OPAALS PROJECT

Contract n° IST-034824

WP6: Socio-Constructivism & Language

Del 6.10 - Multilingual search strategies, search assistant prototypes and the way forward

Project funded by the European Community under the "Information Society Technology" Programme

Contract Number: IST-034824

Project Acronym: OPAALS

Deliverable N°: 6.10

Due date: M36 (May 2009)

Delivery Date: M38 (July 2009)

Short Description:

The Multilingual Search Assistant (MSA) is an online wizard which helps users empowering their queries: if somebody is interested in "apples" the tool allows identifying more precise names to look for, such as "apple juice", "apple wines" or "crab apples". The navigation of more specific or more generic items (keywords) is done through alphabetical lists or a graphical representation of those keywords showing their connections.

The MSA tool allows also to expand a query: for example, for somebody interested in "tuna", the tool allows to search also for synonyms such as "tunny" or "albacore", or search the same expression in other languages, such as "barrilete" or "Atún palomida" in Spanish or "Atum-voador" in Portuguese.

Moreover, the easy-to-use graphical interface of the tool, allows for easy construction of logical searches. For example, a user interested in rice in India or Pakistan, can just enter these three mentioned keywords and the tool present a graphical picture allowing for the selection of all the combinations of those keywords, for example "rice in India and rice in Pakistan" or "rice in India but not in Pakistan".

The tool is available at http://lprapp14.fao.org:9090/mssa/. This document described the system itself, how it can be installed and its functionalities.

Author: Soonho Kim (FAO), Jeetendra Singh (IITK), Prabhakar TV (IITK)

Partners contributed: Margherita Sini (FAO), Prabhakar TV (IITK)

Made available to: Public, 3 July 2009

Versioning

Version	Date	Name, organization
0.1	6 May 09	Soonho Kim (FAO) and Jeetendra Singh (IITK): first draft of technical document
0.2	11 June 09	Soonho Kim (FAO) and Jeetendra Singh (IITK): adding source e code description and useful links
0.5	3 July 09	Margherita Sini (FAO): draft revision
1.0	1 Aug 09	Prabhakar TV (IITK)

Quality check

Internal Reviewers: Ossi Nykänen , Paul Krause

Dependences:

Achievements*	The prototyped Multilingual Search Assistant has been developed. Multilingual searches are now possible. Further extensions of the tool are also possible.
Work Packages	**WP2** - Automatic Code Structure and Workflow Generation from Models: the prototypes aimed to be developed in this Workpackage could benefit from the idea of using the MSA technology for improving multilingual services and user-friendly information retrieval interfaces. **WP5** - Integration with the Digital Ecosystems Platform: as mentioned for WP2, D6.10 may provide to WP5 ideas for the improvement of management of many languages and the resources for doing so. **WP10** Sustainable Research Community Building in the Open Knowledge Space: D6.10 can provide infrastructure and the tools to improve community communication.
Partners	Fachhochschule Salzburg GmbH (Salzburg University of Applied Sciences); T6 Ecosystems srl; TechIDEAS Asesores Tecnologicos; Waterford Institute of Technology; The University of Surrey; Instituto Tecnologico de Aragon.
Domains	Computer Science: the deliverable objective is the development of a new tool. Semantic tools: the system goes in the direction of enriching software semantics capabilities. Human-computer interaction: the system aims to improve information and knowledge accessibility.
Targets	Information Technology specialists, SME
Publications*	
PhD Students*	None
Outstanding features*	Use of online web services connecting to a real terminological resource; exploitation of the semantics of the terminological resource to facilitate information discovery.
Disciplinary domains of authors*	Soonho Kim: Knowledge Management Jeetendra Singh: Information Technology Margherita Sini: Information Management Prabhakar TV: Professor at IIT Kanpur

The information marked with an asterisk (*) is provided in order to address Recommendation n. 4 from the Year 2 review report

Table of contents

Executive Summary

The Multilingual Search Assistant (MSA) aims to provide an easy to use platform for information retrieval in the agricultural domain. Its technology, detailed in this deliverable, however, can be applied to any other domain, with the only constraint that a terminological resource, in the form of a thesaurus or an ontology should exist.

The structure of the tool is modular, and therefore easily expandable. Section one of this deliverable describes the system in general, and section two describes the requirements that drove this development.

Section 3 and 4 goes more in detail giving the technical information for installing and running the system, and section 4 aims to provide details to make the code easy to understand for future expansion.

Last section, n. 5, gives some few additional information about the system, including where to see video and textual help files, test it, and read more about it.

1. *Introduction*

The Multilingual Search Assistant (MSA), also called Multilingual Semantic Search Assistant (MSSA), is a web-based tool helping users to find better information by using proper keywords in an FAO fulltext-based search engine. The main users of this tool are the agri-scientists, students, information service providers seekers accessing the agriculture information repositories which index a huge number of agriculture related publications from all over the world. However, the technology framework can be used for accessing any kind of information that requires a keyword based multi-lingual search interface.

