Semantic annotation tool for publishing structured data on the Web

Newton Calegari
Web Technologies Study Center at NIC.br
Av. Nacoes Unidas, 11541, 7th floor
04578-000, Sao Paulo - Brazil
newton@nic.br

ABSTRACT

One of the main challenges faced by the Semantic Web community is to create applications and useful resources that can enhance the end users' needs and this work brings a proposal of a semantic annotation application which aims to include end users into the structured data on the Web ecosystem, allowing them to contribute publishing RDF data in an easy interface and without learning Semantic Web subjects. This paper describes the different methods of semantic annotation and shows components of an application of semantic annotation that is being developed in a real scenario.

CCS Concepts

•**Information systems** → **World Wide Web;** Social tagging;

Keywords

Semantic annotation, structured data, semantic web

1. INTRODUCTION

Semantic Web has been receiving a lot of attention and its benefits can be seen in a range of applications, it has also evolved towards the definition of standards and languages, it does not seem to address the real issue of delivering to the end-users all the potential that Semantic Web has. As pointed by [2], "the Semantic Web's potential to deliver tools that help end users to capture, communicate, and manage information has yet to be full led, and far too little research is going into doing so". The main motivation of this work is to try to fill the gap between real end-users who publish a lot of data and relevant content on the Web, and the Semantic Web application. The approach of this research consists in developing a semantic annotation tool which can be used by content publishers, such as journalists, to annotate their data and generate structured data to be published on the Web, requiring almost zero knowledge in Semantic Web concepts, RDF or ontology engineering.

WOODSTOCK '97 El Paso, Texas USA
© 2016 ACM. ISBN 123-4567-24-567/08/06...$15.00
DOI: 10.475/123_4

2. SEMANTIC ANNOTATION

One of the features that represent the Semantic Web approach is the availability of machine-readable data on the Web and one of the ways to tackle the issue of turning human-readable data into machine-readable is by using a process called semantic annotation. This process consists in generating metadata for documents, parts of documents or concepts, by creating labels for them enabling the use of resources like advanced search based on concepts, reasoning or data visualization based on ontologies. As described by [1][3], the annotation of some text may be considered semantic when are added to the text other information about its meaning or about the meaning of the elements in which it is composed of.

2.1 Semantic Annotation Categories

According to [5], the semantic annotation process is classified into three categories: Manual annotation is the process used to transform syntactic resources, like plain-text, into more complex structures which may have more information or knowledge, by adding metadata in some level of the document (word, phrase, paragraph). This manual approach requires user interaction in all steps, since text selection until the creation of metadata. Semi-automatic annotation requires, in some time of the process, the human intervention. Applications of this category may differ in its architecture, methods of information extraction and also on the performance to execute the job of annotating documents. The more complex category of annotation is the full automatic semantic annotation, which needs of methods and techniques from Artificial Intelligence field, for instance, deep learning and machine learning. Although the automatization of the process can save a lot of manual effort to do the annotations, it may not be reliable and annotated documents may need a human review.

2.2 Semantic annotation models

Besides the classification previously showed, there is a classification concerning the structure of the annotation, as presented by [1]. The most basic way of annotation uses tags, as Twitter does with #hashtags. In order to aggregate more information, there are more ways of annotating content, such as the key, value pair, used to describe properties of resources, for instance, a image with the key *resolution* and the corresponding value *1024x768 pixels*. In terms of structure, the annotation model using ontologies is the richest, it consists of the relation of a resource, or its parts, with some of its properties respecting a conceptual model

