the logic runs on the servers. Ajax technology is used at the browser-side to ensure a rich and interactive user experience. To meet the requirements of maintainability and availability Google App Engine is chosen as a distribution platform of the web application.

Both technological choices and the design patterns which were used as a model of development, will be introduced in the following paragraphs. Furthermore, from the technical solution chosen and described in Chapter 2, Microdata and Schema.org are implemented as markup language and vocabulary for semantic description of entities and relationships between them.

3.4.1 Model View Controller Design pattern

Developing a software without following the well-established software engineering standards, leads to systems with low or poor maintainability and low performance. For the development of this tool, it was decided to adopt the Model-View-Controller design pattern. This choice allows the decoupling between the visualization components, control and data model.

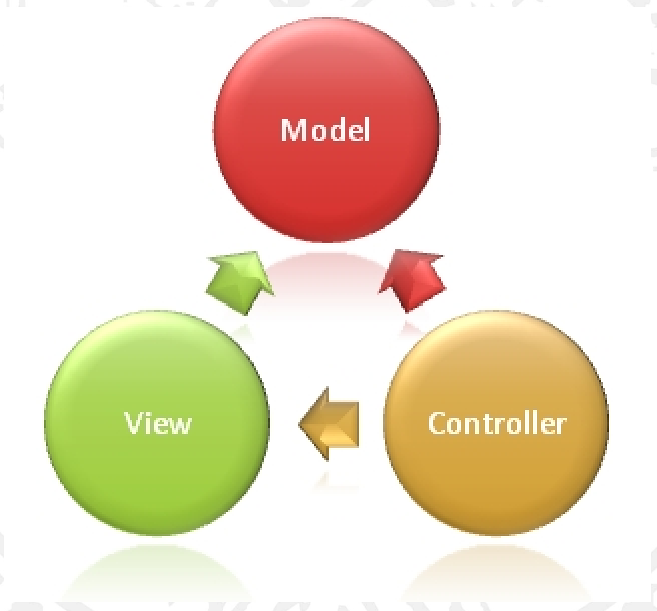


Figure 3.2: Model View Controller design pattern

The pattern is based on the separation of duties between the software components that interpret three main roles:

- The Model, provides the necessary methods to access the useful application data and implements the business logic
- The View, uses and visualizes data from the Model and deals with the presentation logic

- The controller, manages the interaction of users with the View, receiving user commands by changing the status of the other two components.

This layer implements the control logic of the application. The choice of this approach is mainly linked to the context of this tool. Since the data model is decoupled from View and Controller, it is possible, for future needs, to implement even a different markup language, different from the approach adopted in this phase. For example it is possible to change the markup language from Microdata to RDFa Lite, without implying changes in the View or Controller layer.

Ultimately the MVC approach chosen as a development model, allows this tool to respond effectively and efficiently to the requirement of maintainability introduced in the previous paragraph.

3.4.2 Top-Down Approach

There are many approaches available for semantic content authoring which address different aspects of this task by proposing appropriate user interface. Regarding explicit semantic content authoring, recent approaches can be roughly classified into the categories Top-Down and Bottom-Up. The classification is based on the starting point of the authoring process, which can be on ontologies/vocabularies (with upper level of expressiveness) or unstructured content (with lower level of expressiveness)

1. Bottom-Up Approaches: these approaches which are usually called semantic annotation techniques aim to annotate existing documents using a set of predefined ontologies/vocabulary. The basic ingredients of a semantic annotation system are ontologies/vocabulary, the documents and the annotations that link ontologies to documents. The result of the annotation process is a document that is marked-up semantically.
2. Top-Down Approaches: these approaches which are also called Ontology Population[19] techniques aim to create semantic content based on a set of initial ontologies/vocabulary which are extended during the population process. When compared with the bottom-up approaches, these approaches deal with semantic representations from the beginning instead of lifting unstructured content to a semantic level.

In the MicrodataSemantic tool Top-Down approach is used. To achieve this goal, semantic templates technique is used. In this approach each class of the ontology/vocabulary (in this case Schema.org) has an associated template. This means that when the user creates a new content, the tool links entity and property following the class instance provided from Schema.org vocabulary.

3.4.3 Vaadin Framework

Vaadin is a web application framework for Rich Internet Applications (RIA). In contrast to Javascript libraries and browser-plugin based solutions, Vaadin features a complete stack that include a robust server-side programming model as well as client-side development tools based on GWT and HTML5. The rapid development model abstracts away from implementation details such as RPC and cross browser compatibility and has full control over all the layers.

3.4.3.1 Architectural overview

Vaadin Framework is a Java web application development framework that is designed to make creation and maintenance of high quality web-based user interfaces easy. Vaadin supports two different programming models: server-side and client-side. The server-driven programming model is the more powerful one, and essentially lets the developers forget the web and program user interfaces much like the developers would program any Java desktop application with conventional toolkits such as AWT, Swing, or SWT.

The server-side Vaadin framework takes care of managing the user interface in the browser and the AJAX communications between the browser and the server.

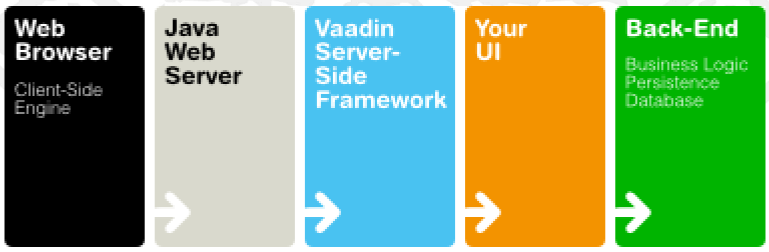


Figure 3.3: Vaadin Framework interaction

Fig.3.3 illustrates the basic architecture of server-side web applications made with Vaadin. This architecture consists of the server-side framework and a client-side engine that runs in the browser, rendering the user interface and delivering user interaction to the server.

The user interface of the application runs as a Java Servlet session in a Java application server, and the client-side engine as JavaScript. As the client-side engine is executed as JavaScript in the browser, no browser plugins are needed for using applications made with Vaadin. This gives it an edge over frameworks based on Flash, Java Applets, or other plugins.

Vaadin relies on the support of Google Web Toolkit for a wide range of browsers, so that the developer does not need to worry about browser support. Because HTML, JavaScript, and other browser technologies are essentially invisible to the application logic, developers can think of the web browser as only a thin client platform.

A thin client displays the user interface and communicates user events to the server at a low level. The control logic of the user interface runs on a Java-based web server, together with your business logic. By contrast, a normal client-server architecture with a dedicated client application would include a lot of application specific communications between the client and the server.

Essentially removing the user interface tier from the application architecture makes Vaadin's approach a very effective one. Behind the server-driven development model, Vaadin makes the best use of AJAX (Asynchronous JavaScript and XML) techniques that make it possible to create Rich Internet Applications (RIA) that are as responsive and interactive as desktop applications. A server-side Vaadin application runs as a servlet in a Java web server, serving HTTP requests.

The VaadinServlet is normally used as the servlet class. The servlet receives client requests and interprets them as events for a particular user session. Events are associated with user interface components and delivered to the event listeners defined in the application. If the UI logic makes changes to the server-side user interface components, the servlet renders them in the web browser by generating a response. The client-side engine running in the browser receives the responses and uses them to make any necessary changes to the page in the browser.

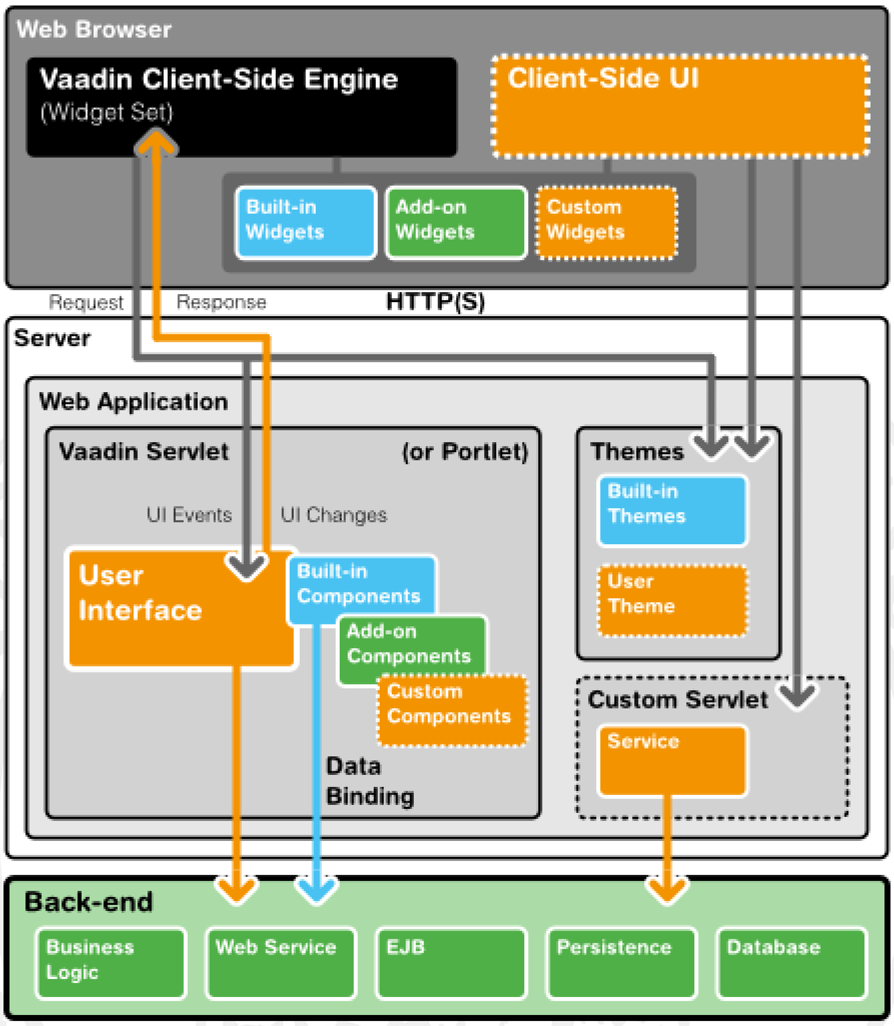


Figure 3.4: Vaadin Framework architecture

Fig.3.4 gives a basic illustration of the client-side and server-side communications, in a running situation where the page with the client-side code (engine or application) has been initially loaded in the browser. In addition to the server-side Java application development, it is possible to develop on the client-side by making new widgets in Java, and even pure client-side applications that run solely in the browser.

The Vaadin client-side framework includes Google Web Toolkit (GWT), which provides a compiler from Java to the JavaScript that runs in the browser, as well a full-featured user interface framework. With this approach, Vaadin is pure Java on both sides. Vaadin uses a client-side engine for rendering the user interface of a server-side application in the browser.

Vaadin Framework defines a clear separation between the structure of the user interface and its appearance and allows to develop them separately. Vaadin's approach to this is themes, which control the appearance by CSS and (optional) HTML page templates.

3.4.4 Google AppEngine

Google App Engine is a platform and SDK for developing and hosting web application using Google's servers and infrastructure. App Engine applications are easy to build, easy to maintain, and easy to scale as the application's traffic and data storage needs grow. With App Engine, there are no servers to maintain: Developers just upload their application, and it's ready to serve the users.

3.4.4.1 Brief Overview

Google app Engine as a platform is designed for scalability, robustness and performance. App Engine offers a reliable performance even under a heavy load and when using very large amounts of data. A request to an App Engine app is routed to a selected app server, and the application is started on the server if necessary.

No state on the server is saved between request. There is no guarantee that the same server will handle two subsequent requests, even if the time period between them is very short. Thus, a runtime instance often comes into existence when a request handler begins, and is terminated when it ends- though the instance may sometimes be retained, depending upon app traffic. Applications run in a secure environment that offers limited access to the underlying operating system.

These limitations allow App Engine to distribute web requests for the application on multiple servers and to start and stop servers to meet the needs of traffic. The sandbox isolates the application in a predetermined environment, ensuring security, reliability and independence from hardware system and the physical location of the web servers themselves. App engine is structured differently from the typical web application server.

At its core App Engine restricts the application uploaded on its infrastructure from any access to the physical layer, preventing from opening sockets, running background process and using other common back-end routines that application developers take for granted in other environment. Google App Engine environment is free for use, with some limitation, based on the traffic and the storage used by the uploaded application.

3.4.5 Detailed Solutions

Below some detailed solutions that provide a complete view of the implemented tool are presented . The development of the semantic content authoring, MicrodataSemantic, is composed of two phases. The first phase is the implementation and development through the use of Vaadin Framework.

The second phase is the deployment of the software on Google AppEngine. Before uploading, it is necessary to setup the environment in which such software is loaded. It is possible to break down the development of the tool in two separate modules. The first is responsible for adding semantic annotations to the content, the second is responsible for verification and validation of semantic attributes present on web pages and on created content.

3.4.5.1 Semantic content creation

The first module will then enrich the semantic content created, by adding certain tags that make such content machine-understandable. Top-Down approach is used to create semantic content. This means that the system shows a series of templates that match with Schema.org classes. In particular, different and distinct classes are identified, to facilitate the semantic annotation process. However, it is possible to interconnect these categories to create more complex and structured content.

Templates are developed to allow better explaining of entities and relationships between different properties. Examples of this template implemented in the tool, contains classes describing the articles Category, People, Organizations, Books, Events and Application Software. Obviously, since the vocabulary Schema.org includes the definition of hundreds of categories, for the objectives of the thesis, a few examples listed above are implemented.

Each class of Schema.org vocabulary is a template implemented in the tool. The user interface shows these classes, whose properties are shown in the form of fields, associated to the template.

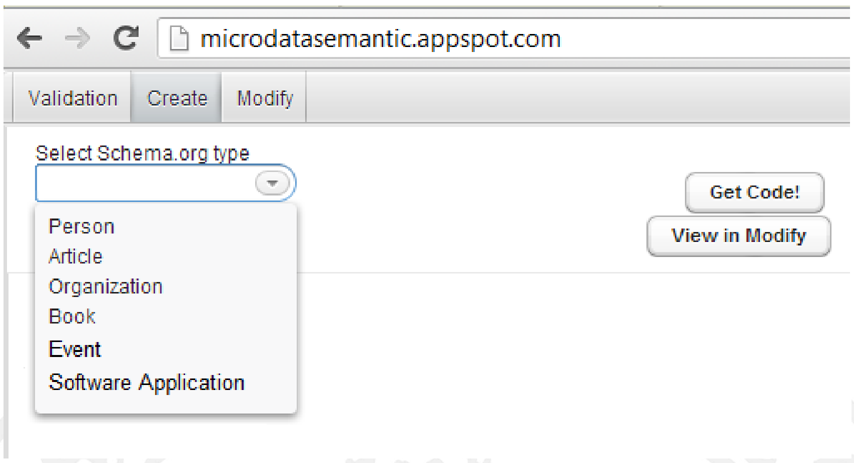
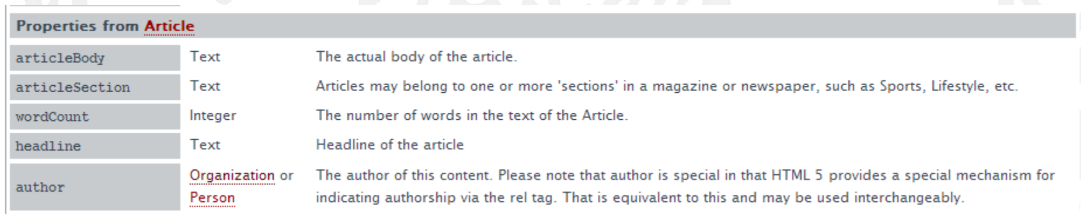


Figure 3.5: Available templates

When the user selects one template, all the properties connected to the associated classes from Schema.org vocabulary are shown in the user interface. Considering the Article classes from Schema.org[2], the main entity is the Article item. Many properties are related with this item, like:



Properties from Article		
articleBody	Text	The actual body of the article.
articleSection	Text	Articles may belong to one or more 'sections' in a magazine or newspaper, such as Sports, Lifestyle, etc.
wordCount	Integer	The number of words in the text of the Article.
headline	Text	Headline of the article
author	Organization or Person	The author of this content. Please note that author is special in that HTML 5 provides a special mechanism for indicating authorship via the rel tag. That is equivalent to this and may be used interchangeably.

