The FLERSA Tool: Annotating Web Content

José L. Navarro-Galindo, José Samos
Department of Computer Languages and Systems, Universidad de Granada
C/ Periodista Daniel Saucedo Aranda s/n, 18071 Granada (Spain)
(+34) 958 240 576
{jlnavar, jsamos}@ugr.es

Abstract. In this paper, FLERSA (Flexible Range Semantic Annotation) is presented as a user-centered annotation tool for Web content. The tool has been developed over a CMS and allows both manual and automatic semantic annotations. The interest for developing the FLERSA tool for annotating Web content arises out of the need to illustrate how to convert the Web Portal/CMS infrastructure into its semantic equivalent, thus extending the benefits of the Semantic Web. Amongst its key features, the following are of particular note: definition of infrastructural annotations based on an ontology (inspired by the Annotea Framework); the evolutionary capacity of annotated Web documents in light of the changes made (using RDFa format); and the possibility of enriching annotations by means of ontologies used as taxonomies/vocabularies; and the automated annotation process based on previous annotations that are used as a training Corpus.

Keywords: semantic annotation, automation, manual, tool, RDFa, metadata, semantic web

1. Introduction

The WWW represented a revolution in the way in which information is accessed, and as such has directly affected modern society. Currently, the number of documents on the network can be counted in the hundreds of billions, with information from all manner of fields and with a wide variety of content: sounds, images, video, data, and so on. Unfortunately, as the Web has evolved, more attention has been paid to the visual presentation of the information than to its semantics. The disadvantages of this model are not apparent when a user accesses information, but rather when they try to perform more complex tasks involving semantics of that information. For example, a user can easily access information from different travel agencies’ websites, but will not be able to obtain “intelligent” results when searching the terms "travel to Amsterdam for the first fortnight of August.”

The emergence in 2001 of the Semantic Web concept marked an important stage in the Web’s evolution. As stated in [1] it was "an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation." This proposition has not yet been realized, and although many efforts have been made in this direction much remains to be done. One of the main issues to be resolved in order to progress towards the Semantic Web is how to convert existing and new Web content that can be understood by humans into semantically-enriched content that can be understood by machines. The semantic markup of Web documents is the first step towards adapting Web content to the Semantic Web. Semantic enrichment is made possible by tagging web content with metadata, which enables entities that are found in the content, and relations between them, to be described. Providing the information elements that currently make up the Web with a well-defined meaning would, among other things, improve its contextual search capabilities, increase interoperability between systems in ‘collaborative’ contexts and, when combined with Web services, ultimately compose applications automatically based on published Web services [2] [3]. However, most Web content remains unstructured because of the difficulty of getting quality markup.

The main contribution of this paper is to present FLERSA, an innovative tool for semantic annotation
of Web Content. It also illustrates the functionalities that offers and the benefits provided.

This article begins by introducing the general features of the FLERSA tool, continuing with a detailed study its design process: requirements, design architecture, development details and functionality. Finally the paper ends with conclusions and bibliographical references.

2. General features

This section presents the semantic annotation tool called FLERSA (FLExible Range Semantic Annotation), a tool for manual and automated annotation of Web content [4]. The term ‘Flexible Range’ indicates that annotations can be defined over different ranges of text within a Web page and multimedia objects within it, independently of its HTML tags. It illustrates the desirable aspects that are going to be studied in subsection 2.1 for Semantic Annotation Systems.

FLERSA is a markup tool designed to generate semantic annotations on web content once it has been created. The document author will be the user who makes the markup; other users will benefit from exploiting the knowledge associated with the documents.

The tool makes exclusive use of open standards XML, RDF, RDFa and W3C’s OWL in order to promote interoperability and extensibility.

The pOWL framework [5] provides FLERSA with multi-ontology support, which facilitates the creation of a knowledge base consisting of agreed vocabularies and taxonomies from which to carry out semantic annotations.

The main ontology of FLERSA is based on Annotea [6], the annotation structure proposed by the W3C. This ontology is used as a base structure for any semantic annotation on a Web document, so that an instance is created for each annotation made. The underlying ontology also allows the possibility of using alternative microformats when making an annotation. A deeper study of this is presented in section 4.1.