The MSA implements four independent functions each associated with the AGROVOC multilingual thesaurus to assist users in choosing better keywords:

- **Build complex search** based on Boolean search;
- **Browse topics** through an animated browser;
- **Expand languages** by supporting cross-language search for five FAO official languages (English, French, Spanish, Arabic, Chinese);
- **Expand synonyms** by query expansion;

The multilingual search assistant (MSA) was designed as a plug-in for any kind of search engine powered by Apache Lucene1. Thus, implementation of the MSA focused on the modification of the query string itself, instead of modifying the target search engine directly. As illustrated in Figure 1, users start formulating their query in the MSA. Then, pressing the "submit" button, the system automatically takes users to the target search engine homepage which shows the results of the user's query. There is no need to change any code in the target search engine; the only thing to be done is add a link to the MSA on the target search engine homepage.

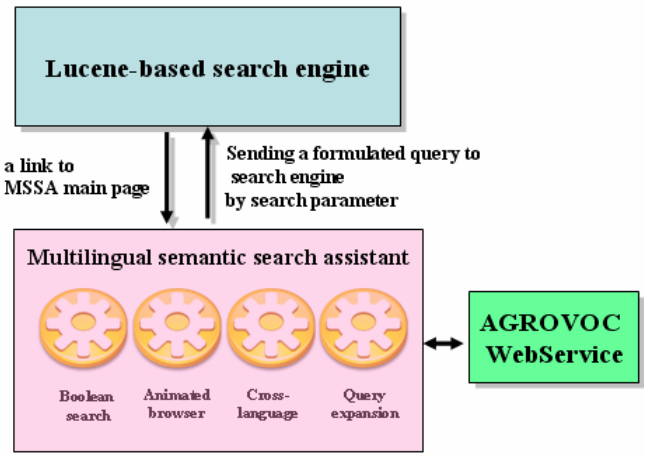

Figure 1. Communication between Lucene-based search engine and the multilingual search assistant system associated with the AGROVOC WebService.

[1] http://lucene.apache.org/java/docs/.

1.1. *Functions*

1.1.1. Build complex search

The function of "build complex search" was implemented with visual Boolean operator interface, which is intuitive and easy to use. We adapted a Venn diagram which consists of two or more intersecting circles representing relationships among given sets. It is frequently used to represent graphically the results of combining basic Boolean operators. However, most Boolean search interfaces employ a text-based or drop-down box (containing 5 Boolean operators) interface. For example, the advanced search in Google (Google 2008) supports text-based Boolean operators. The Venn diagram-based user interface illustrated in Figure 2 makes users feel more comfortable using Boolean operators because they do not have to directly specify Boolean operators. Instead, just clicking the area in which they are interested automatically creates keyword queries with Boolean operators.

