



**Okkam – Enabling a Web Of Entities**  
**Grant Agreement No. 215032**

---

## **D10.1: Mockup “Entity-aware Content Authoring”**

<b>Document Number</b>	D10.1
<b>Document Title</b>	“Entity-aware Content Authoring” First Prototype
<b>Version</b>	1.0
<b>Status</b>	Final
<b>Work Package</b>	WP10
<b>Deliverable Type</b>	Demonstrator
<b>Contractual Date of Delivery</b>	31/12/2008 (M12)
<b>Actual Date of Delivery</b>	11/12/2009
<b>Responsible Unit</b>	ELS
<b>Contributors</b>	EXP, ANSA, UNITN, L3S
<b>Keyword List</b>	
<b>Dissemination level</b>	PU

## Change History

Version	Date	Status	Author (Company)	Description
0.1	12/01/2009	Draft	Stefano Bocconi (ELS)	First version
0.2	09/02/09	Draft	Stefano Bocconi (ELS)	Rework after comments
1.0	11/02/09	Final	Stefano Bocconi (ELS)	Final

## Executive Summary

This document reports the delivery of a prototype to each of the WP10 industrial partners, namely Elsevier and ANSA. The prototypes’ main functionality is to detect entities in news articles or scientific documents and associate them with their OKKAM ID. Due to the diversity of the domains (scientific and news publishing), the above-mentioned functionality is used to serve different user needs in different ways of usage, namely a more interactive one and a more automated one.

## Table of Contents

<b>1. INTRODUCTION</b>	<b>5</b>
<b>2. THE ENTITY-CENTRIC AUTHORIZING ENVIRONMENT</b>	<b>6</b>
2.1. NER AND SE	6
2.1.1. <i>Name Entity Recognition</i>	6
2.1.2. <i>Semantic Enrichment</i>	6
2.2. ONLINE AND BATCH MODES	6
2.3. STATUS OF DEVELOPMENT	7
<b>3. ELSEVIER USE CASE</b>	<b>8</b>
3.1. DESCRIPTION OF USE CASE	8
3.2. ARCHITECTURE OF THE USE CASE	8
3.3. STATUS OF DEVELOPMENT	10
<b>4. ANSA USE CASE</b>	<b>11</b>
4.1. DESCRIPTION OF USE CASES	11
4.2. ARCHITECTURE OF THE USE CASES	11
4.3. STATUS OF DEVELOPMENT	12
<b>5. CONCLUSIONS</b>	<b>13</b>
<b>APPENDIX I</b>	<b>14</b>
<b>APPENDIX II</b>	<b>15</b>
<i>Installation</i>	15
<i>Settings</i>	16
<i>Troubleshooting</i>	16
<b>GLOSSARY</b>	<b>17</b>

# 1. Introduction

---

This document records the delivery of the prototype developed within WP10 during the first year of activity. The WP10 goal is the realization of an Entity-Centric Authoring Environment (ECAE), i.e. an authoring environment where entities form the starting point to retrieve and aggregate additional information related to the authored content.

After 12 months of experience in designing and building the ECAE, we have gained more insight into the goals and tasks that are described in the Description of Work. Therefore, in this report we first discuss some important distinctions and concepts that underpin the work of WP10 and their consequences on the direction of development. Then we explain how the software that was developed applies to the Use Cases of the two industrial partners (Elsevier and ANSA).

As a general comment, this first year’s efforts have been focused on implementing usable tools, even though from a contractual point of view we were scheduled to deliver just a mock-up. This has been done in order to get to an early experimentation phase with real users, and in fact both industrial partners have already begun testing the software that has been developed in the context of this WP. As a consequence of this, theoretical progress on more high-level issues (namely Semantic Enrichment, explained in the following) has received less priority. In the future we will concentrate on that topic also, hopefully supported and stimulated by the feedback provided by the test results.

This document is structured as follows: chapter 2 explains the concepts related to the ECAE, while chapters 3 and 4 describe the prototypes delivered to Elsevier and ANSA, respectively. Our conclusions are documented in chapter 5. The appendices provide additional background information.

## 2. The Entity-Centric Authoring Environment

---

An Entity-Centric Authoring Environment means different things to each of the WP10 industrial partners, since their business needs differ. In the following we introduce some concepts and distinctions to specify what the ECAE means in the WP10 context.