FLERSA has been developed in line with client-server architecture, allowing multiple users to simultaneously make annotations (user-centered) and, even more importantly, to work together and re-use smart documents.

In its implementation, as far as possible cross-browser compatibility has been sought. Not all browsers support the W3C DOM Range [7], thus cross-browser compatibility should ideally be ensured from the implementation side. Compatibility has been achieved with the most widespread browsers: Internet Explorer, Mozilla Firefox, Chrome and Opera.

This is a tool whose client-side has a weak coupling; that is, client-side implementation can be easily adapted to another system. Services provided by the server-side are integrated into the underlying Web infrastructure and their technical details are presented below.

FLERSA’s main feature is dual storage of semantic annotations; that is, the annotations are stored in the server-side database in RDF metadata definition language. Moreover, they are stored embedded in the same document where annotation is undertaken in RDFa language in a way that is completely transparent to the user. This feature combines the advantages of the centralized annotations storage model with those of the embedded model: inference of new knowledge from the database of annotations; availability of annotations self-contained within the document; free access to metadata of Web documents for indexers, search engines and other types of semantic services to improve searches; and finally the possibility of providing information on the internal structure of documents, and the relationship between them.
3. Designing the FLERSA tool

3.1. Requirements

Nowadays, a wide number of annotation tools for producing semantic tags are available. Some existing tools (such as Amaya [8] or Epiphany [9]) provide semantic markup of web pages but do not support automated semantic annotation. Nor do they support evolution of the annotated document as they are usually based on XPointer [10] technology. These tools are not well integrated in semantic-powered publication frameworks, so semantic information is not exploited properly.

Other tools support automated semantic annotation (such as DOSE [11], KIM platform [12], and Melita [13]), but such tools often take a platform-centered approach rather than a user-centered one. They also have complex installation procedures and some are becoming obsolete. A complete study of some of these tools can be found in [14].

The main motivation for developing a new tool was to try to take into account as many aspects of user-centered and semantic information retrieval as possible. From our point of view, some important features for a semantic annotation tool are:

- User-centered Web environment.
- Lightweight framework for common infrastructure. Semantic Web tools should be compatible with commonly used applications such as Web portals and CMS. In order for a tool to be successful, ease of installation and use is also required.
- Automated annotations which use learning techniques (NLP) to automatically annotate domain-specific information from large repositories.
- Ontology-based annotations. Establish links between ontology concepts and the text fragments referring to them.
- Merge traditional and ontology-based information retrieval.
- ‘Pay as you go’ approach for automated annotations, where the system starts with an elemental Corpus. As the Web content grows and manual/automated semantic annotations are validated by the knowledge worker, the Corpus is improved over time and increased overall effectiveness of the annotation system is achieved.

To the best of our knowledge, no tool providing all these features exists.

3.2. Architecture

The system architecture on which the tool has been developed consists of four layers: core layer, data management layer, semantic layer and Web layer (see Figure 2).

Figure 2. Architecture of the FLERSA tool.

In the core layer are the operating system and the network services that this provides and the Web or Application Server.

The data management layer is made up of system components responsible for both content storage of Web documents and also for the annotations on them, and the knowledge base consisting of the ontologies of the system.

The semantic server-side layer is where server application services are developed. All message traffic between Web clients requesting services and the programs that provide them take place at this level. In this layer the implementation of programs that serve the web interface is carried out. The programs implemented here make use of programming libraries and Application Programming Interface (APIs) that provide the underlying layers. Among the most frequently used functions provided by these APIs it is worth highlighting: the facility for storage and re-
trieval of information; facilities for working with visual objects in the front-end programming; and facilities for working with ontologies.

Finally there is the Web interface layer, located at the uppermost level of abstraction of the system architecture, where the user performs all interaction with the semantic annotation tool. At this level the contents of Web documents coexist with metadata and with Web technologies in charge of modifying Web documents in runtime to provide them with semantic annotations in the form of metadata, and also to achieve timely message handling, by using server-side services, to provide the functionality of the tool.