Figure 3.6: Schema.org Article entity

These properties can be simple or complex, and their meaning is explained by the vocabulary as the figure above. Simple properties as “articleBody” are composed by simple text. Complex properties as “author” are linked to new classes which have a new set of properties associated to them. This kind of complex entities make the content more structured, allowing more formalized specification for entity and classes.

The process of semantic annotation consists of 3 steps:

1. The user selects the template for which he is creating the semantic content

[2]http://schema.org/Article

2. The user fills the fields of the template, that are essentially the properties directly connected to the Schema.org classes

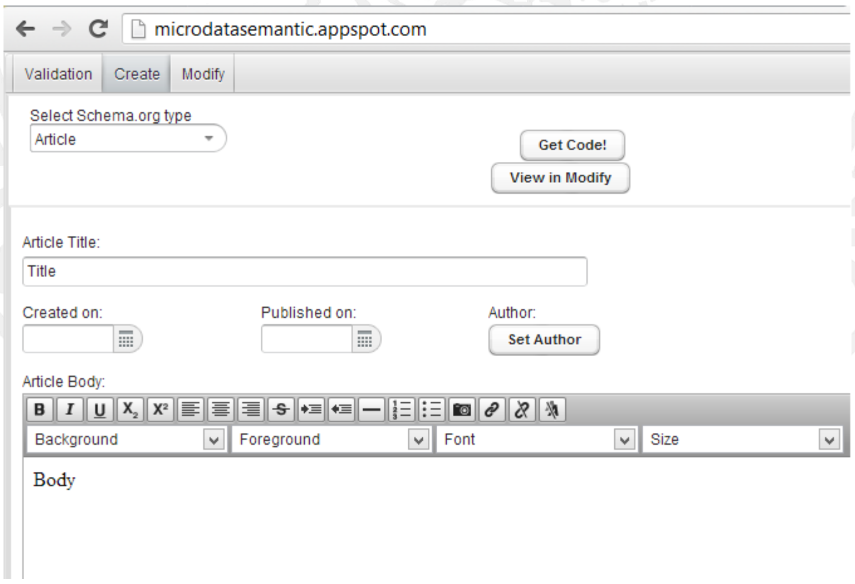
3. The system generates the HTML source code by adding the HTML Microdata attributes corresponding to the Schema.org Classes.

Referring to the Article Template, a simple example of an article composed by a title and body in HTML source code can be:

```
<H1> Title </H1>
<div>Text body</div>
```

The HTML source code generated by the tool, filling the Article Title and Article Body fields as shown below, is instead:

```
<div itemscope itemtype="http://schema.org/Article">
       <div itemprop="headline">Title</div>
       <div itemprop="articleBody">Text Body</div>
</div>
```


microdatasemantic.appspot.com

Validation | Create | Modify

Select Schema.org type: Article

Get Code!
View in Modify

Article Title: Title

Created on:
Published on:
Author: Set Author

Article Body:
Background | Foreground | Font | Size

Body

Figure 3.7: Article entity creation

The entities (the classes from Schema.org) are specified by using the HTML Microdata attributes itemscope itemtype with the associated vocabulary extension. The properties associated to the entities are specified by using the itemprop attributes. The MicrodataSemantic tool add these Microdata attributes inside div or span HTML tags, while the semantics is expressed by using the extension of the classes ("http://schema.org/Article" for example) and the properties associated .

This approach combines ontological rigour with flexible user interface to create semantic content. Data properties are displayed as a simple field to fill, while object properties are displayed as a button that links to other templates (representing other instance of the ontology).

The schemas on which to define the basis to add semantic tags are contained in the package com.microdatasemantic.schematype, that defines the Model of the application, are the following:

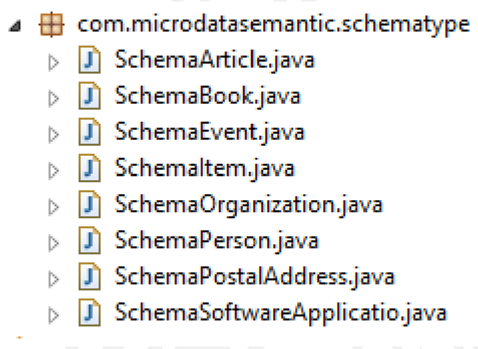

com.microdatasemantic.schematype
SchemaArticle.java
SchemaBook.java
SchemaEvent.java
SchemaItem.java
SchemaOrganization.java
SchemaPerson.java
SchemaPostalAddress.java
SchemaSoftwareApplicatio.java

Figure 3.8: Package Schematype

These classes provide the methods to add semantic tags to the content. Because they are independent from the user interface and the controller, it is possible to radically change the data model, using for example the RDFa Lite language, or a different vocabulary. An example of these classes is provided by the following figure which shows the class SchemArticle.java

```
public class SchemaArticle {
  static String prop;

  public final static String construct_ItemScope(){
    prop="<div␣itemscope␣itemtype=\"http://schema.org/Article\">";
    return prop;
  }
  public final static String construct_ItemPropHeadLine(String name){
    prop= "<div␣itemprop=\"headline\">"+name+"</div>\n";
    return prop;
  }
```

```
public final static String construct_ItemPropDatePublished(String name){
  prop= "<div␣itemprop=\"datePublished\">"+name+"</div>\n";
  return prop;
}
public final static String construct_ItemPropDateCreated(String name){
  prop= "<meta␣itemprop=\"dateCreated\"␣content="+name+">\n";
  return prop;
}
public final static String construct_ItemPropArticleBody(String name){
  prop= "<div␣itemprop=\"articleBody\">"+name+"</div>\n";
  return prop;
}
```

Following the Model View Controller design pattern, each template has a view, a controller and a data model associated. This allows, for example, to reuse these components to enable more complex functionality. In other terms, it is possible to create more structured and nested entity without having to redefine the associated user interface or controller. The components related to the user interface, contained in the package com.microdatasemantic.componentUI, representing the View of the MVC pattern, are as follows:

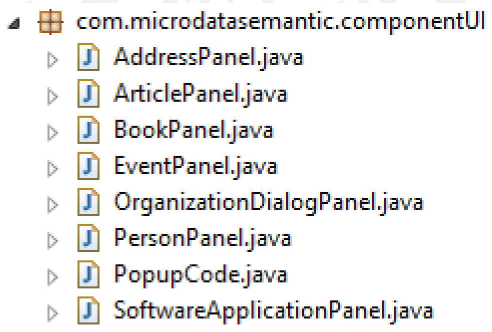
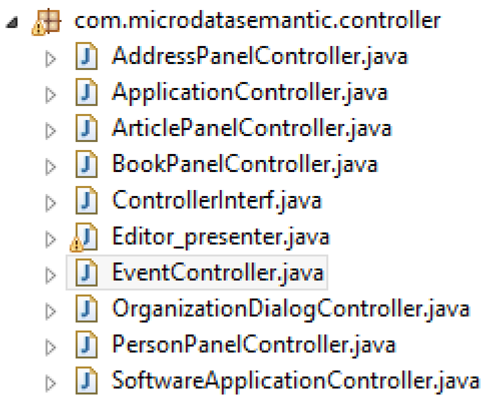

com.microdatasemantic.componentUI
AddressPanel.java
ArticlePanel.java
BookPanel.java
EventPanel.java
OrganizationDialogPanel.java
PersonPanel.java
PopupCode.java
SoftwareApplicationPanel.java

Figure 3.9: Package Component User Interface

The components used in the User Interface, are those offered by Vaadin framework, while the controller of each view, intercepts the user's interactions and changes its state. An extract of the controller associated with the user interface for article news template is shown below.

```
public class ArticlePanelController implements Serializable, ControllerInterf {
...
...
  @Override
  public void saveData(){
    articleData = new String();
    String title = articlPanel.getTextField_Headline().getValue();
    DateFormat dateFormatter = DateFormat.getDateInstance(DateFormat.SHORT);
    Date dateCreat = articlPanel.getPopupDateField_Date().getValue();
    Date datePubl = articlPanel.getPopupDateField_Published().getValue();
    String body = articlPanel.getRichTextArea_ArticleBody().getValue();

    if(!title.isEmpty()||!body.isEmpty()){
      aritcleData=SchemaArticle.construct_ItemScope();
      if(!title.isEmpty())
        aritcleData+=SchemaArticle.construct_ItemPropHeadLine(title);
      if(datePubl!=null){
        String data =dateFormatter.format(datePubl);
        aritcleData+=SchemaArticle.construct_ItemPropDatePublished(data);
      }
      if(dateCreat!=null){
        String data =dateFormatter.format(dateCreat);
        aritcleData+=SchemaArticle.construct_ItemPropDateCreated(data);
      }
      if(!body.isEmpty())
        aritcleData+=SchemaArticle.construct_ItemPropArticleBody(body);
      if(mainController.getPersonController() !=null){
        if(mainController.getPersonController().getPersonalData()!= null
          && !mainController.getPersonController().getPersonalData().isEmpty())
          aritcleData+=mainController.getPersonController().getPersonalData();
        }
    }
  }
  @Override
  public void init() {
    articlPanel.getRichTextArea_ArticleBody().setDescription("Copy␣and␣Paste␣you␣article␣here!");
  }
  @Override
  public void bind() {
    articlPanel.getButton_Author().addClickListener(new SetAuthor());
  }
}
```

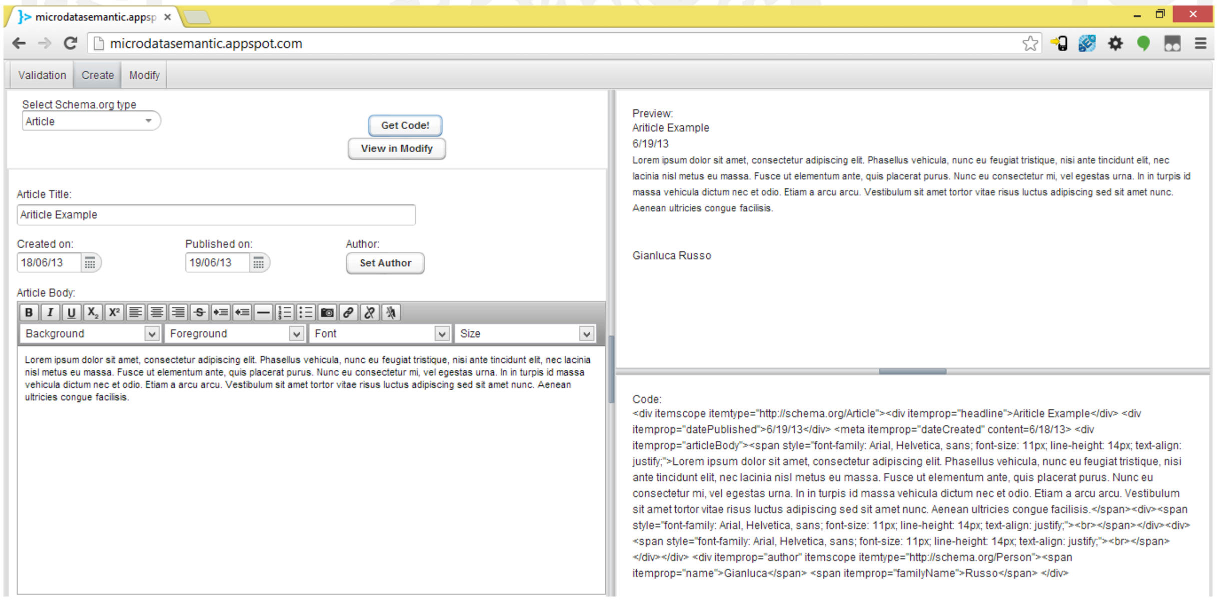
Each controller class, implements ControllerInterface interface that defines init() bind() and SaveData() methods. In addition, Serializable interface is implemented. This is a fundamental requirement for the correct operation of the application on the Google AppEngine platform (allows the serialization of data between client and server). Components that belong to the control logic of the application are included in the package com.microdatasemantic.controller with reference to the following figure:



```
com.microdatasemantic.controller
    AddressPanelController.java
    ApplicationController.java
    ArticlePanelController.java
    BookPanelController.java
    ControllerInterf.java
    Editor_presenter.java
    EventController.java
    OrganizationDialogController.java
    PersonPanelController.java
    SoftwareApplicationController.java
```

Figure 3.10: Package Controller

During the step of content creating, the controller of each template, change the user interface displaying the content either in the form of source code, or in the form of formatted html code as shown in the following figure:



Select Schema.org type: Article

Article Title: Ariticle Example

Created on: 18/06/13
Published on: 19/06/13
Author:

Article Body:

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus vehicula, nunc eu feugiat tristique, nisi ante tincidunt elit, nec lacinia nisl metus eu massa. Fusce ut elementum ante, quis placerat purus. Nunc eu consectetur mi, vel egestas urna. In in turpis id massa vehicula dictum nec et odio. Etiam a arcu arcu. Vestibulum sit amet tortor vitae risus luctus adipiscing sed sit amet nunc. Aenean ultricies congue facilisis.

Preview:
Ariticle Example
6/19/13
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus vehicula, nunc eu feugiat tristique, nisi ante tincidunt elit, nec lacinia nisl metus eu massa. Fusce ut elementum ante, quis placerat purus. Nunc eu consectetur mi, vel egestas urna. In in turpis id massa vehicula dictum nec et odio. Etiam a arcu arcu. Vestibulum sit amet tortor vitae risus luctus adipiscing sed sit amet nunc. Aenean ultricies congue facilisis.

Gianluca Russo

Code:
```
<div itemscope itemtype="http://schema.org/Article"><div itemprop="headline">Ariticle Example</div> <div
itemprop="datePublished">6/19/13</div> <meta itemprop="dateCreated" content=6/18/13> <div
itemprop="articleBody"><span style="font-family: Arial, Helvetica, sans; font-size: 11px; line-height: 14px; text-align:
justify;">Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus vehicula, nunc eu feugiat tristique, nisi
ante tincidunt elit, nec lacinia nisl metus eu massa. Fusce ut elementum ante, quis placerat purus. Nunc eu
consectetur mi, vel egestas urna. In in turpis id massa vehicula dictum nec et odio. Etiam a arcu arcu. Vestibulum
sit amet tortor vitae risus luctus adipiscing sed sit amet nunc. Aenean ultricies congue facilisis.</span><div><span
style="font-family: Arial, Helvetica, sans; font-size: 11px; line-height: 14px; text-align: justify;"><br></span></div><div>
<span style="font-family: Arial, Helvetica, sans; font-size: 11px; line-height: 14px; text-align: justify;"><br></span>
</div></div> <div itemprop="author" itemscope itemtype="http://schema.org/Person"><span
itemprop="name">Gianluca</span> <span itemprop="familyName">Russo</span> </div>
```

Figure 3.11: Preview and Source code during the creation

By placing the contents in the fields provided by the user interface, the tool automatically annotated them according to the template used. Entities, properties and relationships between them are inserted into the HTML code through the use of div and span attributes.

3.4.5.2 Semantic validation and verification

The second module is used for the verification and validation process of semantic annotations. It is possible to further decompose this module into two subsystems. The first carrying out the verification process for web pages accessible via a URL. The second performing the validation of content created by the application itself .Both subsystems have the purpose of verifying the presence of semantic content. In other terms, they perform a search within the document to extract entities and a property from the content, that are expressed in machine-understandable form.

The first subsystem then, receives as input the URL address of the web page to verify. In addition, as requirements, the tool shall analyze how that web page is interpreted by search engines when displayed in search results. An online service that provides this particular type of information is offered by Google RichSnippet tool.

The use of this service thus, permits to extract and display all the information that are needed to meet the user requirements. More specifically, provided an URL to the UI, the controller listens for the event and triggers a series of actions. The controller for this event is provided in the package com.microdatasemantic.controller, Editor_presenter.java class.

After checking the validity of the URL provided by the user, through the library Jsoup, the web page is submitted to Rich Snippet tool service. The response of this service provides all the semantic entities (if any) within the web page.

A field that provides clear information on how these data are interpreted by the search engine (Google), is provided by the Search Preview field. It shows, in case of specific structured data inside the web page, additional descriptions, the so-called "Rich Snippets". The controller, once processed the page, changes the user interface, updating it with the extracted data. The operations performed by the Editor_presenter.java controller, that allows the information extraction from RichSnippetTool, are shown in the following figure.

First of all, once intercepted the web page submission, its reachability is checked first. Then, through the Jsoup library, the tool creates a Document type object, which contains the web page resulting from RichSnippet tool.