### 2.1. NER and SE

First of all, it is important to distinguish two tasks: entity recognition (in literature also called **NER**, i.e. Name Entity Recognition) and Semantic Enrichment (**SE**). The former deals with detecting entities in a text and associating them with their (OKKAM) ID, while the latter involves retrieval, elaboration and presentation of the information related to the detected entities. In the OKKAM project, if NER uses OKKAM IDs, then it is called **Okkamization**.

#### 2.1.1. Name Entity Recognition

In general, NER consists of an Entity Detection phase which parses the text to determine the entities, and an Entity Matching phase, which associates a detected entity with its identifier. In WP10 Okkamization tools are developed by Expert System and they aim at the detection of:

- proteins (Elsevier Use Case)
- people, organizations, locations and events (ANSA Use Case)

While Entity Detection is a stand-alone functionality completely developed by Expert System, for Entity Matching (to provide entities with their IDs) the software queries the OKKAM core to retrieve the OKKAM ID. These tools are operational and are being fine-tuned. We will discuss them while describing the Use Cases.

Entity Detection can also be performed by other existing tools, which for biology, in particular, are freely available as Web Services. Interfaces are being built to connect to these services.

#### 2.1.2. Semantic Enrichment

SE involves combining different sources of information about the same entity. The entity, or more accurately, its identifier, constitutes the link between information sources authored by different people at different moments in time and with different goals/points of view. To integrate different sources, we need to solve two problems:

- syntactic integration: the information has to be expressed according to the same formalism (e.g. RDF statements)
- semantic integration: the information has to be combined taking care that no inconsistencies, contradictions or clashes in point of view are generated

This part is under development, with some prototypes already built, but with a large potential for investigation and research results, as we will discuss in this document.

### 2.2. Online and batch modes

Being two separate efforts, NER and SE may, but do not need to, happen at the same time. Let's examine two cases. In the first case, a user is authoring an article. Generally speaking, there are no strict time-constraints on this endeavour, since it is quality that matters. NER can interactively detect

entities in the text and semantic enrichment can provide additional information. In turn, this additional information can stimulate the user to think/write about other issues, creating therefore a positive feedback loop.

In the second case, a user works under time constraints and has little time to use additional services. The text is produced with no NER and no SE. At a later stage, NER is applied, and based on its outcome, possibly even further down the document process line, SE is applied. The final result is visible to the reader of the processed document, but not to the author at authoring time (author and reader are two distinct roles even though they can be the same person).

This distinction allows two different operational modes: **online** and **batch**. In the former, author, NER and SE all “collaborate” to produce the document, while in the latter the process is more a pipeline where the document is first elaborated by an author, then analyzed by NER and finally enriched by SE.

These two operational modes represent two extremes when considering human intervention: online mode requires the most human effort because NER and SE results are examined by the author, corrected and acted upon. While batch mode can run unattended once the document has been produced, even though some human intervention can be foreseen to guarantee quality standards.

The above-mentioned operational modes are used in the Use Cases for both WP10 industrial partners, Elsevier (online) and ANSA (batch).

### 2.3. Status of development

In both Use Cases Okkamization tools developed by Expert Systems are at a more mature stage, while Semantic Enrichment has some early prototypes, some ideas on paper and many research challenges. According to our plans, SE will be the focus of future research and development activities.

## 3. Elsevier Use Case

---

### 3.1. Description of Use Case

The general business need that this Use Case addresses is to alleviate the curation effort for bioinformatics journals. Curators are expensive but they are needed in order to guarantee that the information contained in articles is correctly entered in relevant databases and therefore searchable. Shortening curation time, even by a small percentage, can result in a considerable saving of money and allow more articles to be curated.

The authoring tool for Elsevier is meant to support authors and curators in editing molecular science articles for Elsevier’s international journal FEBS Letters<sup>1</sup>. This journal has started an editorial experiment to increase the reach of online published articles by facilitating their readability and findability. The experiment consists in adding to each article an abstract, called the **Structured Digital Abstract**, that encodes in a well defined schema the information contained in the article, with links to further information. For example, in the case of protein interactions, the SDA looks like this (hyperlinks included):

[MINT-6168263](#):

[Gsg1](#) (uniprotkb:Q8R1W2), [TPAP](#) (uniprotkb:Q9WVP6) and [Calmegin](#) (uniprotkb:P52194) [colocalize](#) (MI:0403) by [cosedimentation](#) (MI:0027)

The first line describes where in the Molecular INteraction database (MINT<sup>2</sup>) the interaction is described. This information is added by the MINT curators and is not known when the article is written. The entities in the next lines are the three proteins interacting, indicated by their name and Universal Protein Resource code (UniProt<sup>3</sup> is the world's most comprehensive catalog of information on proteins), the type of interaction and the method used to observe that interaction, indicated by their names and the Molecular interaction ontology (MI<sup>4</sup>) code.