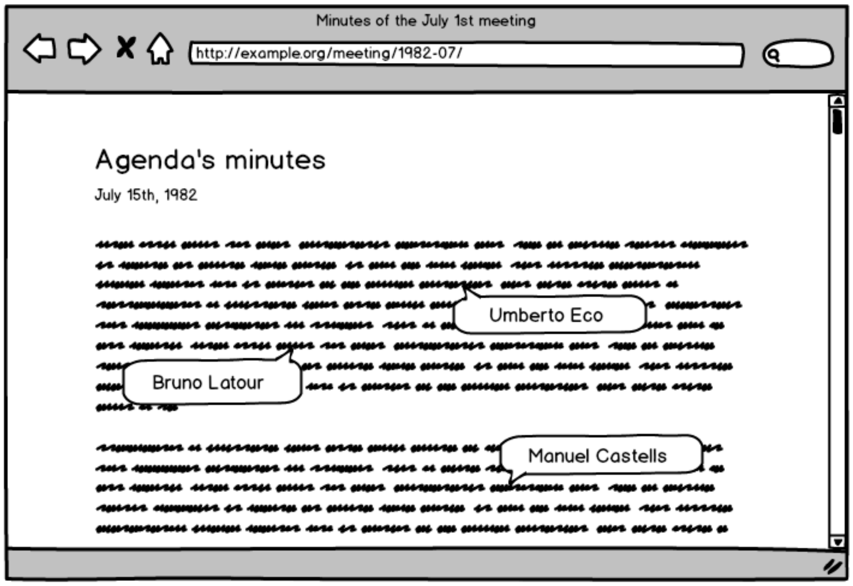

Figure 1: Illustration of annotated concepts within a document

or an ontology. This kind of annotation allows the user to link and to describe resources qualifying them as concepts' instances and defining relations, properties and constraints that eventually exists between resources.

3. PROPOSAL OF A SEMANTIC ANNOTATION TOOL

The tool proposed in this work is an application which has as main users people who publish content on the Web. The goal of this application is to enable more users to generate content with structured data in a way they can be machine-readable. The publication phase consists of users annotating content that will be published. Annotations are made by attributing values to properties of shared vocabularies in order to describe the subject is being annotated. At the end of this process, RDF triples are generated containing the descriptions of the resources. The RDF graph made from the annotation process is stored in a specific database, i.e. Open Link Virtuoso and then can be later used in the other phase.

In order to make the proposal clear, let's consider the following case as an example: There are periodical meetings happening every week at a specific company. All meetings are scribed for future reference or research, however, the generated minutes are written in natural language and are suitable for human reading. Suppose there is a need to collect all subjects discussed and all people who attended to the past meetings. A simple search on natural language data may not retrieve precise results, or a manual search for that can be an extremely difficult job, considering there are hundreds of minutes scribed.

Although there are different ways of doing the job described in the example, the suggested approach can benefit from Semantic Web technologies. Considering the agenda's meeting content is formed by structured data and it uses ontologies to describe the subject discussed in each meeting as well as to describe who were the people that have attended to. Every person who attended the meeting could have been identified with a vocabulary to describe people, for instance, *Friend of a Friend* (Foaf) vocabulary, and then the strings with characters that represents person's name, when annotated with Foaf vocabulary, could be interpreted as instances of Foaf, or better saying, instances of People with properties such as name, age, email. The hypothetical scenario described is one of the many cases in which Semantic Web technologies can be applied in order to obtain useful results, and the tool proposed aims to make possible applications like that. The proposed model for this tool is an application that works in a web-based WYSIWYG editor, in which the publisher can choose from a predefined list of vocabularies which ones to use to describe the terms and concepts of the content to be published. Ontologies and vocabularies available on the application are previously added via a different user interface. Each vocabulary to be used on the application has a specific color that will identify concepts annotated with that respective vocabulary. The user must choose both the vocabulary as well as the properties to describe data present in the text. When the user selects part of the content, a single word or a phrase, the tool shows a box with options and one of these options is an input field where the user defines the concept type to be annotated (type Person, Organization, Book, and so forth) according to the most appropriate vocabulary. After choosing the concept type for the piece of annotated content the application automatically generates a URI for that resource.

The first two RDF triples are generated by the application when the first step (concept and URI definition) is done.

```
<http://example.org/Person/Resource/>
    rdfs:type foaf:Person ;

<http://example.org/Person/Resource/>
    rdfs:label "Annotated content"
```

This resource could eventually be used or cited in other documents, that is why the importance of having a unique identifier for the resource.