From this object (richSnipp), the fields that contain the useful information as Structued Data and Preview, are extracted and displayed in the user interface. This last operation is performed in the "fillRichSnippet" method.

```
private class GetWebsite implements ClickListener, Serializable {

    private static final long serialVersionUID = 1L;
    @Override
    public void buttonClick(ClickEvent event) {
      if (bindUrl()) {
        if (Utility.checkUrl(url)) {
          getRichSnippet();
        }
      }
    }
....
public boolean bindUrl() {
    if (!editor.getTextField_addessBar().getValue().isEmpty()) {
      if(editor.getTextField_addessBar().getValue().contains("http://"))
        url = editor.getTextField_addessBar().getValue();
      else
        url = "http://"+editor.getTextField_addessBar().getValue();
      return true;
    } else
      return false;
}
public void getRichSnippet() {
    try {
      richSnipp = Jsoup.connect(richSnipUrl1 + url + richSnipUrl2).get();
      //cleanValidation();
      fillRichSnippet();
} catch (IOException e) {
      e.printStackTrace();
      Notification.show("Not able to load editor pres " + url, "",
      Notification.Type.ERROR_MESSAGE);
    }
}
public void fillRichSnippet() {
    cleanHtml();
    editor.getLabel_Anteprima().setValue(richSnipp.getElementById("google-preview").toString());
    editor.getLabel_Author().setValue(richSnipp.select("div#extracted-authors > div.section-content ").first
        ().toString());
    editor.getLabel_Publisher().setValue(richSnipp.getElementById("extracted-publishers").toString());
    editor.getLabel_StricturedData().setValue(richSnipp.getElementById("extracted-data-google").toString());
}
```

The second subsystem has the function to validate the content created by the tool. For this specific task, an algorithm that navigates the tree structure of the HTML created and analyzes all the semantic entities present inside has been implemented. Each semantic entity represented by Microdata, is composed by a type and a series of properties that can be simple or complex. The simple properties essentially have textual content, while complex properties contain more entity. This is the case of nested entities that make the content more structured in correspondence of these relationships.

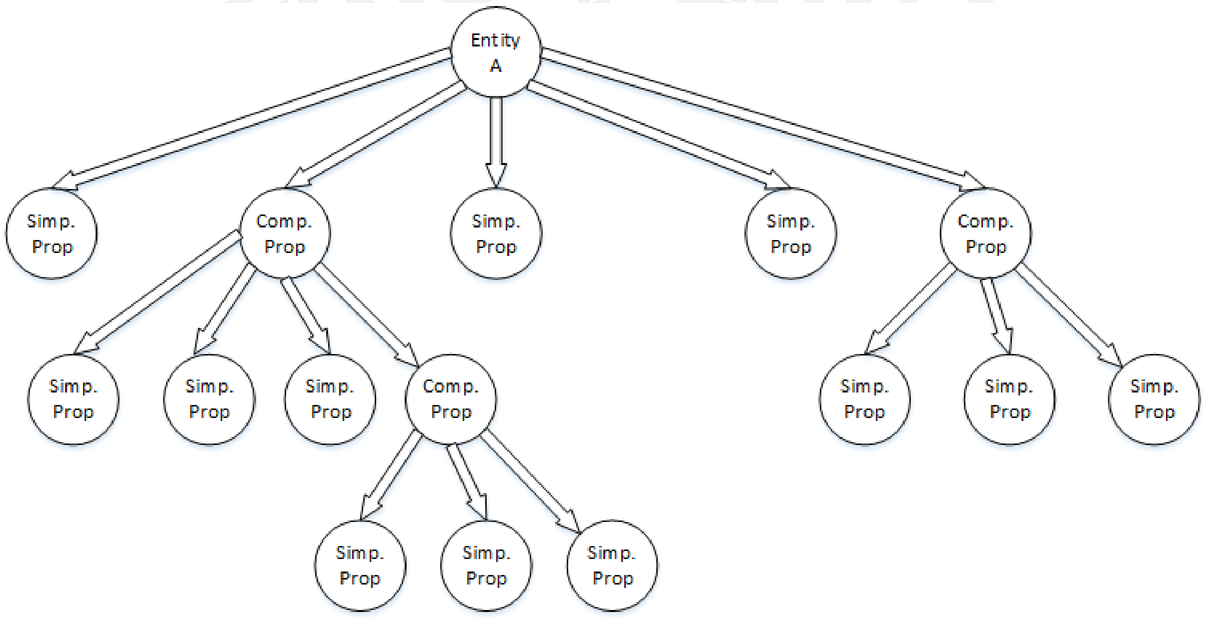


Figure 3.12: Entity structure

The SchemItem class represent the entity structure mentioned above.

```
public class SchemaItem {
  private String itemType;
  private Map<String, String> simpleProp;
  private Map<String,SchemaItem> complexProp;

  public SchemaItem() {
    simpleProp= new HashMap<String, String>();
    complexProp= new HashMap<String, SchemaItem>();
    itemType= new String();
  }
  public String getitemType() {
    return itemType;
  }
  public void setitemType(String itemType) {
    this.itemType = itemType;
  }
  public Map<String, String> getSimpleProp() {
    return simpleProp;
  }
  public void setSimpleProp(Map<String, String> simpleProp) {
    this.simpleProp = simpleProp;
  }
  public Map<String, SchemaItem> getComplexProp() {
    return complexProp;
  }
  public void setComplexProp(Map<String, SchemaItem> complexProp) {
    this.complexProp = complexProp;
  }
}
```

Each SchemaItem object contains the definition of the type and two categories of properties, simple and complex. These last two properties are data structures that store key / value pairs, making fast and efficient operations such as insertion and search of elements. The simpleProp is a collection of string/string value pair that holds the simple properties of the entity.

The complexProp is a collection of string/SchemaItem value pair, that is, the property of the entity is a new object of type SchemaItem, then a new entity. The implementation adopted is HashMap that uses the ordering of keys based on the hash codes. This type of adoption allows an easy representation of the semantic elements within the content created by the tool. The algorithm that performs the search and creation of this collection of semantic entities is inserted in the Editor_controller.java class, within FindItems method.

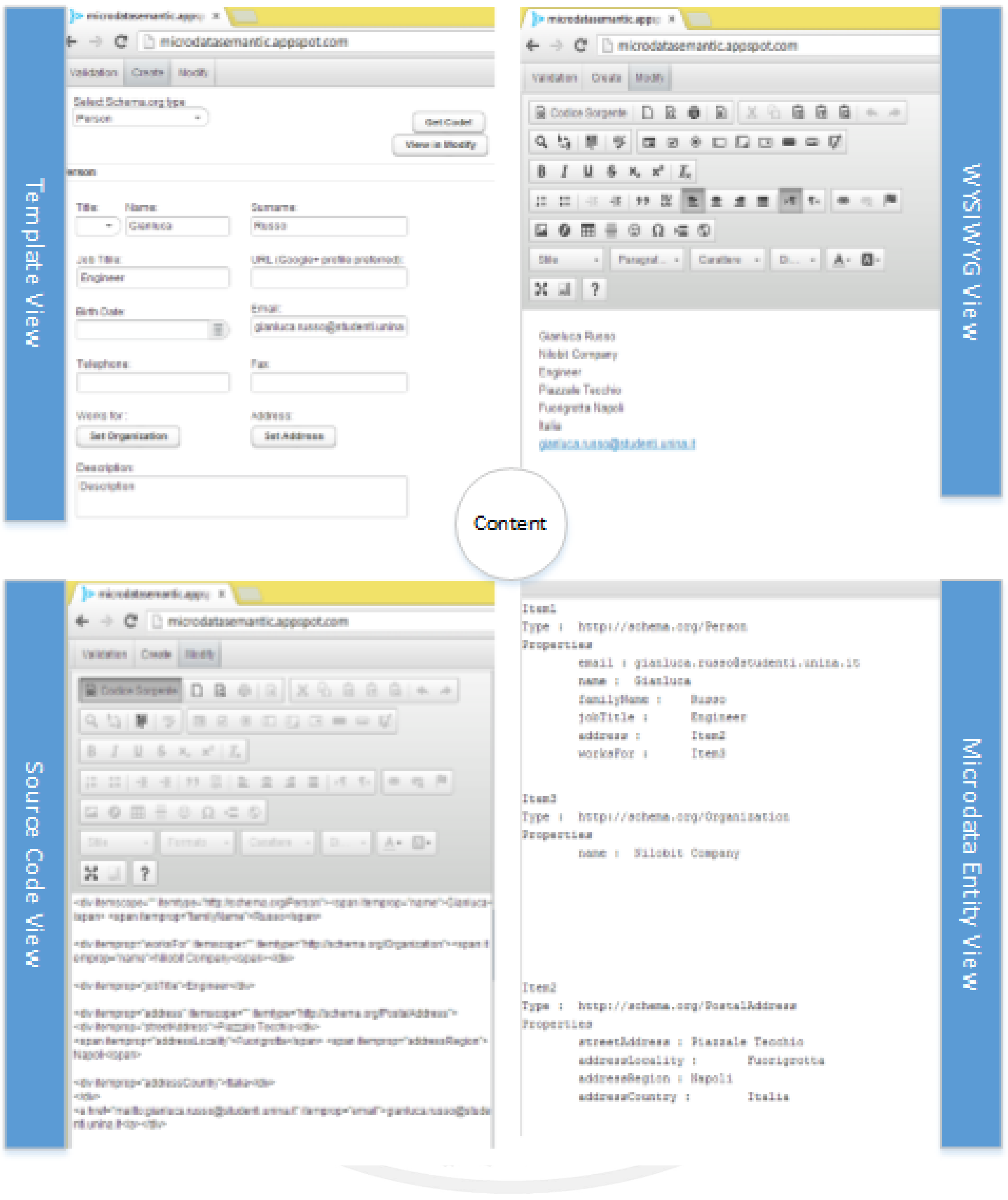
This method receives as input the HTML segment that contains the first definition of itemscope found by Jsoup library, in the HTML content. Itemscope tag, defines the first semantic tag for the formal specification of an entity, which is also associated to the itemtype tag that defines the entity types, according to the vocabulary class used. The second parameter that the method receives as input, is the SchemaItem object. FindItems method bind different properties founded in the HTML segment. In cases where a property is complex, the algorithm makes a recursive call to the FindItems method itself, that creates the new entity. FindItems method is illustrated in the following listening.

```
....
....
```

```
Document doc = Jsoup.parse(editor.getcKEditorTextField_1().getValue());
Element div = doc.select("[^itemscope]").first();
SchemaItem item= new SchemaItem();
String type=div.attr("itemtype");
item.setitemType(type);
findItems(div,item);

....
private SchemaItem findItems(Element div, SchemaItem item1) {
  for (Element element : div.children()) {
    if(!element.getElementsByAttribute("itemtype").isEmpty()){
      SchemaItem item2= new SchemaItem();
      String type=element.attr("itemtype");
      String prop=element.attr("itemprop");
      item2.setitemType(type);
      item1.getComplexProp().put(prop, findItems(element,item2));

    }else if(!element.getElementsByAttribute("itemprop").isEmpty()&& element.getElementsByAttribute("
        itemtype").isEmpty()){
      String prop=element.attr("itemprop");
      String value=element.text();
      item1.getSimpleProp().put(prop, value);
    }
}
  return item1;

}
```

3.4.5.3 User Interface for semantic text annotation

One of the main innovations carried out by MicrodataSemantic tool, is the support to different views on the semantically annotated content. MicrodataSemantic supports four different views for semantic text annotation, which are shown in the figure below and explained in a more detailed way in the sequel.



Select Schema.org type: Person

Title: | Name: Gianluca | Surname: Russo
Job Title: Engineer | URL (Google+ profile preferred):
Birth Date: | Email: gianluca.russo@studenti.unina
Telephone: | Fax:
Works for: | Address:
Description: Description

Gianluca Russo
Nilobit Company
Engineer
Piazzale Tecchio
Fuorigrotta Napoli
Italia
gianluca.russo@studenti.unina.it

```
<div itemscope="" itemtype="http://schema.org/Person"><span itemprop="name">Gianluca</span> <span itemprop="familyName">Russo</span>

<div itemprop="worksFor" itemscope="" itemtype="http://schema.org/Organization"><span itemprop="name">Nilobit Company</span></div>

<div itemprop="jobTitle">Engineer</div>

<div itemprop="address" itemscope="" itemtype="http://schema.org/PostalAddress">
<div itemprop="streetAddress">Piazzale Tecchio</div>
<span itemprop="addressLocality">Fuorigrotta</span> <span itemprop="addressRegion">Napoli</span>

<div itemprop="addressCountry">Italia</div>
</div>
<a href="mailto:gianluca.russo@studenti.unina.it" itemprop="email">gianluca.russo@studenti.unina.it</a></div>
```

```
Item1
Type :  http://schema.org/Person
Properties
        email : gianluca.russo@studenti.unina.it
        name :  Gianluca
        familyName :    Russo
        jobTitle :      Engineer
        address :       Item2
        worksFor :      Item3


Item3
Type :  http://schema.org/Organization
Properties
        name :  Nilobit Company





Item2
Type :  http://schema.org/PostalAddress
Properties
        streetAddress : Piazzale Tecchio
        addressLocality :       Fuorigrotta
        addressRegion : Napoli
        addressCountry :        Italia
```

Figure 3.13: Four views for semantic text annotation

The user can easily switch between these views and even use them in parallel. The views are synchronized so that applying changes in one of these views automatically updates other views.

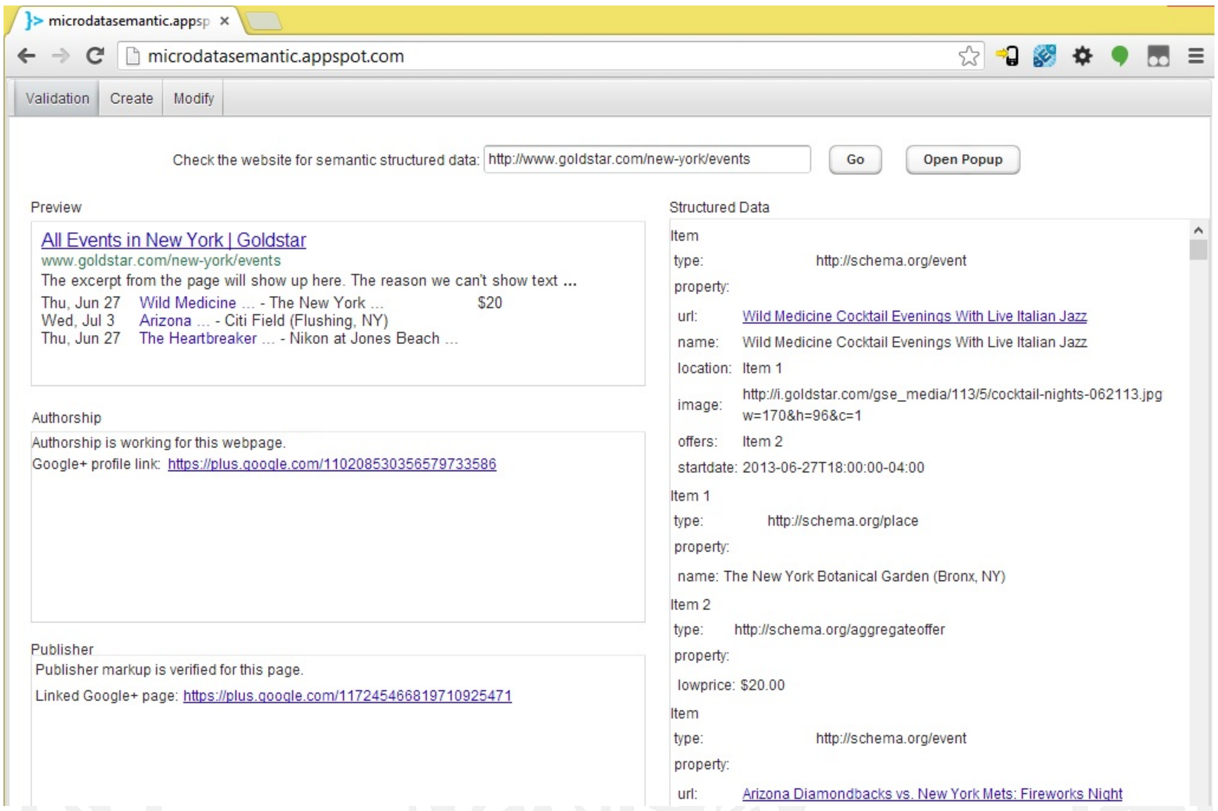
Template View: This view allows to easily create web content that responds to the ontology defined by the Schema.org vocabulary. Starting from the category selection, this view provides all the properties related to the content to be created, allowing the user to take no action on the source code to specify the relationships and semantic tags of the content. This view is also connected directly to the preview of the created content.