The goal of the ECAE tool is to facilitate the author and/or the curator in composing the SDA by detecting entities in the text, determining their uniquely defined ID and link them to an entry in the related database (UniProt for the moment). In the above example, that would imply helping to compose the second line. Referring to the distinction introduced in 2.2, this case uses online mode.

This functionality corresponds to the scenario S25 in D1.1.

### 3.2. Architecture of the Use Case

This functionality is implemented by the following architecture (see also Figure 1): on the client side, an **MS Word plugin** communicates with a Entity Detection web service for the text parsing

---

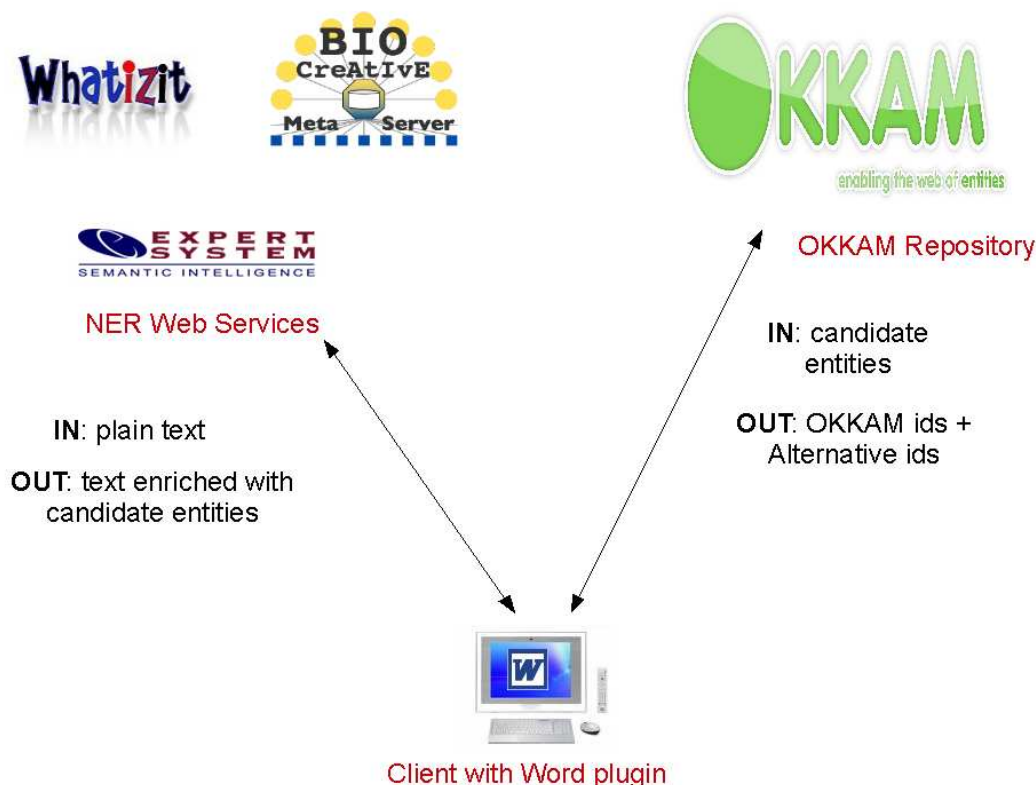
<sup>1</sup> [http://www.elsevier.com/wps/find/journaldescription.cws\\_home/506085/description#description](http://www.elsevier.com/wps/find/journaldescription.cws_home/506085/description#description)

<sup>2</sup> <http://mint.bio.uniroma2.it/mint/Welcome.do>

<sup>3</sup> <http://www.uniprot.org/>

<sup>4</sup> <http://www.ebi.ac.uk/ontology-lookup/browse.do?ontName=MI>

and name entity recognition. At the moment we use Expert System’s web services but we are also programming clients for Whatizit<sup>5</sup> and the BioCreative Metaserver<sup>6</sup>, both biological



**Figure 1 The architecture of the Use Case with the Word plugin**

NER systems that are freely available as web services. The web service returns the text with the detected name entities. The plugin then queries the **OKKAM repository** (providing also contextual information to facilitate the matching within the repository) to determine the OKKAM ID and possibly alternative IDs, in case the entity belongs to a domain where IDs are already established. In our case we have one alternative ID, namely the UniProt ID.