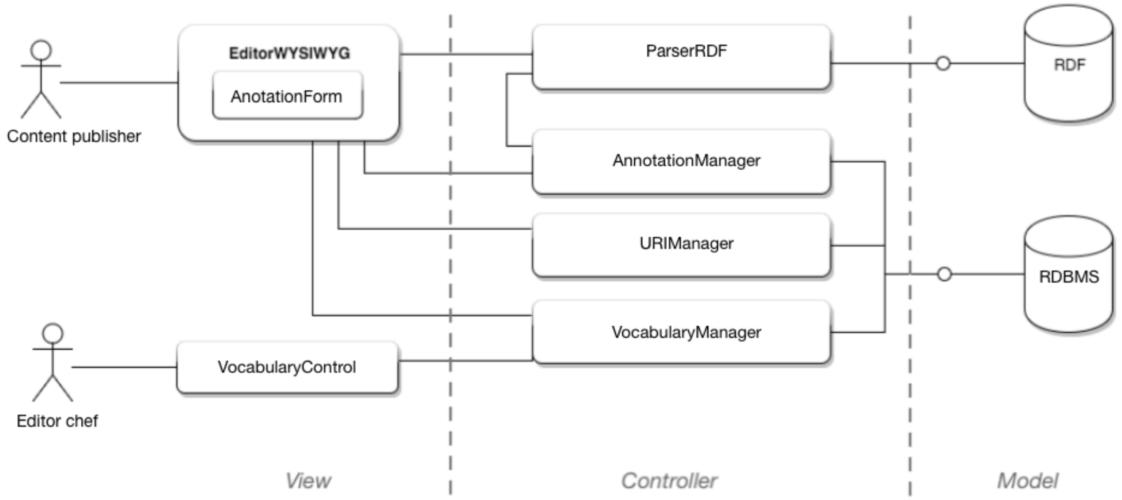

Figure 2: Architectural scheme of the proposed tool

In the same box that has been shown user can use more predicates to define new values for the annotated resource, for instance, one can use the `foaf:mbox` property to include the person's mail address, and the application immediately generates a new triple:

```
<http://example.org/Person/Recurso/>
    foaf:mbox "email@example.org"
```

The user can choose how many properties are necessary to describe the resource.

All content generated by the application is stored in appropriate databases: a graph database for RDF data and a relational database for unstructured data. The application makes possible for end-users who write content for the Web to also publish structured data on the Web without the need to write any RDF statement.

3.1 Architecture

This section presents the technical and architectural aspects of the foundation application for the proposed tool.

Application's components are architectural sorted according to the MVC model, widely adopted by the software engineering community. At the *Model* layer of this architecture, there are the databases specifics for each kind of data generated by the application: a relational database for plain text or HTML content, and the graph database for RDF data. Other components, such as the URI manager and the Vocabulary manager, are organized at the *Controller* layer and the user interface components, such as the annotation form, are at the *View* layer.

The *URIManager* component is responsible for generating URI for annotated resources and ensures that each resource has a single unique identifier. Vocabulary insertion to the database is made by the *VocabularyManager* component, which is in charge of the actions of inserting, deleting, updating, and retrieving vocabularies and ontologies used in the tool. Each annotation should be uniquely identified, this identification allows to appoint an exclusive node on the RDF graph and also allows to retrieve the annotation and its linking nodes. Those tasks are performed by the *AnnotationManager* component and the application makes use of [4] to describe annotations.

The component responsible for converting between different data formats is the *ParserRDF* component, which receives input data from the annotation form (user interface) and from the *AnnotationManager* component and then needs to store that data into the RDF store.

4. CONCLUSIONS

The model of the proposed application is being implemented in a real scenario at the Brazilian Internet Steering Committee (CGI.br), an organization which, among others activities, coordinates the allocation of IP address, registration of the <.br> domain and leads the discussion about internet governance in Brazil. A clear demand for this semantic annotation tool came when it was needed to organize and collect all content, including meeting's minutes, produced by the organization about different subjects and with many stakeholders involved. An application of this kind will enable the semantic annotation on documents about specific matters, such as internet governance, privacy, and security published by the organization.

5. REFERENCES

[1] P. Andrews, I. Zaihrayeu, and J. Pane. A Classification of Semantic Annotation systems. *Semantic Web Journal*, page 27, 2011.

[2] D. Karger. The semantic web and end users: What's wrong and how to fix it. *Internet Computing, IEEE*, 18(6):64–70, Nov 2014.

[3] O. Rodríguez-Rocha, I. Vagliano, C. Figueroa, F. Cairo, G. Futia, C. Licciardi, M. Marengo, and F. Morando. Semantic annotation and classification in practice. *IT PROFESSIONAL*, 17(IT-Ena):33–39, 2015.

[4] R. Sanderson, P. Ciccarese, and B. Young. Web annotation data model, 2015.

[5] T. Slimani. Semantic annotation: The mainstay of semantic web. *CoRR*, abs/1312.4794, 2013.