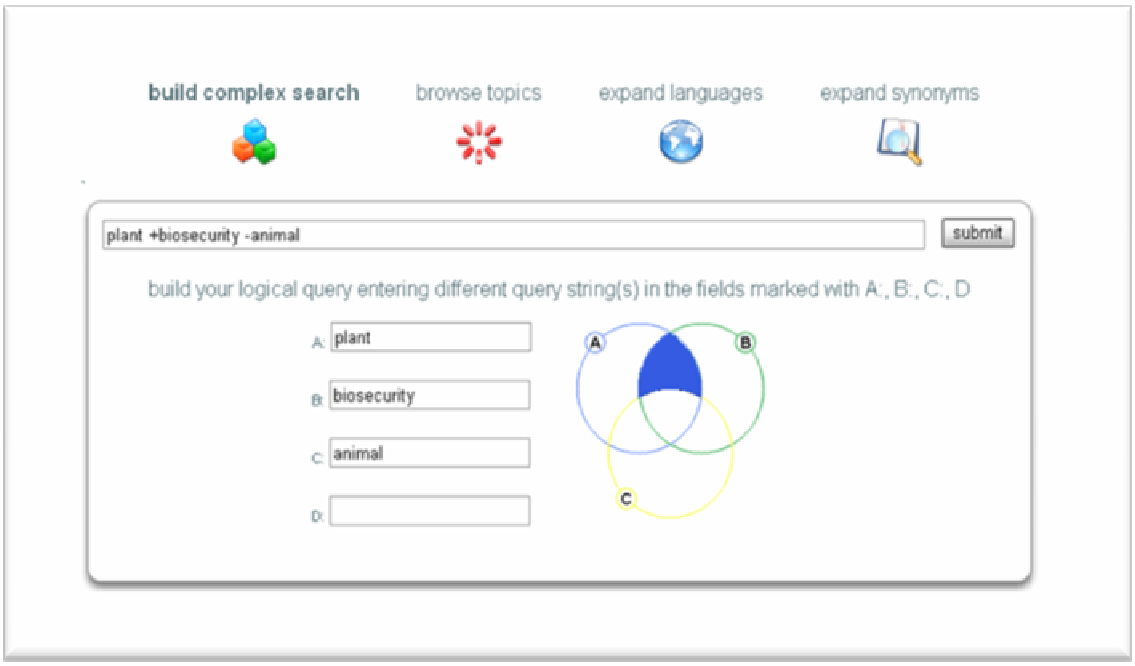

Figure 2. A screenshot of "build complex search" function

1.1.2. Browse topics

The animated AGROVOC concept browser is an intuitive and easy to use browsing interface for AGROVOC thesaurus and AGRIS search. The browser is a visualization tool for indexing and agriculture concepts in AGROVOC thesaurus and can be search agricultural related contents from a lucene-based search repository. The tool represents concepts in visual format, and presents them based on their hierarchical categories. The tool consists of both concepts as well as images. Images are used to elaborate and visualize the concepts. The tool uses the Prefuse (http://www.prefuse.org) visualization toolkit. The user starts the search by the typing the concept name. The AGROVOC thesaurus is searched to retrieve the neighborhood (all terms related to the search term) and is laid out graphically.

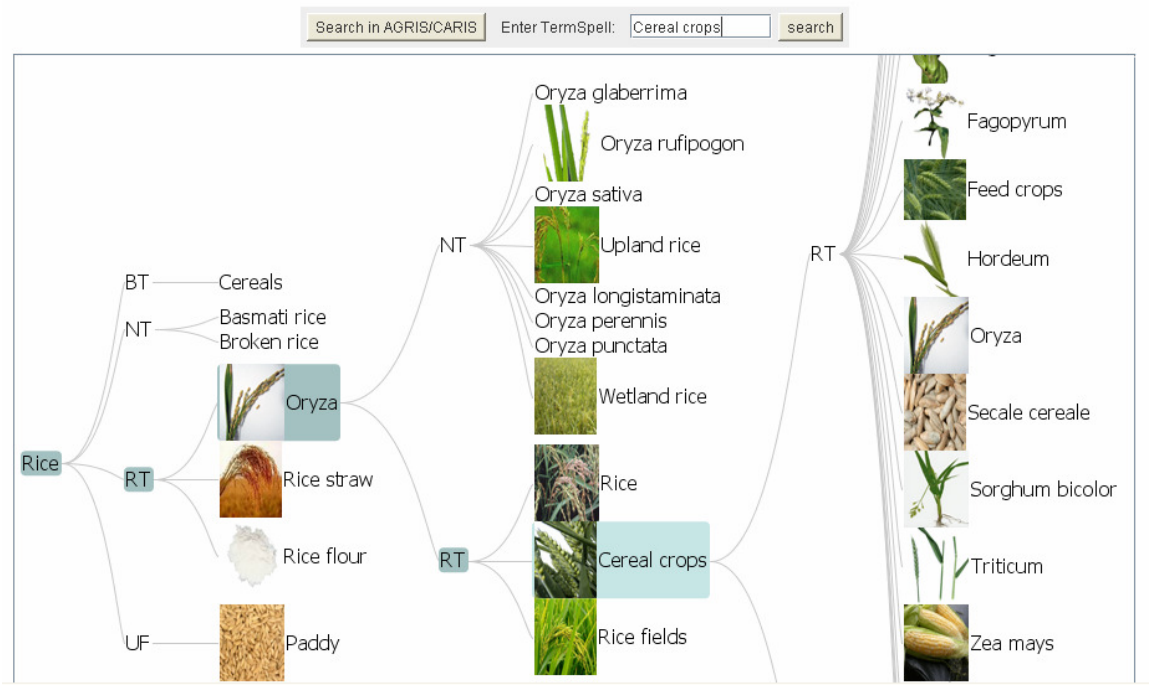

Figure 3. The thesaurus browsing area.

1.1.3. Expand languages

The MSA applies a thesaurus-based query translation approach in which keywords typed by the user are translated to selected target languages using the AGROVOC thesaurus. While a machine-based translation approach has the limitation of representing the sense of the original query, AGROVOC thesaurus-based query translation can be done in a straightforward fashion because every translation is already verified by domain experts. A disadvantage of this approach is that AGROVOC can not cover all domains everywhere in the world. However, this limitation might be overcome by including other thesauri covering different domains. The "Expand languages" function in the MSA is illustrated in Figure 4. Because AGROVOC contains 35,000 terms per language, it is important that the interface shows available concepts to users based on their interest before performing translation and query expansion. We called this the "auto-completion terms" functionality. When user selects a concept, the MSA automatically calls the AGROVOC Web Service to obtain translations into the five FAO official languages. Then users can add or delete languages using the checkboxes shown in Figure 4. Figure 5 shows available terms according to user input.