The plugin can include all of the found entities either in a section at the beginning of the article or in a separate file. The file format is either XML, or Comma Separated Value (CSV). In case of a section, the list includes the following information: name of the entity, its retrieved ID (the UniProt ID for this specific Use Case) and a link to the database where further information resides. Each occurrence of a particular entity in the text is cross-referenced to the entity in the list. The author/curator then checks the automatically composed abstract for correctness and inserts this information, together with other information, in the MINT database. The database generates the SDA which is sent to the FEBS Letters editorial office in Heidelberg as a Word document.

In Appendix II we include the instruction manual of the Word plugin with screenshots.

<sup>5</sup> <http://www.ebi.ac.uk/webservices/whatizit/info.jsf>

<sup>6</sup> <http://bcms.bioinfo.cnio.es/>

### 3.3. Status of development

List of open tasks:

- The Word plugin has been delivered to the FEBS Letters curators, and we are awaiting feedback.
- Development of clients for other NER services has started and needs integration with the rest of the platform.
- Referring to the distinctions introduced in section 2.2, the software performs Okkamization but no SE. We are investigating how to integrate different sources of information such as overlapping protein databases (such as RefSeq<sup>7</sup> and UniProt).

---

<sup>7</sup> <http://www.ncbi.nlm.nih.gov/RefSeq/>

## 4. ANSA Use Case

---

### 4.1. Description of Use Cases

The general business need that these Use Cases address is to make the news archive a more valuable asset by improving the findability of news and provide better links to related news. Once documents are okkamized, OKKAM supports searching for a particular entity with high precision and recall, and this allows retrieval of news about a particular entity as well as links to all news that contain a particular entity.

ANSA has two Use Cases: the first concerns a user of their news archive and the second a visitor to the news portal.

The scenario for the first Use Case is as follows: having acquired ANSA news archive, a user can search news by entities such as people, organization or places. As mentioned above, OKKAM’s role is to provide improved precision and recall when querying the news archive for a particular entity. In fact, searching with an ID avoids problems of synonyms and homonyms.

This corresponds to the scenario S12 in D1.1.

The second Use Case applies to the ANSA news portal and has the goal to provide the website visitors with information related to the article they are reading. This functionality can stimulate more traffic on the website and make visitors more willing to come back.

The scenario for this Use Case is as follows: a visitor of the news portal is reading a news article containing information about different entities such as people, organizations or places. To know more about a particular entity, the visitor clicks on it and (s)he is presented with a list of other news from the ANSA archive containing the same entity. This Use Case corresponds to the scenario S16 in D1.1.

In order to make this possible, the ANSA news archive needs to be okkamized, i.e. the news items need to be parsed and Okkamization applied, thereby associating news with the identifiers of the name entities that they contain.

### 4.2. Architecture of the Use Cases

Referring to the distinctions introduced in section 2.2, in this scenario OKKAM is used in batch mode<sup>8</sup>. The original news are encoded in NewsML, which is an XML news exchange format for general news. The news are fetched and parsed by the news okkamizer provided by Expert System. The name entities are extracted and the OKKAM repository is queried to retrieve their OKKAM IDs. These IDs are then inserted into the news by encoding them in the original NewsML file together with the detected entities (see also Figure 2).

---

<sup>8</sup> Some human intervention can be applied, for example when a journalist publishes a news on the portal, (s)he can request the okkamization of the text and check the results. This is possible if there are no strict time constraints (no real time news) and for a limited number of news (not for an archive)