WYSIWYG View: The What-You-See-Is-What-You-Get view is the classical interface for rich-text authoring. WYSIWYG text authoring is meanwhile ubiquitous on the web and part of the most content creation and management workflows. This view can also show the blocks that compose the content, allowing the user to see different attributes used to insert the semantic within the text. In this view, as well as using the created content inside the tool, it is possible to copy directly another content created externally. This can allow in association with Microdata Entity View, to extract the semantic entities within the content.

Source Code View: The source code view shows the HTML source of the text inlcluding the Microdata annotations. This view is primarily intended for software engineers supervising the publication workflow as well as knowledge engineers. Since it is possible to change the source code to update or change the content, this operation is however very delicate and requires an advanced knowledge of markup techniques and text formatting.

Microdata Entity View: This view summarizes all the entities, which can be extracted from the annotated text. It provides a deeper semantic view when is compared to the Template view. The Microdata Entity view is (as all other views) updateable, by using the button "View Semantic items". This view is useful to curators and to a lesser extent to the authors for quickly verifying that entities and properties were correctly annotated.

Finally, the last view, presented below, allows to verify the entities and properties connected to a web document. By using the external service Google Rich Snippet, this view provides different information about the content of the web page.



Check the website for semantic structured data: http://www.goldstar.com/new-york/events

Preview

All Events in New York | Goldstar
www.goldstar.com/new-york/events
The excerpt from the page will show up here. The reason we can't show text ...
Thu, Jun 27 Wild Medicine ... - The New York ... $20
Wed, Jul 3 Arizona ... - Citi Field (Flushing, NY)
Thu, Jun 27 The Heartbreaker ... - Nikon at Jones Beach ...

Authorship

Authorship is working for this webpage.
Google+ profile link: https://plus.google.com/110208530356579733586

Publisher

Publisher markup is verified for this page.
Linked Google+ page: https://plus.google.com/117245466819710925471

Structured Data

```
Item
 type:                http://schema.org/event
 property:
  url:      Wild Medicine Cocktail Evenings With Live Italian Jazz
  name:     Wild Medicine Cocktail Evenings With Live Italian Jazz
  location: Item 1
  image:    http://i.goldstar.com/gse_media/113/5/cocktail-nights-062113.jpg
            w=170&h=96&c=1
  offers:   Item 2
  startdate: 2013-06-27T18:00:00-04:00
Item 1
 type:          http://schema.org/place
 property:
  name: The New York Botanical Garden (Bronx, NY)
Item 2
 type:      http://schema.org/aggregateoffer
 property:
  lowprice: $20.00
Item
 type:                http://schema.org/event
 property:
  url:      Arizona Diamondbacks vs. New York Mets: Fireworks Night
```

Figure 3.14: Web page entities extraction

The Preview field shows how this content is showed up in search result (of Google search Engine) when a search for this content is performed. The Authorship field provides information about the author of the document. This function checks the attribution of a page's content by an author, who has set up a public profile on Google's social network (Google Plus)[3]. Publisher field shows the same information as well as Authorship field does, but in this case for Publisher. Lastly, Structured Data field, shows all the entity and their properties recognized within the web page.

3.4.5.4 Class Diagram

Fig.3.15 shows the class diagram.

[3]https://plus.google.com/authorship

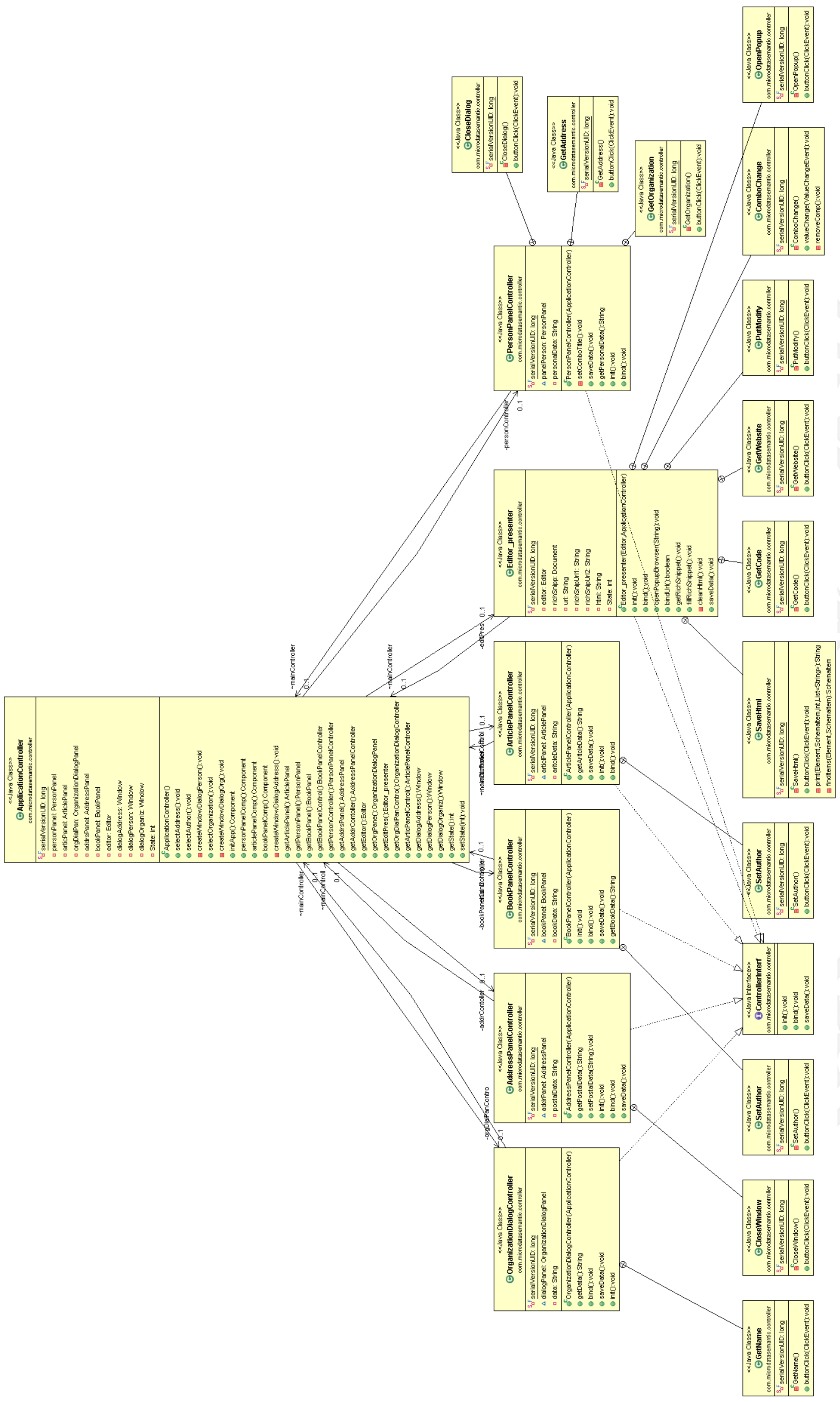


Figure 3.15: Class Diagram

3.4.5.5 Package Diagram

Fig.3.16 shows the Package Diagram.

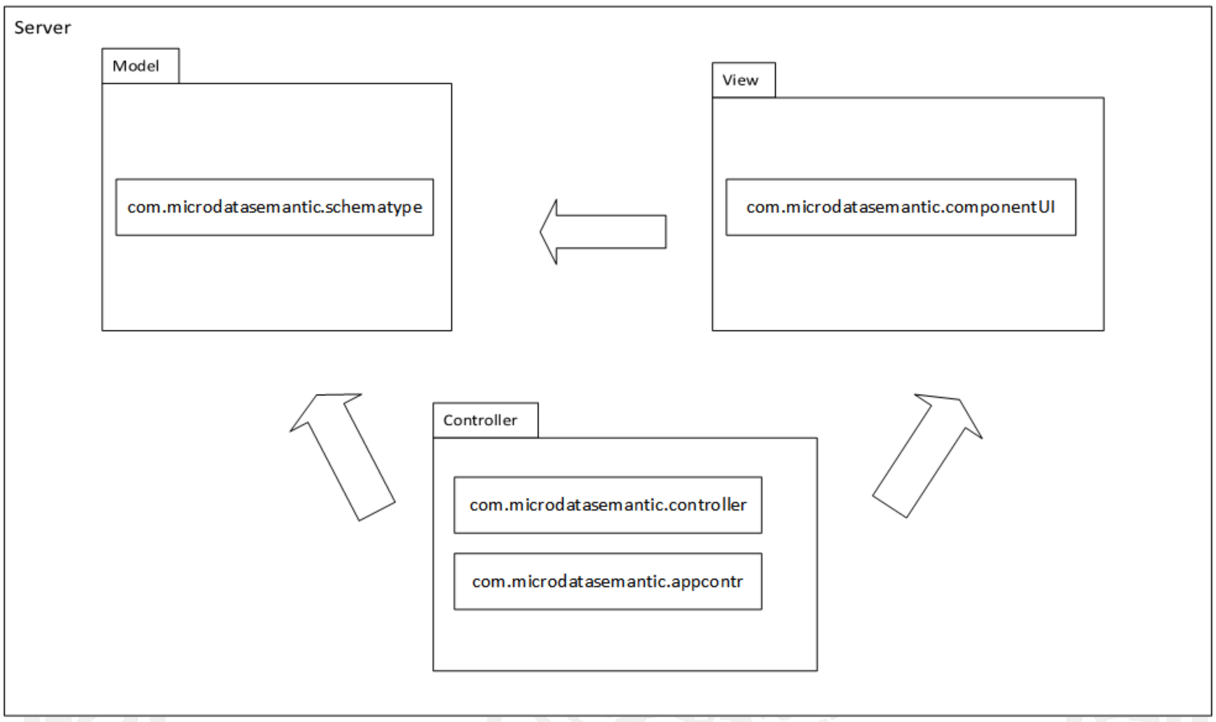

Server
Model
com.microdatasemantic.schematype
View
com.microdatasemantic.componentUI
Controller
com.microdatasemantic.controller
com.microdatasemantic.appcontr

Figure 3.16: Package Diagram

Chapter 4

Case Study

This chapter describes the case study developed during my internship. It introduces the company and business issues, objectives and research questions that have determined the flow study. The work was analyzed using the typical parameters of information retrieval that are collected in the results obtained and discussed at the end of the chapter. Furthermore an empirical study to verify the impact of semantically enriched documents on search engine was conducted. Lastly, an overview of related work is presented, highlighting differences and benefits of MicrodataSemantic tool.

4.1 Company presentation and business problems

This thesis is largely based on the research and development work carried out at the headquarters of the Nilobit[1] company located in Prague, Czech Republic.

Nilobit is a software house founded in April 2004 by a spin-off of the best qualified professionals from large companies within the IT sector, with offices in Italy, Czech Republic and Slovakia Republic. Nilobit is engaged every day in the development of new technologically advanced solutions to improve the performance and business objectives of their customers. Their mission is to support companies to simplify and speed up their workflow, optimize their technological structure, improve performance and reduce time and associated costs.

[1] www.nilobit.com

Among the different business units involved in the development and maintenance of enterprise software, the new web division, as well as providing an integrated system for the management of web portals, has introduced SEO services to support its customers.

By offering this service, Nilobit aims to improve the ranking of their customers in the SERP. Following the study and research carried out during my internship, the company has chosen to address various issues and problems related to optimization techniques for search engines.

During the development of new web portals, the company has therefore put in place all the various techniques that follow the guidelines to improve the quality of web pages. Such techniques that operate directly on the pages of the website, are the so-called "On-Page SEO" and range from ad hoc structuring of the individual URL, to content review for specific keywords. In addition to this traditional approach to SEO, the company wanted to introduce the semantic enrichment of web pages for better results in SERP as a new technique.

This technique, still relatively new, has many aspects that require a detailed study. These aspects cover either purely implementation issues, or the study of involvement that this technique has, to improve the ranking. During the study several issues were faced, relating to how and what techniques to use for markup web pages. Furthermore, what is the impact of the study of this technique on the SERP as well as indirectly on the traffic generated by the improvement of ranking.

The research and analysis were conducted on the company website at an early stage, in order to study the results and then consider the possible application of this technique to their customers. Different problems were highlighted by the web business unit, which initially adopted semantic markup techniques. In the first place, the semantic annotations are manually generated inside source code. This operation involves a series of actions. First, the document's content is contextualised to identify which category the content belong to, and what are the entities, properties and relationships to be marked. Second, to annotate semantically the content, which means being able to access source code for modification.

The last step is to verify the accuracy of these semantic tags. Furthermore, this last step is carried out through the use of external tools, but only after publishing content on the web. This means that even if the content has semantic errors, it must first be published on the internet to be validated, in terms of semantic entities recognized.

4.2 Objectives and Research questions

Starting from the problems highlighted in the previous paragraph, manual techniques as shown, is highly time-consuming, with a high probability of making errors. It is necessary to have a tool that would automate the semantic markup process.

The main objectives of this thesis, are linked to the development of a tool that could automate the semantic annotation process. In the first place, there is the study of the semantic annotations context. This means, a depth study of the most recent techniques, languages and tools that support semantic annotations. In other terms, to determine which are the best techniques and standardized languages that are able to support semantic annotation, even from SEO viewpoint. Secondly, to develop a tool, which is able to better respond to business needs. In this context, such a tool should provide a semantic text authoring integrated environment, which offers the possibility to verify the created content accuracy.

The main goal then, is to enable an environment to create semantically enriched content in an automatic way, that is able to verify the accuracy of entities, properties and the relationships between them, without using external tools. The third objective, which follows directly from the second, is to evaluate the ranking of semantically enriched web pages. The ranking changes should not just be seen as better ranking in the search result page, but as a greater amount of information displayed too. This goal obviously, can only be achieved if the semantic tags, that express entities and properties, were added on the contents, making such content machine-understandable.

The importance of having a unique tool which is able to add semantics to created content and, at the same time to verify the semantic entities accuracy, is remarkable in many aspects. Such a tool will allow to create machine-understandable content-type even to novice users, without having the knowledge of the technologies or languages that are behind the semantic text authoring. Therefore, this would be, like a normal procedure to create the information or content for web pages.

Furthermore, for more experienced users, this environment would provide a range of information that would enable the user itself to evaluate the semantic entity accuracy.

Therefore, from its use, as well as presenting direct benefits for users, even from SEO viewpoint, it will increase the knowledge on the web indirectly. In other terms, the use of this environment, that produces well-formed content from semantic viewpoint (machine-understandable content), allows to create the basis for that shared and reusable knowledge, which is the Semantic Web basis.

Although its benefits are still not directly quantifiable, the Semantic Web of the future will use the web content, which has machine-understandable structure, as base for more sophisticated operations development.

4.2.1 Research Questions

In this context, I have identified several research questions which should highlight some innovative points addressed with this thesis. In the literature[20, 21], the aspects concerning the semantic entities annotations are often separated from the effects that such content may have in the elaboration process by search engines.

These two issues are closely related to each other and in particular the second aspect provides an additional incentive to use the semantic markup process more intensively. This is relatively new as search engines have made changes to their algorithms only recently to show and give more emphasis to machine-understandable content.

These innovative aspects related to the semantic annotation tool development, are formalized through the following research questions:

1. Is The automatic insertion technique of tags using MicrodataSemantic tool able to improve the page's ranking?
2. Is The automatic insertion technique of tags using MicrodataSemantic tool able to improve the page's ranking more than the manual technique?
3. Is The automatic insertion technique able to create content that can be interpreted by the search engines?

4.3 Metrics

The evaluation of this thesis, covers two aspects associated to the semantic annotation context. The first of these, regards the evaluation of MicrodataSemantic tool, introduced in the third chapter. The second aspect concerns the evaluation of the effects of machine-understandable documents on the current web and especially on search engines.

4.3.1 MicrodataSemantic tool metrics

In the overall evaluation of the software created, as well as answering some typical software engineering questions, to evaluate the performance and accuracy of the entities produced by the tool, well-defined metrics are used.

The systematic comparison of the results obtained by the tool with an optimal solution is the methodology used . In other terms, the contents created by the software system are compared with the manual technique, defined here as the optimum.

