The AGROVOC Concept Server Workbench: A Collaborative Tool for Managing Multilingual Knowledge

Panita Yongyuth[1], Dussadee Thamvijit[1], Thanapat Suksangsri[1], Asanee Kawtrakul[1, 2], Sachit Rajbhandari[3], Margherita Sini[3], and Johannes Keizer[3]

1 Department of Computer Engineering, Kasetsart University, Bangkok, Thailand
Email: {panita, dussadee, thanapat}@naist.cpe.ku.ac.th
2 National Electronics and Computer Technology Center, Pathumthani, Thailand
Email: asanee.kawtrakul@nectec.or.th
3 Food and Agriculture Organization of the United Nations, Rome, Italy
Email: {sachit.rajbhandari, margherita.sini, johannes.keizer}@fao.org

Abstract

Ontology plays an important role in the enhancement performance of systems, addressing issues such as knowledge sharing, knowledge aggregation as well as information retrieval and question answering. This paper presents the AGROVOC Concept Server Workbench (ACSW) for multilingual ontological concept construction and maintenance. The ACSW is a web 2.0 based application consisting of two main functionalities that are user management and ontological knowledge management (i.e. concept, scheme, relationship, export, search, validate and consistency check) in order to maintain the knowledge acquisition life-cycle in food and agriculture domain. Knowledge is stored in the form of multilingual concept hierarchy and also kept in the OWL format in order to exchange between machines and to do reasoning. This workbench uses Protégé API as an OWL framework. Moreover the Ontology Game conceptual framework is also presented in order to acquire ontology terms more pleasant.

Keywords: AGROVOC Concept Server Workbench (ACSW), Ontology, Knowledge Aggregation, Knowledge Management, Ontology Game

1. Introduction

Ontology is a formal representation of a set of concepts within a domain and the relationships between those concepts. It is used to reason about the properties of that domain, and may be used to define the domain. Ontology plays an important role in increasing magnitudes with the performance of information processing system such as information integration, taxonomies-based document classification and information retrieval system.

The AGROVOC Concept Server Workbench (here after called ACSW), originated by FAO, is a web-service java tool for collaborative building and structuring multilingual ontology and terminology systems in the area of agriculture with a distributed environment. The main objective of the ACSW (M. Sini, *et al*, 2007) is to create a collaborative reference platform and a "one-stop" shop for a pool of commonly used concepts related to agriculture, containing terms, definitions and relationships between terms in multiple languages derived from various sources. For this workbench, we moved away from a centralized development of ACSW to a Web2.0 inspired way of networked and distributed contributions to create a

system with richer semantics that is going to greatly enhance both the resource indexation and related search, and the information organization in the agricultural domain.

2. Literature Review

One of ACSW goals is: "… to provide a powerful and extensible model that can be used to create other ontology … (M. Sini, *et al*, 2007)". To follow this goal they decide to keep ACSW data in Semantic web content (J. Davies, 2002). The physical model for semantic web content could be either RDF (W3C, 2004) or OWL (Web Ontology Language, 2004). In this project, the developers decide to keep data in OWL format. Currently there are a number of OWL frameworks such as OWL API, KAON, evOWLution, Swede, Jena, Sesame, etc. Three famous frameworks are: Jena, Sesame, and Protégé.

- **Jena** is a java framework for building semantic web applications. It provides a programmatic environment for RDF, RDFS and OWL, and includes a rule-based inference engine (Jena, 2008). It also supports both "Memory Base Ontology Model" and "Persistent Ontology Models". Jena uses SPARQL as a query language to access RDF or OWL data. Jena also provides an API to build various types of knowledge base tools and applications.
- **Sesame** (Broekstra, 2002) is an open source RDF framework with support for RDF Schema, OWL, inferencing and querying (Sesame, 2007). It can be deployed on top of a variety of storage systems (relational databases, in-memory, files systems, keyword indexers, etc.), and offers a large scale of tools to developers to leverage the power of RDF and RDF Schema, such as a flexible access API, which supports both local and remote (through HTTP or RMI) access, and several query languages.
- **Protégé** is a free, open-source platform that provides a growing user community with a suite of tools to construct domain models and knowledge-based applications with ontologies (Protégé, 2008). At its core, Protégé implements a rich set of knowledge-modeling structures and actions that support the creation, visualization, and manipulation of ontologies in various representation formats. Protégé can be customized to provide domain-friendly support for creating knowledge models and entering data. Furthermore, Protégé can be extended by way of a plug-in architecture and a Java-based Application Programming Interface (API) for building knowledge-based tools and applications.