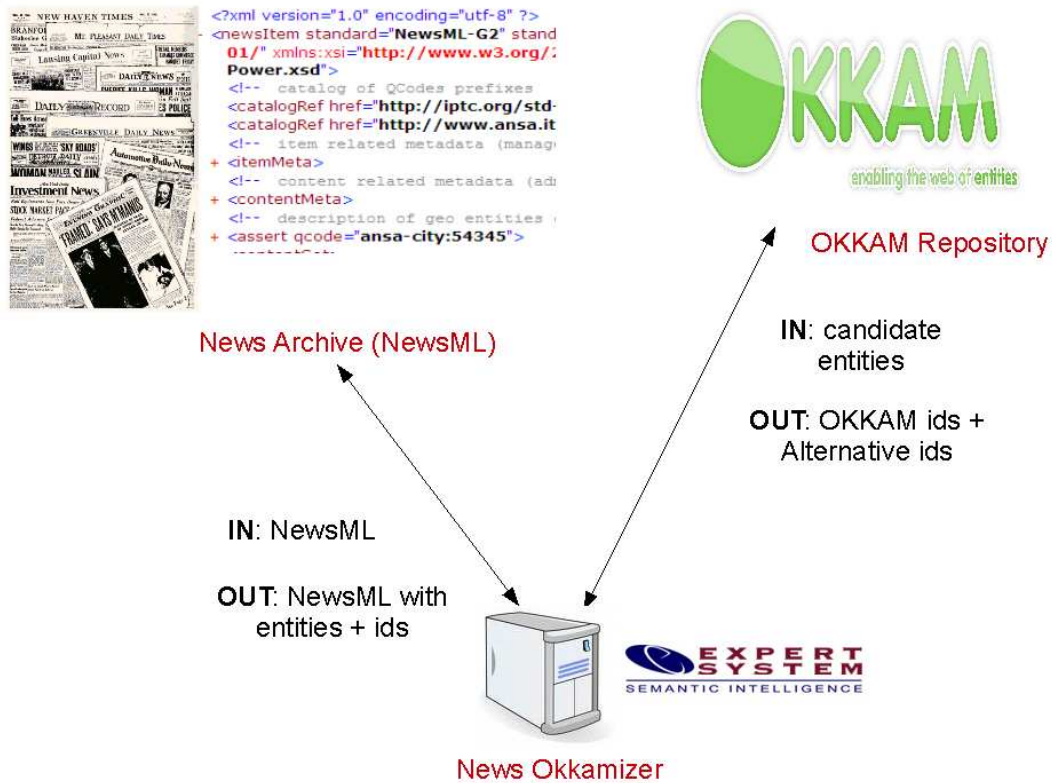


Figure 2 The architecture for the news okkamization

### 4.3. Status of development

List of open tasks:

- ANSA is testing the news okkamizer and has provided feedback to Expert System. Expert System is improving the performance of the tool according to that feedback.
- Different mechanisms to relate news are being studied. These combine entity information with, for example, event information or geolocation information.
- There are still large opportunities for SE, since in the domain of news there are several initiatives that could provide additional information to better link news together, such as DBpedia<sup>9</sup> and Freebase<sup>10</sup>.

<sup>9</sup> <http://dbpedia.org/>

<sup>10</sup> <http://www.freebase.com/>

## 5. Conclusions

---

As described in this document, two prototypes have been delivered to the two industrial partners, Elsevier and ANSA, who are testing their functionality. Though at an early stage, these tools are real and are not just mock-ups. Therefore we provided more than what was initially planned in the Description of Work. The results of the evaluation will drive further development. At the same time, we are investigating forms of entity-based Semantic Enrichment to provide aggregated information for the authors of scientific articles (Elsevier Use Case) and the users of the news archive/portal (ANSA Use Case).

As mentioned in earlier sections, both Use Cases are strongly driven by the business needs of each industrial partner. The domain of scientific publishing has different requirements from the domain of news, therefore the two Use Cases address interactive (online) as well as batch mode usage of the OKKAM infrastructure.

## Appendix I

---

A video showing the functionality of the MS Word plugin is available at:

<http://okkam.dit.unitn.it/okkam/wp10-video.mpg>

Be aware, it is 334 MB!

## Appendix II

The following is taken from the installation file delivered to the FEBS Letters curators.

### Installation

Download the zip file and uncompress it.

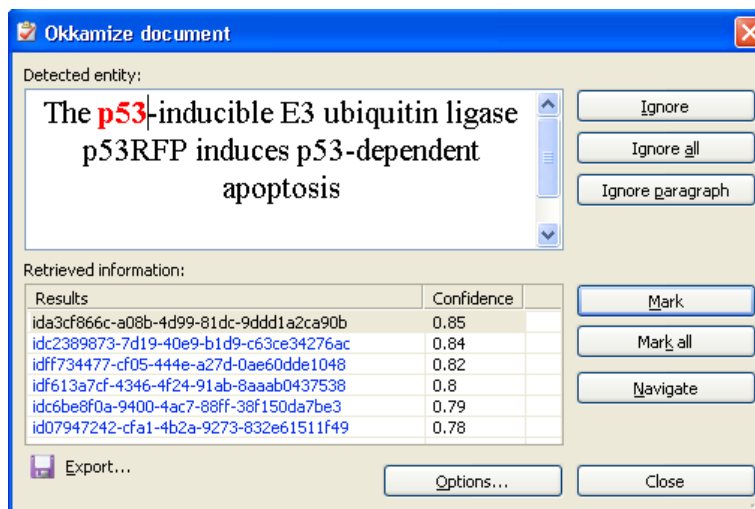
Run setup.exe. The installation program might ask to close several running programs (possibly all of them). It is safe to only close Word and the ones that use Word (for example Outlook).