To compare the results of MicrodataSemantic system, 13 contents were collected in the three categories News Article, Events and Software Application description. For each content the following analysis was performed: Initially, the text was carefully analysed and manually annotated by recognizing entity references to Schema.org classes. Recognized the context, all the properties available for the extracted entities were annotated in reference to Schema.org classes.

As a result, a list of very complex and highly structured entities and properties was obtained. They define a highly semantically enriched content. Then, MicrodataSemantic tool was used to automatically annotate the text by using the predefined template.

The indexes used in this evaluation phase are Precision, Recall, F-Meause and Effort[22], as defined below:

1. Precision: Size of the intersection between the set of entities that would be marked and the entities that have been marked (that is the entities which have been marked correctly by the tool) / size of the set of entities that have been marked, which also includes those that should not have been marked but have been wrongly marked. The precision can be seen as the tool accuracy and fidelity measure.

2. Recall: Size of the intersection between the set of entities that would be marked and the entities that have been marked (that is the entities which have been marked correctly by the tool) / size of the set of entities that should have been marked. Recall can be interpreted as a completeness measure.

3. F-measure: 2 x (x Precision Recall) / (Precision + Recall). This metric measures the test accuracy. It is a combined measure that balances the importance of Precision and Recall.

4. Effort: required time for the semantic annotation, measured for the manual technique and automatic technique provided by the tool. For this measurement, there are several factors

to consider. First, it is a subjective metric, since it is measured by the individual user. This means that in the manual technique, an experienced user with in-depth knowledge of the context and the markup languages, certainly takes less time than a novice user. In any case, this metric gives a comparative information on the effort produced, with the same level of knowledge, among the manual and automatic techniques.

4.3.2 SEO Metrics

The second aspect evaluated in this thesis is associated to how the semantically enriched documents are interpreted by the search engines. In this case, no rigorous metrics and formal definitions are used for the evaluation, but an empirical model is followed, based on events observation.

The results are based on the comparison of two documents categories: those of a machine-understandable type, that is with semantically enriched content, and those of machine-readable type, that is the traditional web documents. The aspects taken into account for the final results formulation, are focused on the ranking changes verification and monitoring of information that the search engines provide for both categories of documents.

In addition, in an entirely heuristic model, the variation of the number of visits has been traced matching with ranking and displayed information changes. In particular, for each document category, two versions were created. Obviously, the content of web pages may not be equal, otherwise it would create a duplicate situation.

In general, the documents belonging to the same type, basically express the same type of information content. More specifically, three types of content have been published on the web: News Articles, Software Application and Events.

For each type, as said, two web pages have been created, with and without semantic annotation. These pages are reachable on[2][3][4][5][6]. The machine-understandable documents type, have been semantically annotated using the MicrodataSemantic tool.

[2]www.softwareparcoauto.it
[3]www.gestione-agenti.com
[4]http://www.camic.cz/a1030-sai-incrementare-la-visibilita-del-tuo-sito/news.tab.it.aspx
[5]http://www.camic.cz/a950-il-tuo-sito-web-e-indicizzato/news.tab.it.aspx
[6]http://www.camic.cz/eventi-dell-anno.tab.it.aspx

The search engine chosen to make these evaluations is Google. This choice was made since Google was the search engine with greater market penetration among the others, according to an analysis of search engines market[7] reported below.

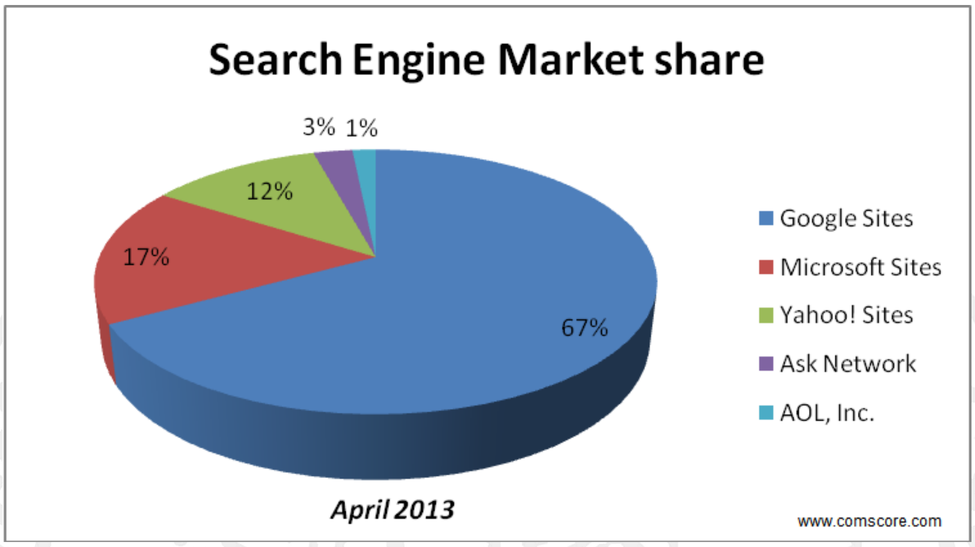
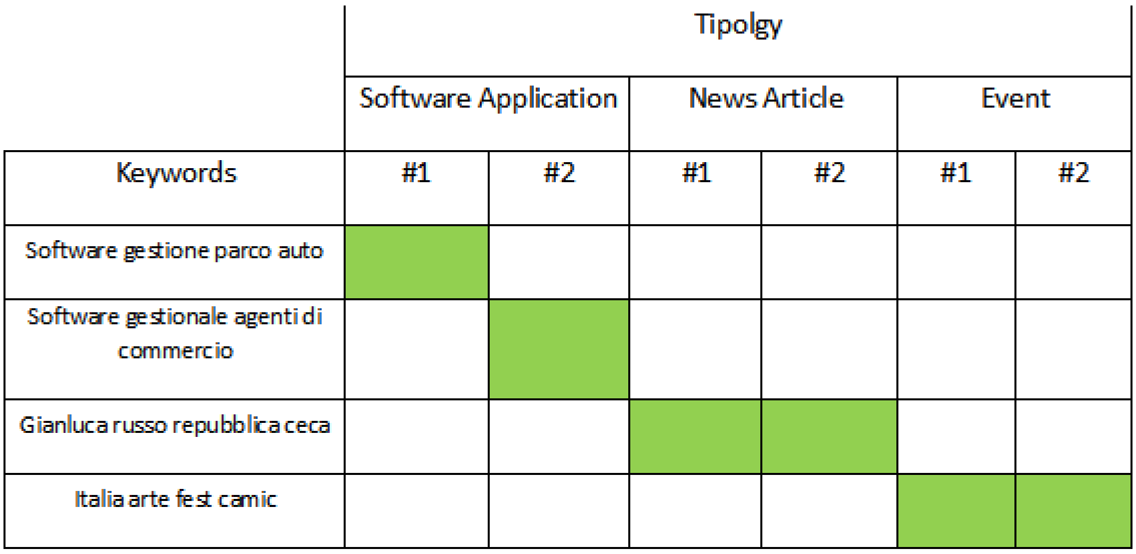


Figure 4.1: Search Engine market share

Moreover, Google is nowadays the search engine that can provide more information in search results with Rich Snippets. Once these web pages are submitted to Google's crawlers[8], the most suitable keywords for the published content have been defined in order to make the query more appropriate.

[7] www.comscore.com
[8] https://www.google.com/webmasters/



	Tipolgy					
	Software Application		News Article		Event	
Keywords	#1	#2	#1	#2	#1	#2
Software gestione parco auto	🟩					
Software gestionale agenti di commercio		🟩				
Gianluca russo repubblica ceca			🟩	🟩		
Italia arte fest camic					🟩	🟩

Figure 4.2: Keywords

During the period of analysis, the following information were traced for the two document category types:

- Changes of ranking;
- Differences between the displayed information in SERP;
- Changes in the number of visits due to variations of ranking and information showed.

The last measured parameter is considered especially for semantically annotated documents. In fact, for these documents, the display of more information in the search results is not immediate, but it is processed later by the search engine. The comparison is performed between the visits obtained in the period in which the documents are shown with more information in the SERP, and the previous one that is without it.

4.4 Results

The results obtained from the evaluation of the metrics presented in the previous section, are shown below, dividing the analysis for the MicrodataSemantic tool results, and the SEO results.

4.4.1 MicrodataSemantic tool results

Before presenting the results obtained for the MicrodataSemantic software, some aspects concerning these produced results must be considered.

The first aspect to be considered is related to the fact that the documents were semantically marked by using either the automatic tool, or through the manual technique. In the latter technique, which generates the optimum to be achieved by the automatic tool, the semantic annotations are performed by an experienced user with an advanced understanding of the context.

This aspect therefore, affects the measurement of the effort to produce semantic annotations. In the following, we refer to "entity" in the results, referring to either entity, or properties related to the document's content. This is the second aspect to be considered.

The test documents on which the analysis was carried out, were subjected to semantically markup twice, in the first instance through the manual technique and in the second place with the automatic technique provided by the tool. The following table shows for each document, the total number of entities available in the content, and those that the software is able to annotate.

Content	***N° Entity***	***Entities Annotated***	***% Annotated***
News#1	53	15	28.3%
News#2	28	8	28.6%
News#3	35	10	28.5%
News#4	22	5	22.7%
News#5	14	2	14.2%
News#6	18	12	66.6%
News#7	12	7	87.5%
News#8	11	4	80%
Event#1	18	11	68.7%
Event#2	19	12	85.7%
Event#3	21	14	70%
Software App#1	50	41	82%
Software App#2	56	44	78.5%
Total Average	***27.4***	***14.2***	***51.8%***

Table 4.1: Entities annotated

The following graph highlights more clearly in percentage the results obtained by the tool:

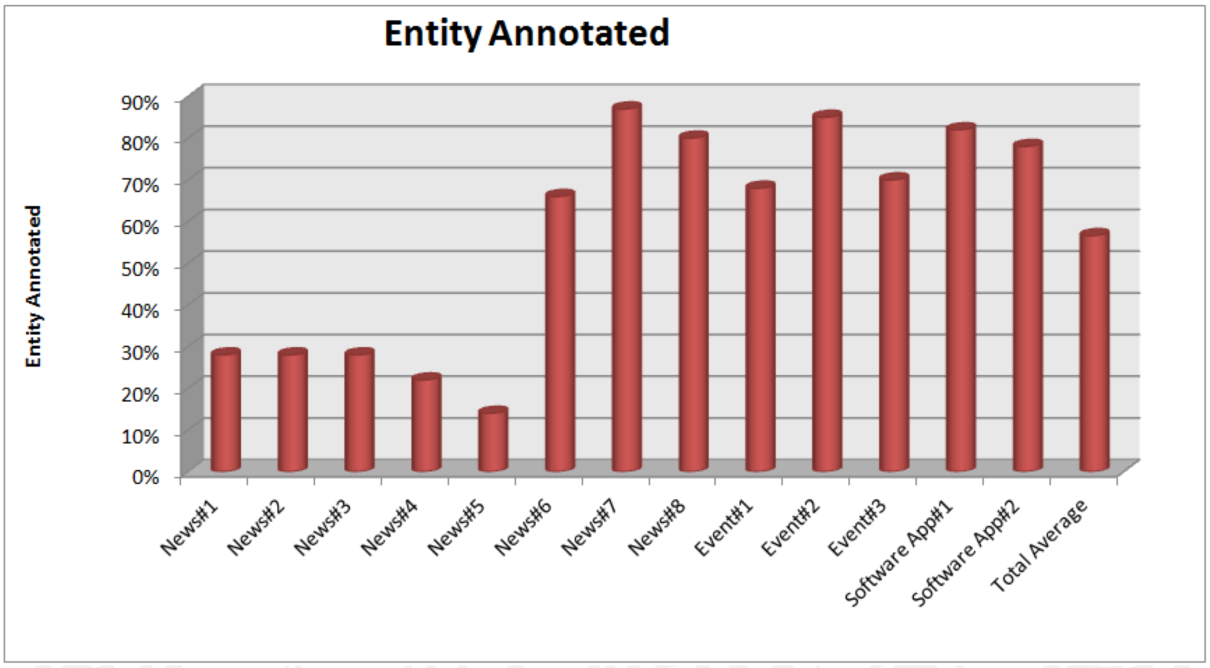


Figure 4.3: Percentage of entities annotated

The percentages of entities annotated by the tool are differentiated by category. In particular for the News Article category, the results are quite low, with an average of 40% of annotations on the overall marked entity.

This result is mainly due to the impossibility to semantically annotate the Body content of an Article. For those results with low percentage most of the entities are located in the Body Article. Instead, better values were measured for that articles with a low number of entities in the body. For the Event categories, the software responds well to the markup needs, but even here it has the limitation of not being able to semantically annotate the entities in the Event description.

For Software application category there is the same problem highlighted above, but the template structure of the software aplication description allows to semantically annotate a large number of entities.

From these results it is possible to measure for each document Precision, Recall and F-Measure value, shown in the following table. Moreover, the table contain Effort values products either from the manual techniques, or from the automatic tool, measured in minutes.

Content	***Precision***	***Recall***	***F-measure***	***Effort by tool***	***Effort by user***	***% time***
News#1	0.68	0.28	0.39	3	16	
News#2	0.72	0.29	0.41	2	12	
News#3	0.71	0.28	0.40	3	13	
News#4	0.71	0.22	0.34	2	11	
News#5	1	0.14	0.25	3	8	
News#6	0.63	0.66	0.64	2	8	
News#7	0.63	0.58	0.60	2	5	
News#8	1	0.36	0.53	2	5	
Average	***0.76***	***0.37***	***0.44***	***2***	***11***	***81%***
Event#1	0.73	0.61	0.66	2	7	
Event#2	0.66	0.63	0.64	2	7	
Event#3	0.7	0.66	0.68	3	8	
Average	***0.70***	***0.63***	***0.66***	***2***	***7***	***71%***
Software App#1	0.71	0.82	0.76	4	14	
Software App#2	0.71	0.78	0.74	4	16	
Average	***0.71***	***0.80***	***0.75***	***4***	***15***	***73%***
Total Average	***0.73***	***0.48***	***0.61***	***2.6***	***11***	***76%***

Table 4.2: Precision, Recall, Effort and F-measure values

The following graph shows the percentages obtained in terms of Precision, Recall and F-measure:

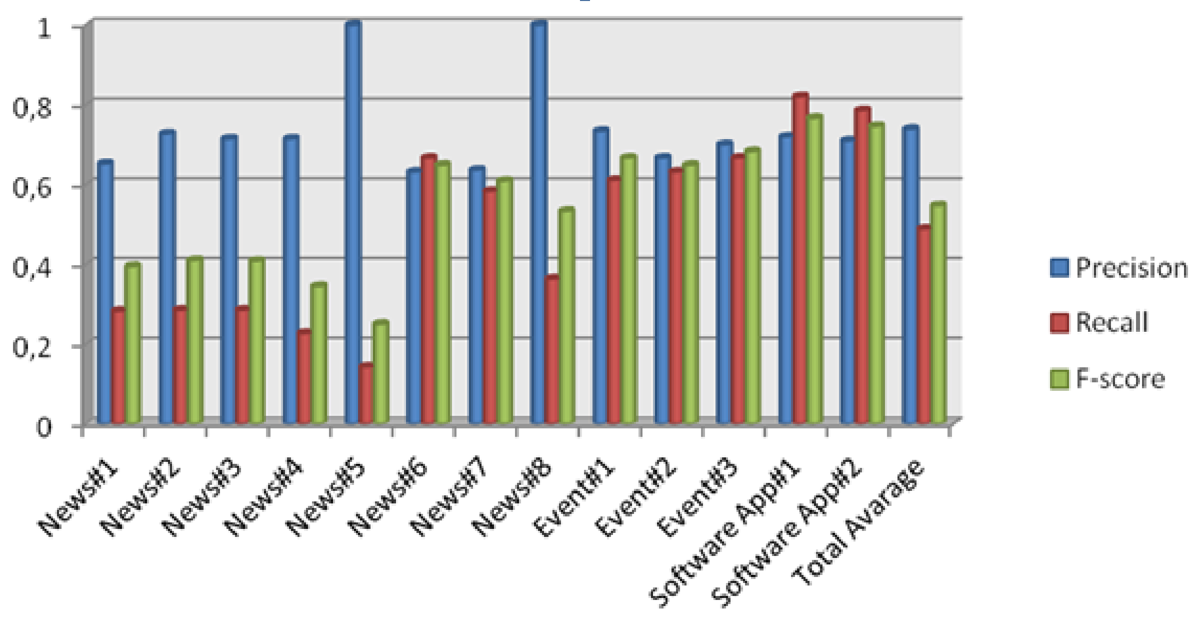


Figure 4.4: Precision Recall and F-score values

A first result of the analysis is that the Recall values depends on the document type. In News Article, Recall value is lower than those obtained from the Event and Software Application documents. This indicates that the coverage of the annotated entities for Article documents is not complete as well as for Event and Software Application documents, for which the values of Recall, Precision and F-measure, overcome the total average.