Table 1 shows the feature comparison between Jena, Sesame, and Protégé.

Table1**:** The features comparison between 3 famous OWL Frameworks.

Features	Jena	Sesame	Protégé
Language	Java	Java	Java
Supported model	RDF/OWL	RDF/OWL	RDF/OWL
Inference support	Yes	Yes	Yes (plug-in)
Query Language	SPARQL	SeRQL, SPARQL	SPARQL, Protégé API
Type	Provide API	Web Application, Provide API	Stand alone application, Provide API
Allow having duplicate key in the model	Yes	No	Yes
User control *	No	No	No
Data validation by expert *	No	No	No

***** These features available in ACSW.

What should be an API for ACSW?

Fig 1 (B Liu *et al*, 2005), Sesame-DB has a better query response time than Jena-DB. So we choose Sesame as an API to build ACSW at the beginning. Later, ACSW needs to have more functions then we start to do some experiments in order to observe what should be appropriate API for ACSW. Finally, we have chosen protégé since it provides a better query response time, provides API to manage domain and range, and also allow having duplicate statement in the model.

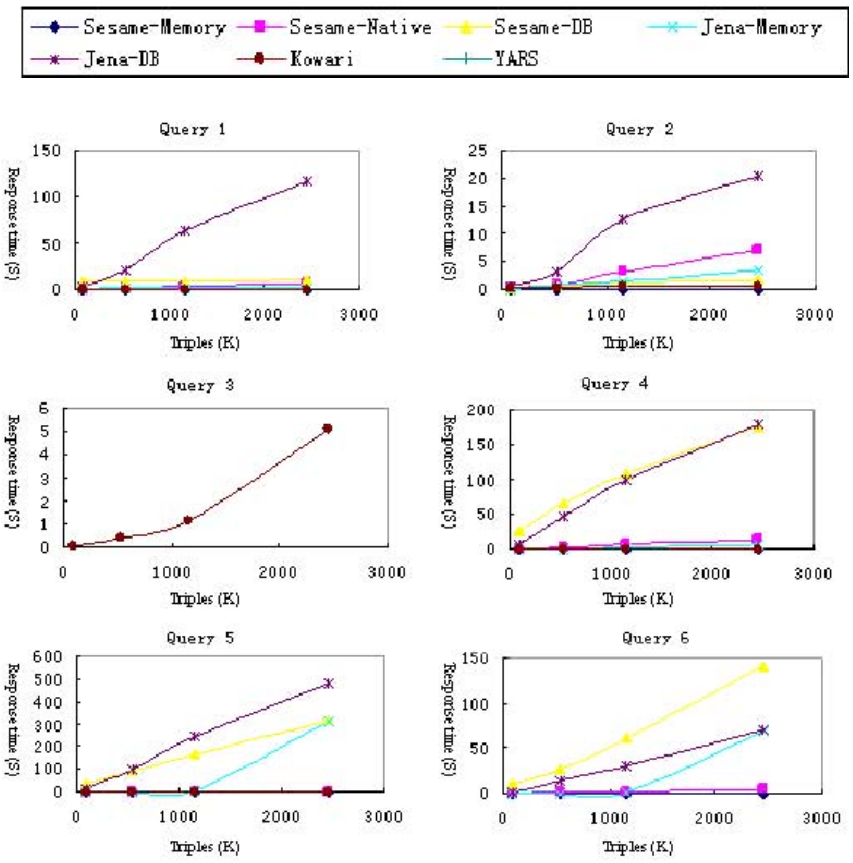

Fig 1: Query response time [Figure from B Liu, *et al*, 2005]

Fig 1 shows query response time of sesame-memory, sesame-native, sesame-DB , Jena-memory, Jena-DB, Kowari and YARS

3. The ACSW's architecture

Fig. 2 shows the generic ACSW platform for Multilingual Ontological knowledge construction and maintenance extended with Authoring Tools.