Open Word, go to View->Toolbars and select Okkam Tools. The Okkam toolbar looks like this:



From left to right, this are the buttons you can click:

**Okkamize**: the all document is scanned and interactively annotated. The **Okkamize document window** pops up:



In the upper part entities are detected and highlighted, in the lower part their Okkam ID is retrieved. In case of more hits, to select the correct match the user can press the **Navigate** button and a browser will show the entity profile. The user can choose to:

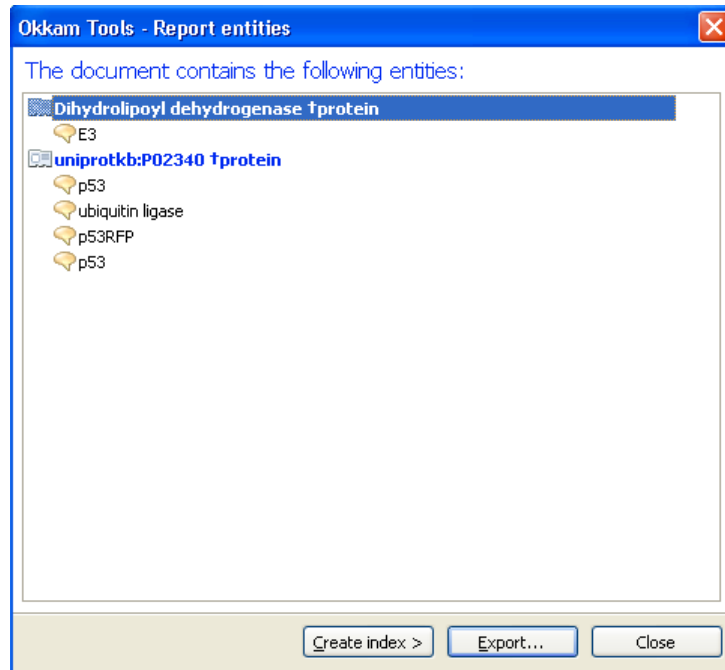
- **Ignore** this entity (and thus to not insert any annotation in the file)
- **Ignore all** occurrences of this entity
- **Ignore paragraph** to skip the whole paragraph
- **Mark** this occurrence
- **Mark all** occurrences of this entity
- **Navigate** described above

The Ignore and Mark all functionality work paragraph-wise, so as soon as the paragraph is being examined by the plugin (and not immediately for the whole document).

**Mark the current selection**: when an entity is not recognized, it can be marked selecting it and pressing this button.

**Remove Okkam’s markings from the current document**: removes all annotations

**Report Okkam’s information from the current document’s markings**: generate a report of the entities found. The **Report entities** window pops up:



Where all detected entities are shown. The user can:

- **Create index:** an index of all found entities is generated at the beginning or at the end of the document, and all markings can be optionally removed. All the occurrences of the found entities are linked to these entries in the index.
- **Export:** the entities are exported in either a Comma Separated Value file or an XML file.

## Settings

These settings are already the default, but we repeat them here.

Options to be set in the **Options** menu:

- Entities analysis: “Online webservice” selected, value <http://host.expertsystem.it/OkkamWebService/services/OkkamWebService>
- Okkamization: “Use the following webservice.....” value <http://api.okkam.org/okkam-core/services/WebServices?wsdl>
- do not select the option “Cache results in:”
- Clear (leave blank) the field “Indexed repository for okkamized entities:”

## Troubleshooting

In case of no entities being detected, try to select the cache option and to press **Clear**. Then deselect the cache.

## Glossary

---

<b>ECAE:</b>	Entity-Centric Authoring Environment.
<b>NER:</b>	Name Entity Recognition, composed of an Entity Detection phase which parses the text to determine the entities, and an Entity Matching phase which associates a detected entity with its identifier.
<b>Okkamization:</b>	NER done with OKKAM IDs.
<b>SE:</b>	Semantic Enrichment, which involves combining different sources of information about the same entity.
<b>RDF:</b>	Resource Description Framework, a W3C standard <sup>11</sup> .

---

<sup>11</sup> <http://www.w3.org/RDF/>