A further interesting result is the Effort for produce the semantic annotation, compared between the manual and automatic technique. The following graph shows the improvement in terms of saved time by using MicrodataSemantic tool :

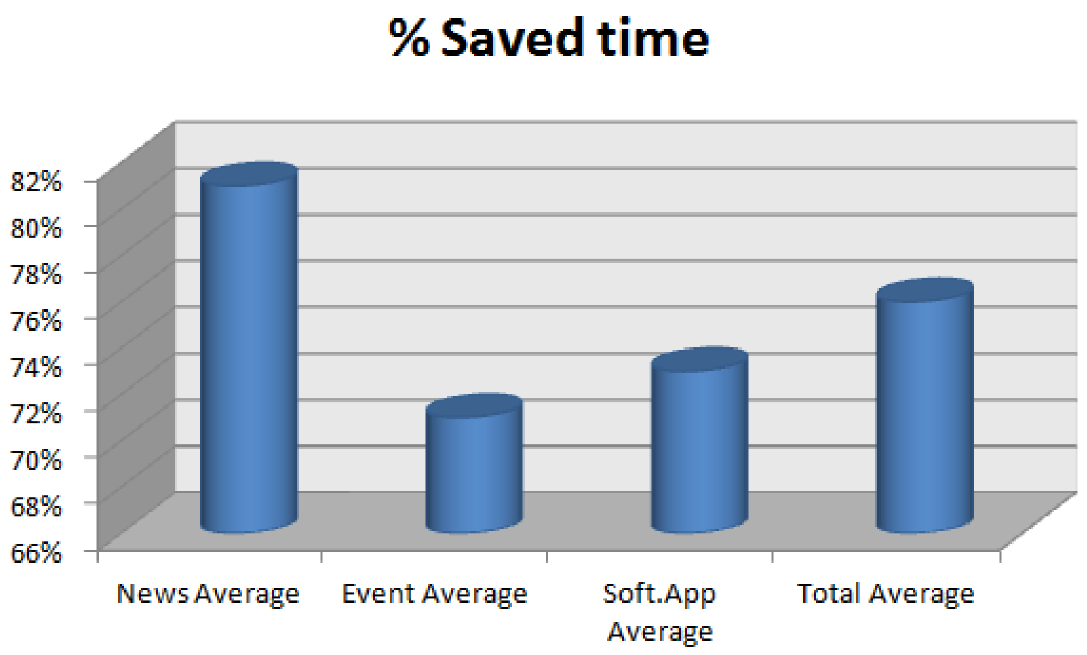


Figure 4.5: Percentage of saved time by using MicrodataSemantic tool

Obviously the result is to be considered at the same level of knowledge. Even if a good average is obtained, that is a 76% saved time for marking the content by using the automated tool, the number of annotated entity is smaller than the manual technique. This situation is clear for News Article category, where at 81% of decrease time the percentage of marked entities is just 32% compared with manual technique.

The graph below shows a summary of the whole average values obtained by the tool.

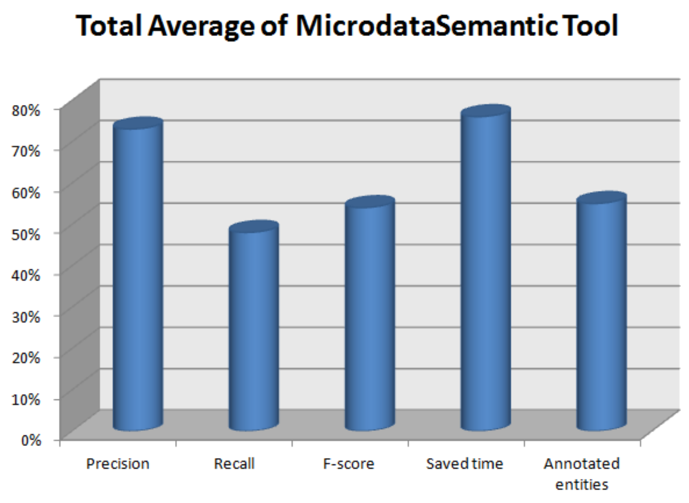


Figure 4.6: Total average of MicrodataSemantic Tool

4.4.2 SEO results

The results collected to study the influence of semantically enhanced documents were divided into three categories: ranking, differences of information presented and changes of the number of visits.

Ranking

In the ranking analysis, as already introduced in metrics paragraph, the ranking of documents (machine-understandable and machine-readable) published on the web were traced. The tracking was performed using the Chrome App SEO SERP Workbench[9] and the search engine selected was Google.com.

For the Event documents category, trends were plotted for two web pages[10] [11] which have a similar content. This has also facilitated the keyword to choose in order to track rankings of both pages. Since both documents describe an Italian music festival published on the Italian-Czech Chamber of Commerce website (a Nilobit Partner), "Italian art fest camic" is the keyword chosen. The results obtained are the following:

[9]https://chrome.google.com/webstore/detail/seo-serp-workbench
[10]http://www.camic.cz/v177-italia-arte-fest-umbria-music-fest/eventi-precedenti.tab.it.aspx
[11]http://www.camic.cz/v245-italia-arte-fest-praga/eventi-dell-anno.tab.it.aspx

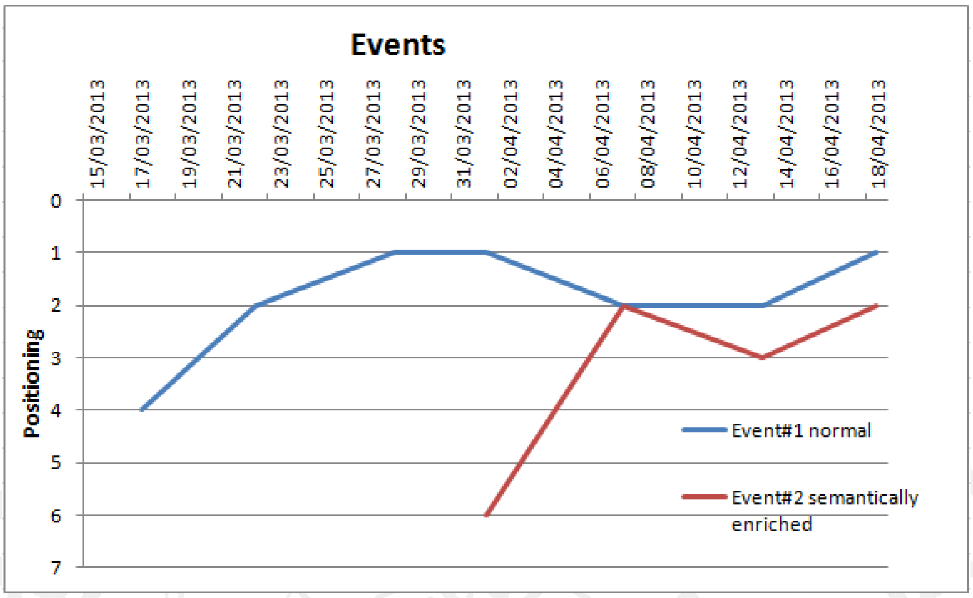


Figure 4.7: Event ranking

These two documents were published within a week's time, this explains the lack of some values for Event # 2, the document with semantic tags. For both web pages, they are within the first page and in the first positions after few days.

For the Software Application documents category two web pages were created which describes two Nilobit business software. Since these two pages have different content, it is not possible to apply the same procedure as above. Then, two different keywords were chosen for these two web pages.

The keyword chosen for the first web page with semantically annotated content[12] was *"software gestionale agenti di commercio"*. The page ranking, initially between the 30th and the 40th position (green line in the following graph), had a sharp rise up to the first page after few days. In conjunction with ranking improvement, a more information appeared in the search results for this webpage.

[12]www.gestione-agenti.com

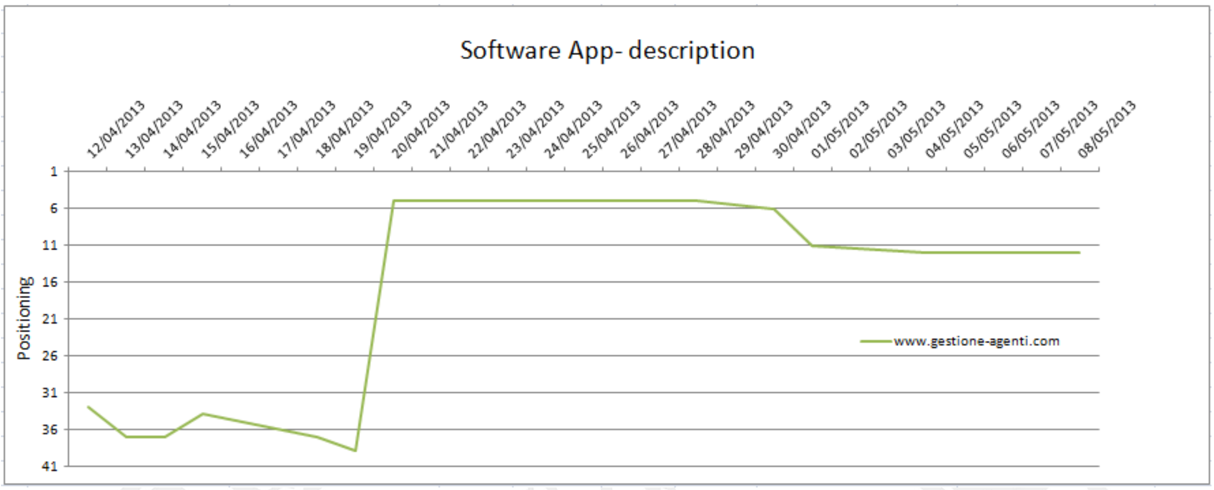


Figure 4.8: Software Application#1 ranking

The second web page published[13] had a constant trend, oscillating between the second and third page of search results. This is the web page witch content that is not semantically enriched.

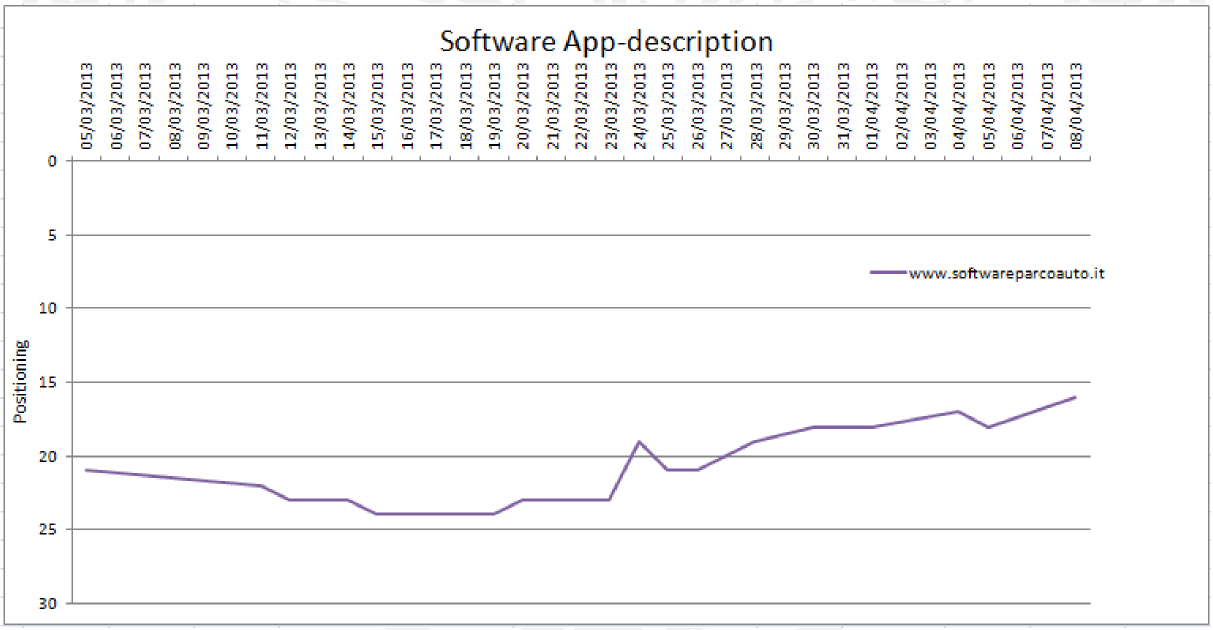


Figure 4.9: Software Application#2 ranking

Considering the News Article documents category, the trend has been traced for two articles published on the Italian-Czech Chamber of Commerce website[14][15]. Since both articles have

[13]www.softwareparcoauto.it
[14]http://www.camic.cz/a1030-sai-incrementare-la-visibilita-del-tuo-sito/news.tab.it.aspx
[15]http://www.camic.cz/a950-il-tuo-sito-web-e-indicizzato/news.tab.it.aspx

the author and the term Czech Republic in common, "gianluca russo repubblica ceca" was the keyword used for search queries. In particular, the two articles were published within a month's time. Even in this case, as before, the publishing date does not correspond with the date display in the search engines.

Obviously, the two pages should be thoroughly indexed by the search engine to be extracted. As leaked from the previous results, the document with semantic content after being well interpreted by the search engine, has an improvement in ranking. Even in this case, the positive change of ranking coincided with the increase of the information showed in the search results.

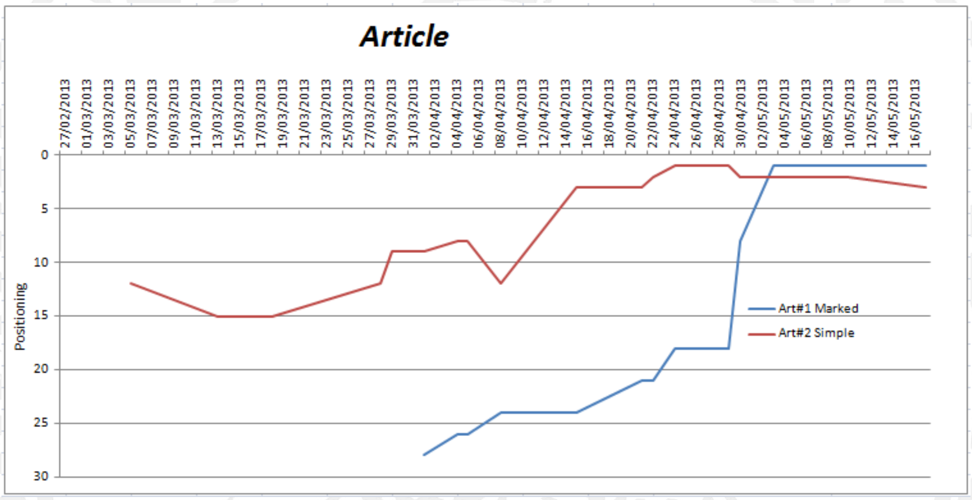


Figure 4.10: Article ranking

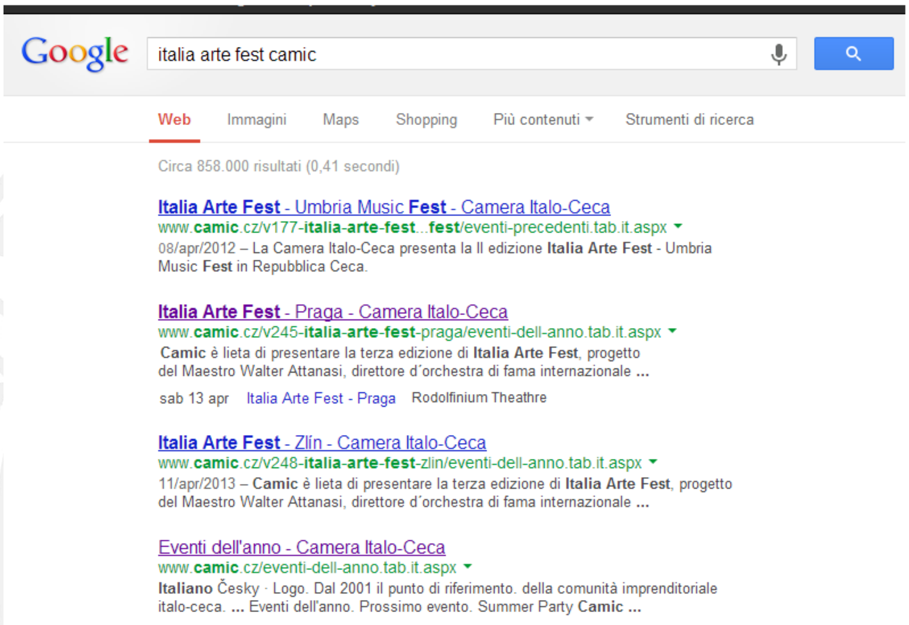
During the results formulation of the semantically enhanced documents, the semantic annotations recognition by search engine was verified through the WebmasterTool/ Structured Data service[16]. It allows to check when and which entity the search engine is able to understand.