3.3. Development details

In this subsection the main details of FLERSA tool implementation from a technical standpoint will be studied briefly – from the system components used, to the strategies and utilities used during their encoding.

The Joomla CMS [16] is used as the underlying Web infrastructure. Joomla is a Web Content Management System that is fairly widespread on the Internet and that enables the fast creation of portals and the easy generation of multimedia content from an environment that abstracts from the web technologies involved. Joomla is an open source tool under GNU/GPL license that is encoded in a modular way and allows extensions with the necessary functionality to be included. In particular, a module for Joomla has been implemented that provides the Web annotation markup tool.

As a framework for working with ontologies, pOWL has been used. This is an open source framework for parsing, storage, query, manipulation, service and serialization of knowledge bases in a collaborative Web environment.

As regards the programming of the tool it is worth noting that diverse programming techniques have been used such as the MVC design pattern (Model-View-Controller) and AJAX technology (Javascript and Asynchronous XML). In server-side programming, the host language in which Joomla is encoded has had to be adapted-to, therefore it has been programmed in PHP using the APIs offered by Joomla and pOWL. On the client-side, Javascript programming language has been used as it is a language compatible with all current browsers. Where possible, AJAX technology has also been used, provided by the Mootools Javascript library [17], which speeds up interaction between the user and the Web via asynchronous message transfer in XML format between server and client.

A specific JavaScript library has been developed to solve the cross-browser compatibility problems of the W3C’s Range object model. It ensures that the tool is compatible with the most widely-used browsers such as Internet Explorer, Mozilla Firefox, Chrome and Opera.

When implementing the process of defining flexible range semantic annotation [18], use has been made of the previous Javascript library and of a CSS style sheet for visual delimitation of the Ranges being defined.

RDFa generation is automatically carried out as semantic annotation is being marked-up. The automation is programmed in JavaScript and is responsible for adding SPAN tags with metadata about those texts on which the semantic annotation is performed. The metadata that are currently being saved are: author, context where noted (URI), creation date, text referred-to, granularity of the text (letter, word, sentence, paragraph, or free text) and annotation type (text or image).

4. Front-end functionality.

In the context of a CMS, the term ‘front-end’ refers to the specific CMS area specially designed to be accessible for users and web browsers, where access to the system functionality is achieved.

The tool is available at the Web address http://www.scms.es/joomla. It has a test user (user and password: demo) from which to make annotations to documents hosted on the site. On the website there is also a video with examples of the tool in use.

Figure 3. User main menu.

Figure 3 shows the initial menu which appears when a user logs in to the FLERSA web site. The options available are:
Submit an Article: From this option users can create new Web content considered as Joomla CMS system articles.

FLERSA Annotation Tool: This is the main option. From here, users can access only their own articles. Once an article is selected, the annotation toolbar panel (see Figure 4) appears. It provides facilities for working with semantic annotations associated with the selected article. A detailed study of each type of functionality is provided in the sub-sections of this article that follow.

Figure 4. FLERSA toolbar panel.

Semantic Search: From this option users can make queries about the articles, benefiting from the advantages offered by metadata from the semantic annotations.

Logout: To close the user session.

4.1. Creation of manual annotations

The FLERSA tool makes it possible to create new manual semantic annotations. The objective of the annotation process is to associate metadata with a fragment of text or with a picture. For more detailed information please refer to the paper concerned [18].