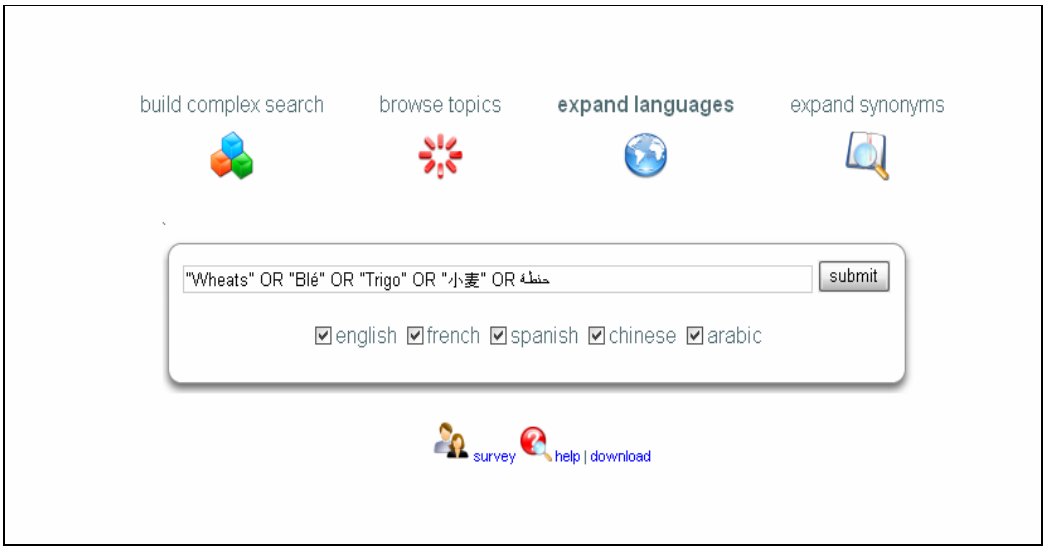

Figure 4. A screenshot of the function "expand languages"

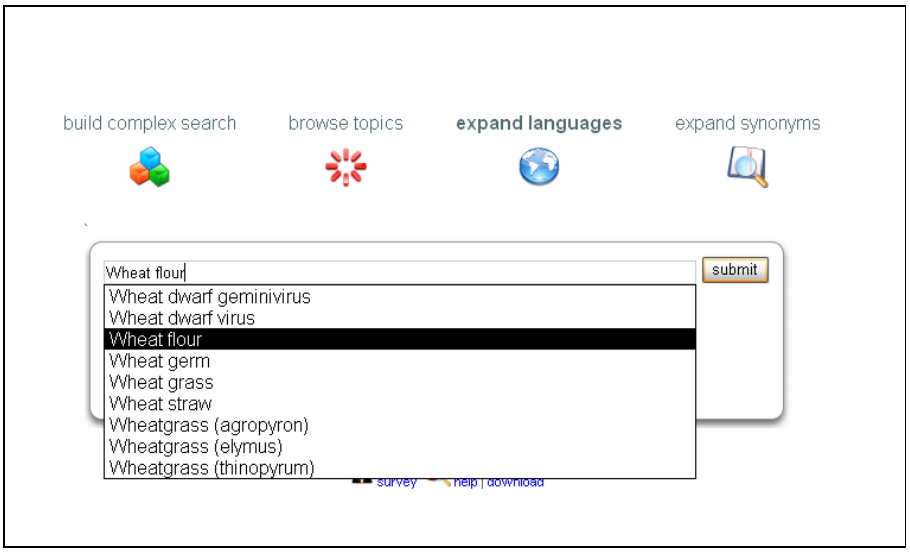

Figure 5. A screenshot showing "auto-completion terms" corresponding users' input.

1.1.4. Expand synonyms

Domain-specific knowledge is the most important resource for the query expansion. AGROVOC can again play a role, since it provides a variety of synonyms and acronyms and is already officially approved in the AGROVOC communities. So, the domain-specific knowledge for "expand synonym" was implemented using the AGROVOC Web Service. For example, the term "water balance" which a user selects is expanded by adding the AGROVOC synonyms "water budget", "water saturation" and "evaporate demand" shown in Figure 5.

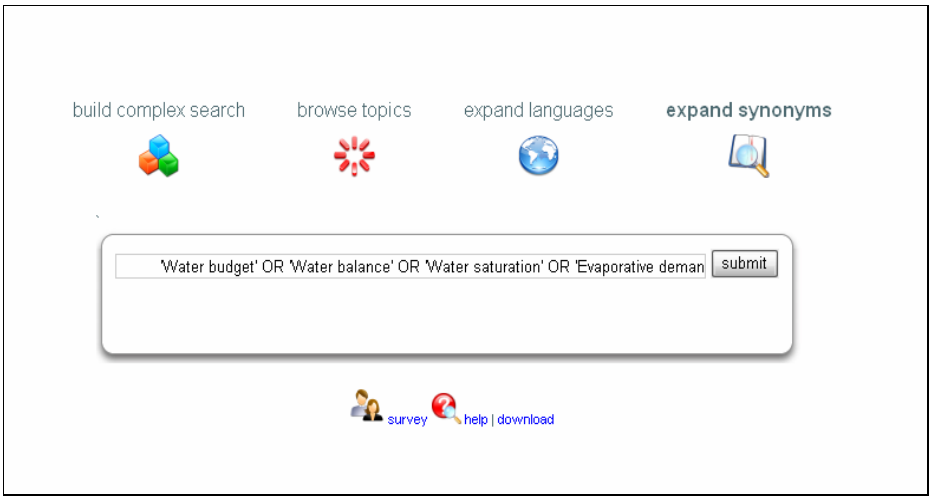

Figure 3. A screenshot of "expand synonyms" function

2. Requirement

The MSA is developed with Java web technology including JSP (JavaServer Pages), Java Web Services, and AJAX. This is the minimum requirement needed before installation of MSA 1.0 can be successful.