Information displayed

The semantically enriched documents have reached even an improvement of the information displayed within the search results. For such documents the search engine, once captured the entities and properties, has provided more details in the search results about the information extracted in the content .

[16]Google WebMaster tool

The following figure shows the Rich Snippets that the search engine has extracted for Event documents. In particular both web pages monitored appear in the search results (first and second position). Furthermore, for the semantically annotated document in second position of this search results, the day, the place, the name and the link to the event described in the web page are extracted. This give more information to the user.



Circa 858.000 risultati (0,41 secondi)

Italia Arte Fest - Umbria Music **Fest** - Camera Italo-Ceca
www.**camic**.cz/v177-**italia-arte-fest**...**fest**/eventi-precedenti.tab.it.aspx
08/apr/2012 – La Camera Italo-Ceca presenta la II edizione **Italia Arte Fest** - Umbria Music **Fest** in Repubblica Ceca.

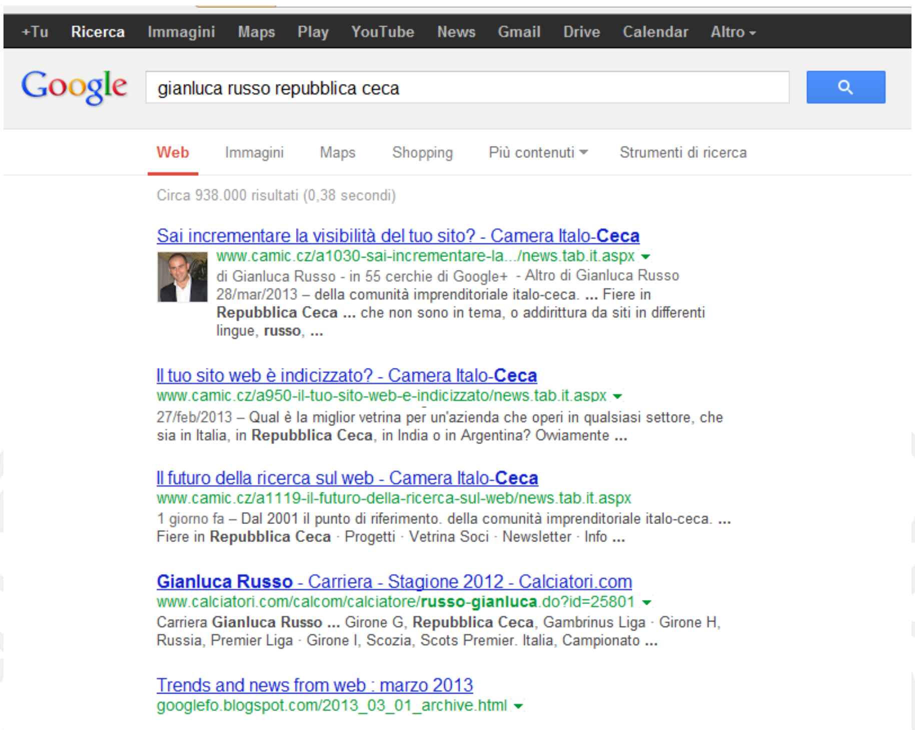
Italia Arte Fest - Praga - Camera Italo-Ceca
www.**camic**.cz/v245-**italia-arte-fest**-praga/eventi-dell-anno.tab.it.aspx
Camic è lieta di presentare la terza edizione di **Italia Arte Fest**, progetto del Maestro Walter Attanasi, direttore d´orchestra di fama internazionale ...
sab 13 apr Italia Arte Fest - Praga Rodolfinium Theathre

Italia Arte Fest - Zlin - Camera Italo-Ceca
www.**camic**.cz/v248-**italia-arte-fest**-zlin/eventi-dell-anno.tab.it.aspx
11/apr/2013 – **Camic** è lieta di presentare la terza edizione di **Italia Arte Fest**, progetto del Maestro Walter Attanasi, direttore d´orchestra di fama internazionale ...

Eventi dell'anno - Camera Italo-Ceca
www.**camic**.cz/eventi-dell-anno.tab.it.aspx
Italiano Česky · Logo. Dal 2001 il punto di riferimento. della comunità imprenditoriale italo-ceca. ... Eventi dell'anno. Prossimo evento. Summer Party **Camic** ...

Figure 4.11: Rich Snippet #1

Even for News Article and Software Application web pages, same phenomenon presented above was observed. In particular, the following figure shows the information that the search engine extracts from the News Article semantically enriched web page. Even in this case, the two types of documents (with and without semantically enriched contents) appear in the search results. For the first category, information about the author, date of publication and a link to other contents of the author are extracted. The second category is instead listed as the traditional result. They respectively are in first and second position.



Circa 938.000 risultati (0,38 secondi)

Sai incrementare la visibilità del tuo sito? - Camera Italo-**Ceca**
www.camic.cz/a1030-sai-incrementare-la.../news.tab.it.aspx
di Gianluca Russo - in 55 cerchie di Google+ - Altro di Gianluca Russo
28/mar/2013 – della comunità imprenditoriale italo-ceca. ... Fiere in **Repubblica Ceca** ... che non sono in tema, o addirittura da siti in differenti lingue, **russo**, ...

Il tuo sito web è indicizzato? - Camera Italo-**Ceca**
www.camic.cz/a950-il-tuo-sito-web-e-indicizzato/news.tab.it.aspx
27/feb/2013 – Qual è la miglior vetrina per un'azienda che operi in qualsiasi settore, che sia in Italia, in **Repubblica Ceca**, in India o in Argentina? Ovviamente ...

Il futuro della ricerca sul web - Camera Italo-**Ceca**
www.camic.cz/a1119-il-futuro-della-ricerca-sul-web/news.tab.it.aspx
1 giorno fa – Dal 2001 il punto di riferimento. della comunità imprenditoriale italo-ceca. ... Fiere in **Repubblica Ceca** · Progetti · Vetrina Soci · Newsletter · Info ...

Gianluca Russo - Carriera - Stagione 2012 - Calciatori.com
www.calciatori.com/calcom/calciatore/**russo-gianluca**.do?id=25801
Carriera **Gianluca Russo** ... Girone G, **Repubblica Ceca**, Gambrinus Liga · Girone H, Russia, Premier Liga · Girone I, Scozia, Scots Premier. Italia, Campionato ...

Trends and news from web : marzo 2013
googlefo.blogspot.com/2013_03_01_archive.html

Figure 4.12: Rich Snippet #2

In this example should be noted that Google search engine shows the information about the author of the web page only in a particular case. In the semantically enriched document, the author entity must be connected to a verified public profile on Google Plus (G+) social network. In this way, in the search results is also possible to display the profile picture on the side of the link, as in the case of example.

Change in number of visits

Another important issue is associated to the result of the two aspects previously observed. The final analysis of the semantically annotated documents study is related to the variation of visits observed in relation to the change of ranking and to the increase of information in the search results. This measurement was traced for Software Application web page[17] and the number of visits, over a period of one month, were analyzed through Google Analytics[18]. The result is shown in the following figure :

[17]www.gesione-agenti.com
[18]http://www.google.com/analytics/

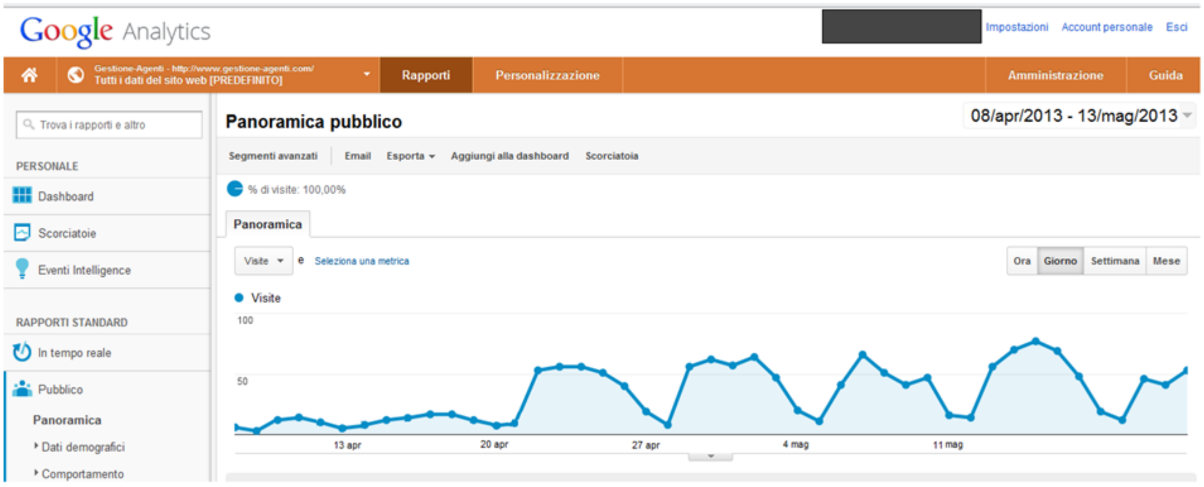


Figure 4.13: Change in visits

The bell-shaped trend is the classic weekly visit. During weekdays the number of visitors decreases drastically. The further observation is on the growing of visits after April 20th. Comparing the results highlighted previously on this web page this corresponds to the drastic increment in ranking from the 30th to 5th position, and also to the increase of information in the SERP's. The positive aspect of the two results (improve in ranking and information) have certainly contributed to the visits increase, so we can cocnlude a direct link between these results. In final analysis, an increment of the visits number can be observed week by week as a trend.

4.5 Discussion

The results presented above have a very important value, because they express the actual values measured in the study. Considering some data individually can lead to hurried and unreasonable conclusions. Therefore we must read these results more carefully, according to the context in which they are explained. Starting from the results obtained for the MicrodataSemantic tool, the first and intuitive conclusion is that this is a not completely effective tool, that fails to explain the all semantic information of the content.

The average values of Recall obtained for News Article category were lower compared with other categories of documents analyzed. A plausible reason for this result is due to the fact that most of the entities to be marked are contained in the Article body. Consider for example an article that has an high number of references to people, organizations, descriptions of events, etc. in the article body. For all of these entities, MicrodataSemantic tool is not able to effectively respond to a flexible and complete annotating demand.

The reason for this limitation reside in the template-based adoption for marking up the content. The tool, starting from a single entity that describes the main meaning of the page, provides the annotations for the complex and simple entity properties. However, the system has demonstrated good accuracy when Events and Software Application documents are annotated. The system offers in these cases, the capability to semantically annotate a greater number of entities.

Nevertheless, also in these two categories the system suffers from the impossibility to markup all the entities in the content. Furthermore, the Precision values evaluated from the results analysis shows that in two cases the tool has reached the maximum value. Behind this result, however, the content of such documents has a limited number of entities to be marked.

The Top-Down approach chosen for the semantic annotation, of course comes up against the need of a strong marking flexibility. When compared with the Bottom-Up approach, this approach deal with semantic representations from the beginning instead of lifting unstructured content to a semantic level. However, analyzing the results of effort to create semantically enriched content, there are considerable differences in terms of time. By comparing the time effort for mark up same documents through the two annotation techniques, two results can be observed. The first is that by using MicrodataSemantic tool the user is able to obtain a significant reduction of marking time, which is an average of 76% saved time compared with the manual technique. The second aspect to be considered, however, is related to the number of entity marked by the tool, which is lower than those founded and marked with the manual technique.

Nevertheless, in some specific cases, the tool is able to keep a comparable level of annotated entity compared to the manual technique, but with a substantial advantage in terms of time. Lastly, the times obtained in manual technique were measured on an "advanced" user, with a thorough understanding of the covered topics. Considering the obtained results, the manual technique has severe limitations for traditional users because the semantic annotating time would be higher and with a higher probability of errors.

From the analysis conducted on the three examined metrics, we obtained an improvement either for the ranking, or for the amount of information displayed in the SERPs, and in the increase of visits number. As pointed out in the previous paragraph, the semantically enhanced documents (through MicrodataSemantic tool) have had more value than traditional web documents. In this analysis, all three types of published documents (News Article, Software Application, Event) have obtained an improvement without distinction for their content. The ranking improvement of the machine-understandable documents is also followed by an increase of information displayed in search results.

An interesting result to point out is related to the increase of visits observed for the Software Application web page (semantically enriched). For this document, after the changes in ranking, from 30th to fifth positioning in the SERP and with the increased information displayed, there was an increase of 150% in the number of visits.

Considering instead the research questions introduced at the beginning of the chapter, it is possible to answer positively to these questions. During the analysis we observed a positive trend for the ranking of the semantically annotated web pages compared to traditional documents. The semantically enhanced contents have brought the web pages to the top of search results. Furthermore, in relation to the third research question, the search engine has provided more information in the SERP for these documents, unlike traditional documents. This implicitly means that the search engine has correctly read and recognize the semantically annotated entities provided in the content, and has offered additional information to the user in the search results.

Analysing the system software developed, there are considerable implicit benefits in the implementation of this web application architecture. Based on the Google AppEngine cloud infrastructure, MicrodataSemantic Tool can benefit of the scalability, performance, and reliability features provided by GAE. MicrodataSemantic tool runs on a traditional Web browser, without the need to install any software or plugin, using Ajax technology to provide a significant increase in usability of the user interface to the user.

The system is easily maintainable, since using the Model View Controller design pattern is possible to increase the functions and characteristics of the software without increasing the complexity. Following the technologies and languages that are standardized, for future needs is possible to adopt different annotating language or vocabulary, without having to make substantial changes to the software. In this way, the system is open to the adoption of new standards.

4.6 Related Work

There are already many Semantic Annotation tool available that can be distinguished for the approach or for the type of annotation used. RADiFy[19], WYMeditor[20], Datapress[23], Loomp[21] and FLERSA[24] are some examples of annotating tool which adopt the bottom-up approach. They implement RDF as semantic markup language, using different vocabularies such as Dublin Core, FOAF etc.. Another tool that adopts the bottom-up approach is RDFaCE[22], a versatile tool that in addition to RDFa Lite markup language, has recently introduced Microdata as additional markup language and Schema.org as vocabulary. It also features advanced suggestion of entities functionality.

Other tools that adopt the top-down approach as MicrodataSemantic are OntoFeeder[23], a plugin for the Content Management System that uses RDFa and Microformats to add meta information. SAHA 3[24] is another example of a browser-based tool for semantic annotation of web content (with RDF) that uses the top-down approach.

Other tools suggested by the Schema.org vocabulary for the semantic annotation of content[25] are Schema Creator[26], The Microdata Generator[27]. These tools can be considered as direct competitors of MicrodataSemantic tool. Another suggested tool is web.instata[28] but it offer a translating system for markup the content either with Microdata format, or RDFa Lite format.

The following table provides a comparison between the two annotation tools suggested for the Schema.org vocabulary, the tool with Bottom-up approach RDFaCE and MicrodataSemantic, based on the quality attributes that are introduced below:

Usability: Usability is a measure of the quality of a user's experience in interacting with a system. In[25] , usability is defined as consisting of the six factors:

1. Fit for use (or functionality). The system can support the tasks that the user has in real life.

[19]http://duncangrant.co.uk/radify/
[20]http://www.wymeditor.org/
[21]http://loomp.org/
[22]http://rdface.aksw.org/
[23]http://www.ontos.com/
[24]http://www.seco.tkk.fi/services/saha/
[25]http://schema.rdfs.org/tools.html
[26]http://schema-creator.org/
[27]http://www.microdatagenerator.com/
[28]https://github.com/mhausenblas/web.instata

2. Task efficiency. How efficient is it for the frequent user?

3. Ease of remembering. How easy is it to remember for the occasional user?

4. Subjective satisfaction. How satisfied is the user with the system?

5. Understandability. How easy is it to understand what the system does?

Ease of use (or user friendliness) is defined as the combination of factors 2 to 5.