The process is straightforward: first a fragment of text or a picture has to be selected using the mouse, and the ‘New’ button of the FLERSA toolbar panel clicked. The toolbar panel will turn into an annotation editor as shown in Figure 5. At this point, a unique identifier is assigned to the fragment of text or the picture selected. The identifier is always used as the ‘subject’ of the RDF statements (subject-predicate-object triplets) that determines the semantic information of the annotation. Each annotation is considered as an instance of an infrastructural ontology, so an RDF statement is needed for describing each of its properties. The default properties (also called predicates) that make up a basic annotation are:

- **Annotates**: As in Annotea, it associates an annotation with the Web page (resource) on which the annotation is made.
- **Author**: The name of the user responsible for the creation of the annotation.
- **Body**: The text fragment of the Web page that is noted. The full text is stored to facilitate future search tasks. This field is ignored if the noted object is multimedia.
- **Context**: The context of the Web page specified in the ‘Annotated’ property. This corresponds to the URI that defines the position where the text or multimedia object is noted. For example http://w3.ex.org/p.htm#6543
- **Created**: As in Annotea, this refers to the date and time of creation of the annotation.
- **Modified**: As in Annotea, date and time of last modification of the annotation.
- **Granularity**: Indicates the type of granularity of the annotation. The types of granularity are: phrase, paragraph, character, word, or freetext. This property is not used when noting multimedia objects.
- **Section**: Indicates the section within the Web page where the annotation is made. The section types are: Text and Image. As the initial concept behind FLERSA ontology was to use it for the annotation of papers in Web format, the taxonomy of types of texts consists of: abstract, section, evaluation, references, discussion, future-work and introduction.

![Figure 5. Inspection window.](image-url)
These properties are automatically generated by the FLERSA tool according to the fragment of text or image that is being annotated. It is also possible to manually add the following properties belonging to the infrastructural ontology, which are of particular interest:

- **Related**: Establishes a relationship between the annotation and reference ontologies that are used as taxonomies. It is used to associate an annotation with the concept which is discussed in it. It is one of the most important properties because it is used for specifying the concept that the annotation deals with, and at a later date it will be very useful for making ‘intelligent’ queries within the annotations database.

- **Type**: Indicates the type of annotation that has been made. The types of annotation are: example, advice, change, seealso, explanation, question and comment. These types have been inherited from the Annotea framework.

Returning to Figure 5, it is worth noting that RDF statements relating to an annotation can be individually consulted via the inspection window by using the left and right arrow buttons. The ‘Add’ button is used for adding new properties for the current annotation. The ‘Finish’ button is used for concluding the current annotation. The ‘Type’ label indicates ‘m’ if the annotation is manual, or ‘a’ when it is automated. The ‘Validated’ label is only used for automated annotation, and therefore will be studied later in this paper.

When adding a new property into an annotation, the inspection window outlined in Figure 6 turns into the addition window shown above in Figure 7.

The FLERSA tool is able to manage the different vocabularies (microformats) that will appear in the ‘predicate’ field. The test environment of the FLERSA tool is preconfigured for using FOAF, RDF(S) and Dublin Core, as well as the infrastructural vocabulary (called ‘scms’) for defining properties of annotations. According to the RDF standard, a URI or literal can only be used as the value of the ‘object’ field in an RDF statement. URIs can be entered manually or through the window called ‘Ontology Selector’, which is shown in Figure 7. The ontologies available shown in the window are configured by the system administrator, as studied in Section 5.

The main function of the ontology selector window is to show the available ontologies offered by the FLERSA tool which can be used in RDF statements. Below, Figure 8 shows an example of an ontology about cars, in which classes of cars are used as a taxonomy of concepts. When adding RDF statements to an annotation, the concepts are used in the ‘object’ field for establishing relations that describe what subject is to be covered in the associated text fragment or picture (the ‘Related’ property).
4.2. Edition of annotations.

The FLERSA tool allows preexisting annotations to be easily edited. The steps are: click on the ‘Edit’ button of the toolbar panel of Figure 4 and then click on an existing annotation. As can be seen, the mouse icon changes from an arrow to a cross. Finally the toolbar panel turns into the ‘Inspection window’ of Figure 5 and shows the properties of the selected annotation. It also offers the possibility of adding new RDF statements or deleting any of those which were manually created previously.

4.3. Deletion of annotations.

The tool also supports the deletion of preexisting annotations. The steps are: click on the ‘Delete’ button of the toolbar panel and then click on an existing annotation. The coloured delimitation of the annotation duly disappears.