- Apache Tomcat 5.5 or upper versions: http://tomcat.apache.org/ for supporting JSP
- AGROVOC Web Services Version 2.0
- Prefuse tool kit for browsing interface (http://www.prefuse.org)

2.1 *Installation of MSA*

To install the MSA, simply unzip the whole archive (MSA.zip) into the Tomcat webapps directory. Then, under Tomcat webapps directory, a new folder called "MSA" is created, which is a root folder of all source codes of MSA application, shown in the left side of Figure 1. In the right side of Figure 1, there are several folders and files which are necessary to run MSA. Then the next step is to restart the Tomcat server. Note that it is not the final step of installation of MSA. Completed installation needs to another step (2.2 Installation of AGROVOC Web Services 2.0) for installing AGROVOC Web Services 2.0. So, some functions in the MSA might not be working properly in this stage.
If testing this MSA application in the local Tomcat server, the following URL might be working: http://localhost:8080/MSA/index.jsp (port number might be various based on Tomcat server setting). In this case, the Tomcat server was installed in the local machine. In other case, corresponding URL will be needed to test MSA. Figure 2 shows the front page of MSA when testing installation.

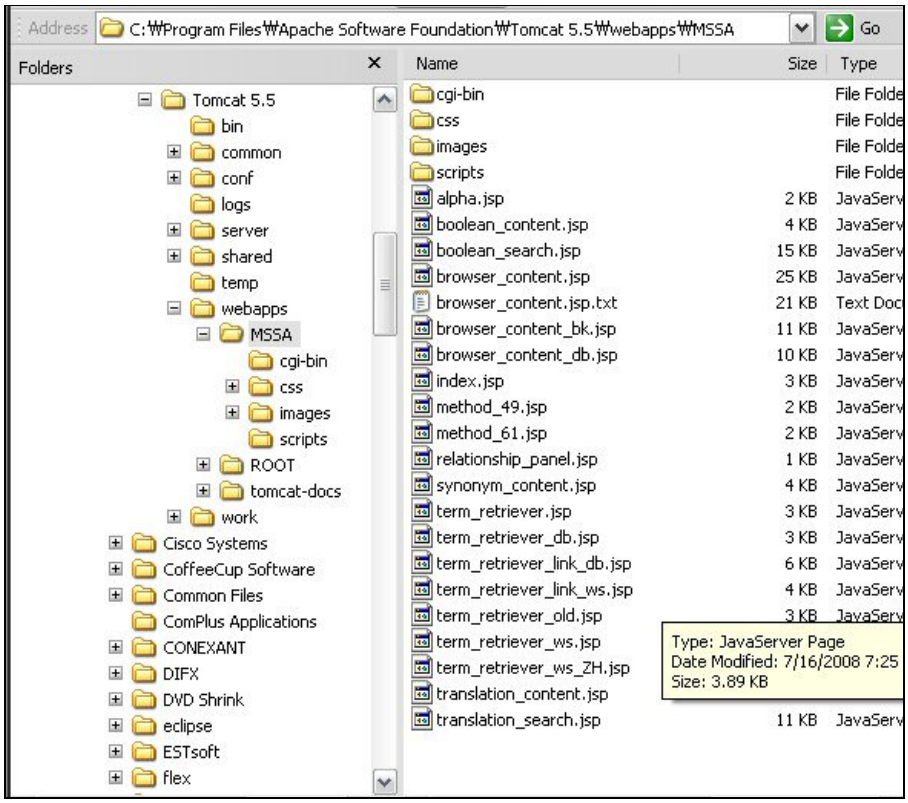

Figure 4. A folder structure of MSA application in a Tomcat server, after unzipping given archive file (MSA.zip).