Providing a Single Point of Entry Interface with support of Inline Editing of annotations improves the usability of an authoring system. It means the environment in which users annotate documents should be integrated with the one in which they create, read, verify and edit them. So, there is no added user effort involved in creating a semantic content versus a manual approach, because the real work is done by the software through capturing semantics that is already being provided by the user. Inline editing allows editing items in one single step by clicking on them.

Customizability: Customizability is the ability of a system to be configured according to users' needs and preferences. Instead of being a static form strictly dependent on a given schema, a semantic annotation system should provide different views for different personas using the system.

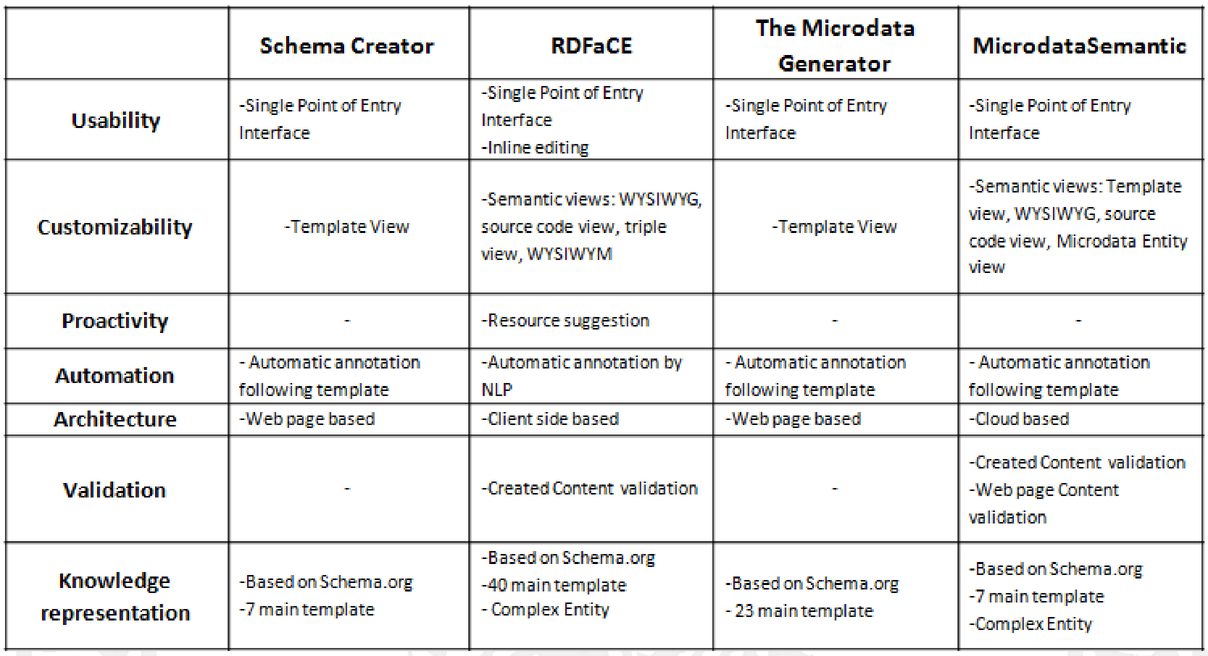
Proactivity: Proactivity is the ability of a system to act in advance of a future situation, rather than just reacting. It means taking control and making things happen rather than just adjusting to a situation or waiting for something to happen.

Automation: Automation is the ability of a system to automatically perform its intended tasks thereby reducing the need for human work. In the context of semantic authoring it means the provision of facilities for automatic markup of documents to facilitate the annotation of documents content.

Architecture: Architecture defines the structure and enables, according to the context, to further define other intrinsic system features such as scalability, robustness, etc.

Validation: In the semantic authoring context, Validation is the ability of a user to validate the created content. A semantic annotation system should support the validation of semantic accuracy integrated within the environment without use external services.

Knowledge representation: knowledge representation is the ability to create machine-understandable content. Based on vocabularies it defines the number of concepts which can be semantically annotated.



	Schema Creator	RDFaCE	The Microdata Generator	MicrodataSemantic
Usability	-Single Point of Entry Interface	-Single Point of Entry Interface -Inline editing	-Single Point of Entry Interface	-Single Point of Entry Interface
Customizability	-Template View	-Semantic views: WYSIWYG, source code view, triple view, WYSIWYM	-Template View	-Semantic views: Template view, WYSIWYG, source code view, Microdata Entity view
Proactivity	-	-Resource suggestion	-	-
Automation	- Automatic annotation following template	-Automatic annotation by NLP	- Automatic annotation following template	- Automatic annotation following template
Architecture	-Web page based	-Client side based	-Web page based	-Cloud based
Validation	-	-Created Content validation	-	-Created Content validation -Web page Content validation
Knowledge representation	-Based on Schema.org -7 main template	-Based on Schema.org -40 main template - Complex Entity	-Based on Schema.org - 23 main template	-Based on Schema.org -7 main template -Complex Entity

Figure 4.14: Tool comparison

Schema Creator is a web application "to help you quickly build and get started with schema.org Microdata". The Schema Creator approach is based on template as MicrodataSemantic tool does. It define several templates where users can enter different properties to create the desired content. It also provides the source code view and a preview of the created content. As a template-based tool for creating semantic annotation, however, Schema Creator does not provide a complete range of properties associated with each entity.

In particular, the user can not insert nested entity inside the content. The end-user of Scheme Creator must therefore shape the content to be marked with the properties available from the system. MicrodataSemantic tool instead, offers the possibility to semantically annotate complex entities in order to create much more structured entities. The Microdata Generator is another editor very similar to Scheme Creator, it offers a greater number of entities that can be semantically annotate but lacks also the possibility of integrate complex entities as Schema Generator does.

RDFaCE is a tool for semantic content creation with Bottom-Up approach. It provides a system to define the individual entities and properties available in the content. It uses both Microdata and RDFa Lite as markup language and supports different vocabularies in addition to the recent introduction of Schema.org. An important aspect of RDFaCE, which makes it a very popular tool in the semantic annotations environment, is entity suggestion functionality. Using external API, RDFaCE uses the services provided by external tool for Natural Language Processing. The recognition of entities in the content and the accuracy provided by these services nowadays are not comprehensive and have many errors of disambiguation. They can be considered a starting point for annotations.

The main difference between RDFaCE and MicrodataSemantic, over the approach adopted, is that RDFaCE provides client-side annotation to modify directly the content, while MicrodataSemantic Relies on the cloud functionality for managing the semantic content lifecycle. Moreover, MicrodataSemantic ensures scalability and the distribution of the tool automatically. The main advantages of MicrodataSemantic compared to other tools are threefold: Providing different views to semantically markup documents as well as supporting automatic content annotation which improves the customizability and automation remarkably.

Furthermore, since MicrodataSemantic processes the annotations server-side within the user's browser and does not require any central storage backend, it is highly scalable with AppEngine cloud technology. Moreover, the most important feature that is lacking in other tools analyzed and makes MicrodataSemantic an innovative tool for the approach used, is the ability to verify the correctness of the semantic content . MicrodataSemantic provides a test environment for semantic annotations that is integrated into the system. There is no need to use external tools for the validation and verification either of semantic entities created, or for external web pages contents. This feature allows the end user to understand more accurately the content created, in addition to facilitating the production of semantic accuracy.

Chapter 5

Conclusion

In this chapter, the reader can find the final observations on this work. In addition to summarising the obtained results and including a final overview of the semantic annotations from either SEO or Semantic Web viewpoint, future developments that have been hypothesized for MicrodataSemantic tool are described.

5.1 Generalisation of the results

The idealization of the Semantic Web envisioned by Berners-Lee was held more than ten years ago. During this period, several specific areas of research were opened, several articles were produced and software tools were developed with the common idea of a "smarter" internet. However, there is still a big gap between the dream of "computers and people who work in cooperation"(Berners-lee 2001) and the reality of the user in his daily routine who needs a more structured way to improve Web search, work and leisure. In a market where information is generated and consumed even more rapidly, the challenge of adopting the idea of the Semantic Web, without loss of time and money is very hard.

The creation of vocabularies, ontologies and taxonomies to describe in a unique manner an environment that, in many ways, is ambiguous and subject to personal interpretation, is not an easy task. Despite this idea, there is still a large gap between the advantages that the Semantic Web can offer and the end user.

Through this challenge, which goes beyond the academia doors and enters the lives of ordinary people, this thesis has attempted to demonstrate that the semantic annotation is the "shortest bridge" between the Berners-Lee dream and practice. Nowadays, though minimal and still in an embryonic state, we can observe some services usable by the user that embrace the Semantic Web idea. Google Knowledge Graph is an example of how it is possible to increase the performance of research by the use of inference. Based on an extensive collection of semantically annotated documents, it provides direct answers to specific research, without the need to resort to the opening the links showed in the search. It also increases the ability to search by displaying the information related to the sought topic.

This is one of the most important aspects that the big search engines are trying to address. It will only be possible to reach this goal if there is a tacit agreement in the Internet ecosystem, consisting of those who generate content (users) and those who manages the extraction (search engines). On the one hand, those who create content must ensure they are machine-understandable. On the other hand, those who process them will have to understand, extract and show the content in the best way for the end user.

Through all the research that has been developed over the years, various concepts and technologies involving the semantic annotation have been investigated, such as the specific vocabularies and languages, essential grains for the consolidation of the proposals made by this work. It could be said that the ultimate goal of this work, which aims to develop a tool for semantic annotations of web content to facilitate its retrieval by search engines, has been reached. With a wider perspective to use this work to semantically express all entities, properties and relationships within a document, MicrodataSemantic tool can appear limited. But this result might not be different, since we are trying to express traditional concepts and relationships that in a normal conversations may have a subjective interpretation. This is a general, inherent limitation of the Semantic Web, that is, to express a concept that can be ambiguous in natural language in a direct and unequivocal way.

The results of the evaluation tests show that semantic annotations generated by the developed tool, have brought significant benefits in the retrieval process by the search engines. In this context no other SEO techniques were used to improve these results. Of course, this result must be viewed in the broader and complex world of search results. It is difficult to scientifically prove that the semantic annotation of web pages led alone to a better ranking.

To get a wider picture we have to consider that complex algorithms and a considerable number of other complex aspects substantially affect the search results. Moreover, although current techniques for better ranking have good results, not all the little details that are considered when the search results are composed are completely clear. This aspect is also obvious in a web that is positioned as democratic and fair. It could be envisaged that there are disparities in situation where the search results are polluted by false and misleading links, attracted only by advertising purposes and general spam as it happened in the early days of search engines. These techniques, whether from the viewpoint of the person who publishes the content whose aim is to increase visibility, from the viewpoint of the web and users in general, are able to create an environment where the content is even more usable and verified.

5.2 Future work

Although MicrodataSemantic offers many positive aspects and helps the semantic annotation of web content for the correct interpretation by the search engines, it is obviously not an unique solution to the problem. This project is at an early stage, but is already being used within the company in which it was developed.

New features and functionality are planned to increase the value of this tool and make it a basic tool for the future business applications. It is worth noting that during the development of this work several issues were raised, and due to time constraints could not be further analyzed, but they are used as suggestions for future work. Examples include:

- Increase the number of templates for the content creation and the number of entities that can be marked. This allows an increase in capacity for the semantic enrichment of content, offering multiple categories and multiple contexts to be marked.
- Insert the new view WYSIWYM (What you see is what you mean) that allows editing in real time and directly in the WYSISYG editor (What you see is what you get) the entities in the content. This would allow an increase in the functionality and usability of the tool for the end user in terms of entities that it is possible to annotate.
- Adding automatic suggestions by NLP (Natural Language Processor), through the use of services that are already available on the market[1][2][3][4]. They do not offer full and complete entities and properties recognition, but they can provide a good starting point for the semantic annotation process.

[1]http://www.alchemyapi.com/
[2]http://www.opencalais.com/
[3]http://www.ontos.com/
[4]http://www.evri.com/

Bibliography

[1] Tim Berners-Lee, James Hendler and Ora Lassila, “*The Semantic Web, A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities*”, Scientific American, May 2001

[2] Jacco van Ossenbruggen, Lynda Hardman and Lloyd Rutledge “*Hypermedia and the Semantic Web: A Research Agenda*”, Journal of digital information, 2002

[3] Tim Berners-Lee, "*Information Management: A Proposal*", CERN (March 1989, May 1990).

[4] SA Dobson, VA Burrill, “*Lightweight Databases*”, 1995

[5] J. Clark. “*XSL Transformations (XSLT)*”, W3C , November 1999. At: http://www.w3.org/TR/1999/REC-xslt-19991116.

[6] Michael Kifer, Jos de Bruijn, Harold Boley, Dieter Fensel, “*A Realistic Architecture for the Semantic Web”,* First International Conference, RuleML 2005, Galway, Ireland, November 10-12, 2005. Proceedings

[7] Ying Ding , Dieter Fensel , Michel Klein , Borys Omelayenko, “*The Semantic Web: Yet Another Hip?”,* Data & Knowledge Engineering, 2002

[8] Gord Hotchkiss, Steve Alston, “*Eye Tracking Study: An in Depth Look at Interactions with Google Using Eye Tracking Methodology*”, Enquiro Search Solutions Incorporated, 2005

[9] Annika Hinze, Ralf Heese, Markus Luczak-Rosch, and Adrian Paschke, ”*Semantic Enrichment by Non-Experts: Usability of Manual Annotation Tools* ”, The Semantic Web – ISWC 2012 Lecture Notes in Computer Science Volume 7649, 2012, pp 165-181

[10] Stephen Dill, Nadav Eiron, David Gibson, Daniel Gruhl, R. Guha, Anant Jhingran, Tapas Kanungo, Sridhar Rajagopalan, Andrew Tomkins, John A. Tomlin, and Jason Y. Zien, "*SemTag and Seeker: Bootstrapping the Semantic Web via Automated Semantic Annotation*", IBM Almaden Research Center, 2003

[11] Jung-Hwa Choi, Young-Tack Park, "*Ontology-based automatic semantic annotation for named entity disambiguation*", Proceeding ISC '07 Proceedings of the 10th IASTED International Conference on Intelligent Systems and Control Pages 257-262 , 2007

[12] Sheth, A., Bertram, C., Avant, D., Hammond, B., Kochut, K., Warke, Y.: Managing "*Semantic Content for the Web*", IEEE Internet Computing, 6(4), (2002) 80-87

[13] Mark Birbeck, "*Microformats and RDFa Are Not as Far Apart as People Think*", JUN 2008

[14] Emily P. Lewis, Tantek Celik, "*Microformats Made Simple*“, October 2009

[15] Tom Heath,Christian Bizer, *"Linked Data: Evolving the Web Into a Global Data Space"*, 2011

[16] Matthew MacDonald, *"HTML5: The Missing Manual"* , 2011

[17] Arvidsson, F.; Flycht-Eriksson, A. "*Ontologies* I" 26 November 2008.

[18] Benjamin Melançon; Jacine Luisi; Károly Négyesi; Greg Anderson; Bojhan Somers; Stéphane Corlosquet; Stefan Freudenberg , *"The Definitive Guide to Drupal 7",* Apress. July, 2011

[19] K.Siorpaes and E. Simperl, *"Human intelligence in the pocess of semantic content creation"*, Journal World Wide Web archive Volume 13 Issue 1-2, March 2010 Pages 33-59

[20] J. L. Navarro-Galindo and J. Samos, *"Manual and automatic semantic annotation of web documents: the flersa tool*", New York,2010

[21] S. Araujo, G.-J. Houben, and D. Schwabe, "*Linkator: Enriching web pages by automatically adding dereferenceable semantic annotations*", 2010

[22] J. Makhoul, F. Kubala, R. Schwartz, and R. Weischedel, "*Performance measures for information extraction*", 1999

[23] E. Benson, A. Marcus, F. Howahl, and D. Karger, *"Talking about data: Sharing richly structured information through blogs and wikis",* in The Semantic Web – ISWC 2010, ser. Lecture Notes in Computer Science

[24] José L. Navarro-Galindo, José Samos, "*The FLERSA tool: adding semantics to a web content management system"* in International Journal of Web Information Systems

[25] S. Lauesen, "*User Interface Design: A Software Engineering Perspective*" Addison Wesley, Feb. 2005