4.4. Clearing all annotations.

This function is useful when deletion of each of the annotations in a page is required. It is achieved by clicking on the ‘Reset’ button of the toolbar panel.

4.5. Permanent storage of annotations.

By clicking on the ‘Save’ button of the toolbar panel, all semantic annotations of a page are saved.

As highlighted in Section 2, the annotations are stored in the server-side database in RDF metadata definition language. When a user clicks on the ‘Save’ button, annotations are coded in XML format and they are sent to the server using AJAX technology.

Moreover, they are stored embedded in the same document where annotation is undertaken in RDFa language in a way that is completely transparent to the user. JavaScript technology is used for embedding automatically the RDFa code of annotations within the document.

4.6. Creation of global annotations.

Sometimes annotations are needed where their scope extends to whole pages. Users may be interested in determining through a semantic annotation the author of a Web page, or in specifying the concept that the page deals with.

The ‘Global’ button of the FLERSA toolbar panel performs this functionality. When clicking on it, the ‘Inspection window’ appears showing information related to the current whole page. It also offers the possibility of adding/deleting RDF statements.

4.7. Visualization of RDF.

W3C has a service for identifying and listing RDF from a web page called RDFa Distiller [19]. When clicking on the ‘W3C’s RDFa Distiller’ button of the toolbar panel, this service is triggered and a window showing the RDF that corresponds to semantic annotations of the current web page is shown.


The automated creation of semantic annotations is the key facet of the FLERSA tool. Thanks to this feature, knowledge workers are able to train a system to automatically establish relations between semantic annotations within a document and concepts provided by the taxonomies of the Knowledge Base.

A sound infrastructure is needed for undertaking this task. In FLERSA, a hybrid approach has been developed that combines two well-known natural language processing (NLP) techniques: the Vector-Space Model and n-grams. The theoretical basis behind these is complex, and in-depth discussion is beyond the scope of this paper. For detailed information please refer to the paper concerned [20].

When annotating a Web document in automatic mode, the FLERSA tool is able to work at both global and local level.

When the ‘Local Auto-Annotation’ button is clicked in the toolbar panel, the tool works at local level. The web document is split into text fragments
at paragraph level and a mark-up of each paragraph is made. Then a categorization process is undertaken for each fragment. The result of it is saved as a semantic annotation, in which categorization information is included (the ‘Related’ property). The statements generated can be consulted, as usual, from the inspection window provided by the ‘Edit’ option of the toolbar panel.

Our test environment for local automated annotations is made up of specialized information about several issues. Six features of the different cars are considered, namely: styling, interior, engine, driving, costs and safety.

When the ‘Global Auto-Annotation’ button is clicked in the toolbar panel, the tool works at global level. The entire text of a Web document is considered for the categorization process. A semantic annotation for the whole text is created, similar to the ones described in subsection 4.6, in which categorization information is included (the ‘Related’ property). The statements generated can be consulted, once again, from the inspection window that provides the ‘Edit’ option of the toolbar panel. As in the previous case, the statement generated is accessible from the inspector window in the ‘Edit’ option of the toolbar panel.

Our test environment for global automated annotations is made up of simple information about brands and models of cars. It is used for determining the brand that is discussed in a Web document. There are six brands of vehicle under consideration: Audi, BMW, Citroen, Fiat, Ford and Honda. These brands are also modelled by a domain ontology used as concept taxonomy.

4.9. **Semantic-powered search tool.**

When clicking on the ‘Semantic Search’ link of the user menu (Figure 4), an advanced search tool appears like the one shown above in Figure 10.

The function of the semantic search tool is to exploit semantic information from articles in order to achieve efficient results for queries. So through a query it is possible to filter articles on the basis of any property of semantic annotations defined in them: page of annotation, author, body of annotation, context, date of creation, granularity, date of modification, concepts that the annotation deals with, section and type.

The most useful property for querying is ‘Related’. This enables users to undertake searches throughout the articles database, filtering out the ones that do not deal with the issue specified in the query. Moreover, the tool makes inferences about the taxonomies of concepts. For example, a user can undertake semantic annotation on several articles that deal with cars to specify the model and brand of each; later, the user can make a query searching articles which deal with general concepts such as ‘car’ or ‘Audi’ and achieve specific results such as pages that deal with ‘BMW’ or ‘A4’.