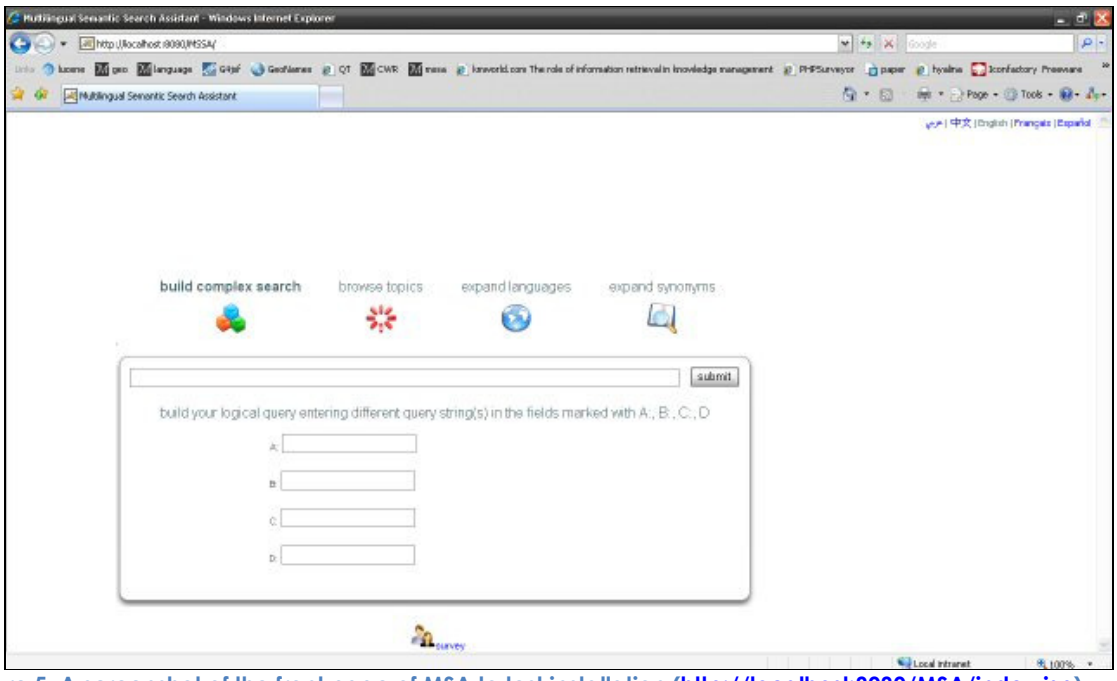

Figure 5. A screenshot of the front page of MSA to test installation (http://localhost:8080/MSA/index.jsp).

2.2 *Installation of AGROVOC Web Services 2.0*

Setting AGROVOC Web Services 2.0 in the Tomcat server.

The installation of AGROVOC Web Services 2.0 needs to copy necessary files into "WEB-INF" folder in your Tomcat server. Necessary files can be downloaded from the AIMS website (ftp://ftp.fao.org/gi/gil/gilws/aims/kos/webservices/agrovoc_ws_client.zip). When unzipping the "agrovoc_ws_client.zip" file, there are two folders: One is "SampleAgrovocWSProxy" and the other is "WEB-INF" folder.

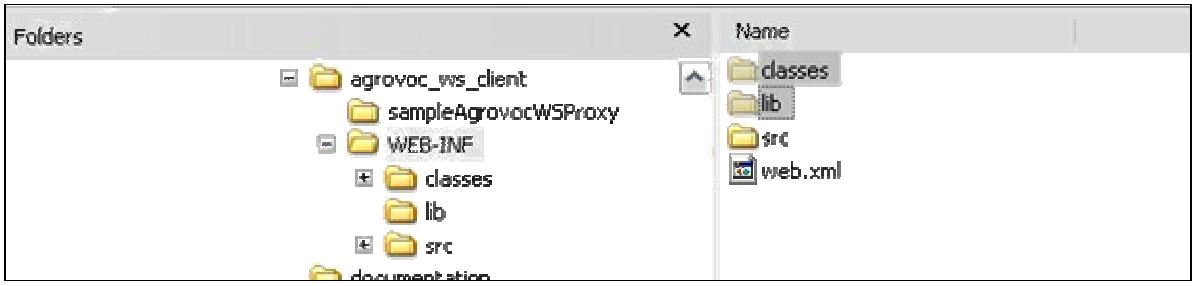

Figure 6. A folder structure when unzipping the file "agrovoc_ws_client.zip". When clicking "WEB-INF" folder (in the left) inside a root folder of agrovoc_ws_client folder, then there are three folders (classes, lib, and src) and one file (web.xml)

When opening "WEB-INF" folder, there are three folders (classes, lib, and src) and one file (web.xml). All files in the classes folder need to be copied to the corresponding classes folder inside Tomcat server WEB-INF directory (If there is no "WEB-INF" directory in Tomcat server, you can create it by yourself under a root directory of your application. In the same way, you can create classes and lib directory as well). All files in the lib folder need to be copied to the corresponding lib folder inside Tomcat server WEB-INF directory. Then, AGROVOC Web Services 2.0 is ready to run in the server side.

Calling AGROVOC Web Services 2.0 from the MSA

Calling AGROVOC Web Services 2.0 from MSA is already settled down in the source codes of MSA. There is no necessary installation for it. Basically, one file of MSA source codes called "translation_search.jsp" contains all works for using AGROVOC Web Services 2.0.

2.3 *Graphic Browser*

Installation of visual browser is required prefuse tool kit (a prefuse folder) that is inside the MSA folder and a file agrovoc.html and a myapplet folder that is also in the MSA folder. These three things (agrovoc.html, myapplet folder, prefuse folder) are necessary for visual browser. JRE 1.6 is required for visualization. Visual browser is also using web services so web service must be active.

2.4 *Final Checkpoints*

Section 2.1 and 2.2 describes how to install MSA and AGROVOC Web Services. In this section, final checkpoints give a final confirmation of installation of MSA. Shown in Figure 4, the proper installation of MSA contains five folders (cgi-bin, css, images, scripts, and WEB-INF). WEB-INF folder was created during installing AGROVOC Web Services 2.0 described section 2.2. Other folders were created during installing MSA explained in section 2.1.

The following is a list of final checkpoints to install:

- Is there a folder called "MSA" in the webapps directory of Tomcat server?

- Are there five folders (cgi-bin, css, images, scripts, and WEB-INF[3]) under the "**MSA**" folder?
- Are there two folders (classes and lib) under the WEB-INF folder?
- Are there the following seven files under the "lib" directory?
    - axis.jar
    - commons-discovery-0.2.jar
    - jaxrpc.jar
    - commons-logging-1.0.4.jar
    - saaj.jar
    - wsdl4j-1.5.1.jar
    - webserviceutils.jar

- Are there the following folder and file under **MSA** directory
    - prefuse folder
    - myapplet folder
    - agrovoc.html

- Are there the following five classes under the "classes" directory?
    - org\fao\gilw\aims\webservices\ AgrovocWS.class
    - org\fao\gilw\aims\webservices\ AgrovocWSProxy.class
    - org\fao\gilw\aims\webservices\ AgrovocWSService.class
    - org\fao\gilw\aims\webservices\AgrovocWSSoapBindingStub.class
    - org\fao\gilw\aims\webservices\AgrovocWSServiceLocator.class

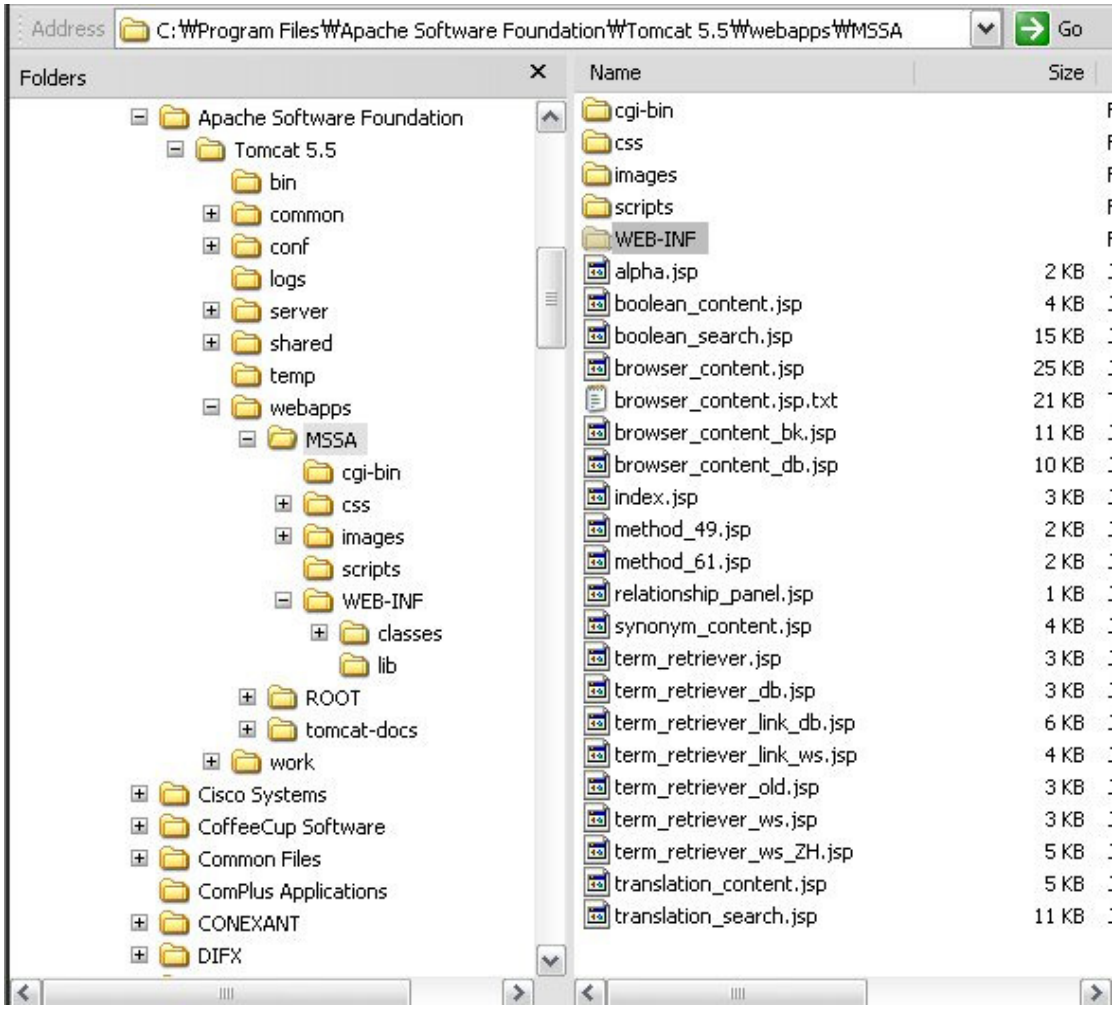

Figure 7. A folder structure and files which are necessary to run MSA. Proper installation contains all folders and files.