5. **Back-end functionality**

In the CMS context, the term ‘back-end’ refers to the specific CMS area specially designed for administrators and content partners where access to configuration tools is gained.

The FLERSA tool has the administrator module as shown in Figure 11. It was developed as an administrator module of Joomla CMS and provides the following functionality:

− **Selection of concepts.** The ‘Reference Ontologies’ tab is used for selecting which ontologies will appear in the ontology selector window of Figure 7. The classes from the ones selected can be used as concepts when adding RDF statements in a semantic annotation. The configuration window for reference ontologies is shown in Figure 11; it can be seen how ‘wine’ and ‘car’ are selected as reference ontologies.

− **Selection of vocabularies.** The ‘Microformats’ tab is used for selecting which ontology properties can participate in the semantic annotation process. The properties from the ones selected can be used as predicates when adding RDF statements in a semantic annotation. The configuration window for microformats is shown in
Selection of concepts which participate in the automated annotation process. The ‘Automated models’ tab is used for this task. The automated annotation process only works with concepts belonging to the selected ontologies here, as the system needs to undergo a training process. Concepts from selected ontologies are taken and complex computations related to them are made. These calculations will be used later when NLP techniques are applied. As the theoretical basis for this is complex, for more detailed information please refer to the paper concerned [20].

Regarding ontology management, development of modules was not required as the pOWL framework provides this – as can be seen in Figure 13.

The pOWL framework has capabilities for importing and exporting ontologies in RDF/OWL format of arbitrary size. It also supports ontology edition, inference, privileges for users and groups, RDF storage, and retrieval RDQL query language.

This section explains all the steps needed to make a standard manual annotation. The steps are:

- Access to the website available at the Web address http://www.scms.es/joomla
- Log into the system using user and password ‘demo’. A user menu will appear on the right.
- Click on ‘FLERSA Annotation Tool’ for access to the article selection form.
- Click on any article title for access to its content. The content of an article will appear, along with a toolbar panel window called ‘FLERSA Tool’ as shown in Figure 4.
- Mark up a text fragment of the article via mouse selection. Then click the ‘New’ button. This will bring up an annotation like the one shown in Figure 14.
- For the moment, a standard annotation has been done. The annotation is made up by the standard properties studied in subsection 4.1: annotates, author, context, created, type, body, granularity and section.
- Click on the ‘Add’ button to enrich the annotation with new statements. A window like the one shown in Figure 6 will appear.
− Click on the drop-down list of predicate fields. Select ‘scms’ vocabulary and the ‘Related’ property. This property is used to explain the concepts that the content of annotation deals with.
− Click on the drop-down list of the ontology selector window. Select the ‘auto’ item and the taxonomy will be shown as in Figure 15.
− Select the ‘Interior_and_practicality’ concept of the ontology window using the mouse. The concept will be automatically transferred to the object field of the annotation window. Select it in this window.
− Finally, click on the ‘Save’ button to add the statement.

![Figure 15. Adding a statement that uses “Related” property.](image)

7. Conclusions and Future Work

In this paper, a semantic annotation tool known as FLERSA has been presented (available at the Web address [http://www.scms.es/joomla](http://www.scms.es/joomla)), in which all the desirable features of an annotation tool studied in subsection 3.1 have been developed. These features are: user-centered, lightweight, manual and automated annotation, ‘pay as you go’ approach, avoids the ‘Deep Web’ problem for annotations in documents, configurable ontology-based annotations, adaptable to another CMS system and full information retrieval.

We are currently still working on FLERSA, refining the automated annotation process. As regards future work, the next step will be experimentation – applying the tool to specific domains such as educational or medical environments. Work will continue along these lines, with the aim of improving automation of the annotation process and conversion of the infrastructure of specific Web Portals/CMSs into their semantic equivalent, thus extending the benefits of the Semantic Web.

References