3. Source code description

3.1. *Overview*

Table 1 shows an overview of main pages including four functions and help pages and maps to corresponding source codes.

Table 1. Overview of main pages corresponding their source codes

page	Source file	description	Related pages
Main page	Index.jsp	Displaying four functions, language bars (at top right)and help and survey menu (bottom)	Boolean_content.jsp Browser_content.jsp Translation_content.jsp Synonym_content.jsp Help_content.jsp
Build complex search	Boolean_content.jsp	Displaying "build complex search function"	Boolean_search.jsp All image files under " images/bsgraphics"
Browse topics	Browser_content.jsp	Displaying "browse topics" function	agrovoc.html and two folders prefuse and

			myapplet under MSA
Expand languages	Translation_content.jsp	Displaying "expand languages" function	Translation_search.jsp and all files under "script" folder
Expand synonyms	Synonym_content.jsp	Displaying "expand synonym" function	Translation_search.jsp and all files under "script" folder
Help	Help_content.jsp	Displaying "help" page.	n/a

3.2. *Function: "build complex search"*

This function was developed by Java script. All contents are located in Boolean_content.jsp. This page is connected to Boolean_search.jsp which covers all events. All images are located in the folder "images/bsgrpahics".

3.3. *Function: "browse topics"*

This function has been developed using java applet. We have a file insight MSA folder called agrovoc.html, a java class file myapplet.class is using in agrovoc.html while agrovoc.html is working for calling a applet. myaaplet folder have 16 class files but three class files are important for deploying applet, applettree.class, display.class and treeview.class. one prefuse folder is in the MSA folder, that is necessary for displaying tree structure in the view. Finally agrovoc.html is calling applettree.class and aplettree.class is calling treeview.class and display.class, here both file is using prefuse library and web service.

3.4. *Function: "Expand languages"*

This function was developed by JSP and AJAX and all data for showing are from AGROVOC Web services. Two different technologies have been used:

- JSP: To retrieve corresponding terms in other languages of which a user types, MSA system tries to connect AGROVOC web service using "translation_search.jsp" which is a client program of AGROVOC Web Service.

- AJAX: To show auto-suggestion terms when users type any letter, AJAX technology was adapted in MSA. All related codes are located in the folder "scripts". Each language has its own JavaScript file. For example, auto-suggestion for English was implemented in the file "auto_suggestion_en.js". The reason why is that we already put all English terms in the file to improve speed to show auto-suggestion terms rather than to bring all terms from AGROVOC server in running time. However, we need to update auto-suggestion terms in the .js file manually. This is a kind of disadvantage.

3.5. *Function: "Expand synonym"*

This function is very similar with the function "Expand languages". Main difference was the method to be called to AGROVOC web services, since this function is for "synonym". Technologies underneath are same as the function "Expand languages". This was

developed using JSP and AJAX (see section 4.4) and all data for showing are from AGROVOC Web services.

4. Future work

Preliminary demonstration of the prototypes invoked much appreciation and interest from users and Information Specialists. However, several improvements have been proposed:

- combine some elements together (e.g. expand synonyms AND languages together, use the browser for building boolean queries, expand languages and synonyms in a boolean query, etc.);
- in the boolean operators picture show the names of terms;
- boolean query: be able to realize "(A and B) OR (C and B)" or "A and C not B";
- be able to select a different source (e.g. use AGROVOC, FAOTERM, NAL, or CABI);
- navigate graphical browser in multiple languages;
- build complex query using the graphical navigator;
- be able to select multiple destinations engines to which submit the search;
- indicate if the search should be on free text or subject only (integrate this wizard with the AGRIS search wizard);
- do not use customized formats for web services but use SKOS-web services (so we can generalize KOS sources);
- once it uses the AGROVOC concept server data... see how we can exploit the use of the concept URI;
- use KOS mappings to expand also with other terminology from other systems;
- use FAOTERM as exist in web services;
- assess, in both qualitative (by asking the users directly) and quantitative (by doing statistical analysis) how this tool helps their AGRIS Searching;
- allow users to combine some of the above options (different KOS, different SE) to create a "My Search Assistant".

5. Useful links and references

- Access MSA at : http://aims.fao.org/en/tools/multilingual-semantic-search-assistant
- Technical Documentation is at http://aims.fao.org/sites/default/files/uploads/file/Technical_document_MSSA_v_2.pdf.
- Paper published in AFITA 2008: http://mssa.googlecode.com/files/MSSA_final.pdf
- Lucene Home page http://lucene.apache.org/
- Lucene Tutorial http://www.lucenetutorial.com/