The ACSW consists of three main parts: the Ontological Knowledge management component, the User management component and the authoring tools.

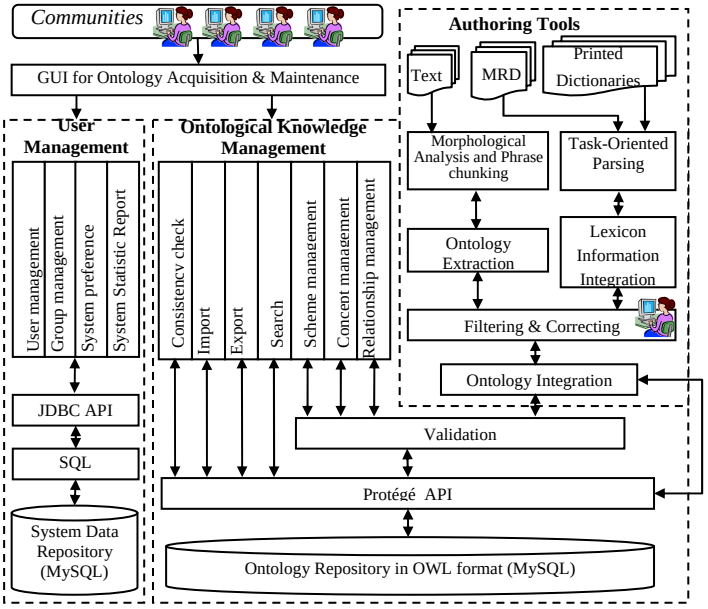

Fig.2 Overall System architecture

3.1 Ontological Knowledge Construction and Maintenance

Since the workbench supports collaborative ontological knowledge construction and maintenance, a good ontological knowledge and user management are needed.

Ontological Knowledge Management

Ontology is kept in OWL (web ontology language) format by using MySQL as the persistence repository. Protégé has been used as OWL framework to do many actions with data in OWL format such as querying , adding OWL statement, deleting OWL statement and exporting data. There are 7 functionalities that user can use for managing ontology.

Concept Management Function. This module provides functionality of concept navigation. The end users can start to create or delete concept from concept hierarchy. After adding a new concept, user can add, edit or delete more information in each component as follows

- *Basic Information*, such as create-date = 2006-10-03, update-date = 2006-10-03.
- *History of change* for tracking the version of concepts with terms in any language.
- *Scope note* for reminding some important information for sharing with the other users.
- *Terms:* that related to the concepts in any language for supporting multi-lingual aspect. Accordingly, when user browses the concept such as “public administration” then he/she could see the terms in the other languages such as “public administration (en)” and “administration publique (fr)”
- *Definition* of the concept in any language for supporting the meaning of the concept especially the technical terms. For example, the definition of the concept Cycadaceae (en) is “ancient palmlike plants closely related to ferns in that fertilization is by means of spermatozoids (en)”
- *Relationship* between ‘users’ selected concept to other concepts.
- *Image* that associated to the concept.

According to the above information, the collaborative ontology construction could be managed more consistently and efficiently. This function also allows administrators to manage about permissions for ontology editors, validators, etc.

Search Function. This function consists of basic search and advance search.

- *Basic search:* User can search concept by using term as the query and results are returned as the concept which has that term. More options variable for providing a better result in this module are using regular expression (contain, exact match and start with), case sensitive and include description.
- *Advance search:* Using the advance search, user can make the result more accurately by filtering concept using concept relationship, sub-vocabulary (geographic, scientific term, etc), term code, and concept status or classification scheme.

Relationship Management Function. The data model of this system is an ontological one which is kept in OWL format. Basically OWL format is a triple pattern (subject-predicate-object). User can use relationship management module to add, edit or delete some predicate that were used in this system. The relationship hierarchy consists of 2 types of relationship properties (e.g object property and data type property). In case of adding new relationship, the users can also add more related information to that relationship. They can also edit or delete the related information components which are listed below.

- *Label* of relationships in any language such as "has category".
- *Definition* of relationship in any language. For example, relationship is "belong to category". Definition is "to map any domain concept to any category".
- *Properties* of relationship such as symmetric, transitive and inverse functional.
- *Domain & Range:* Boundary of subject and object of that relationship. For example, "has image" has "domain concept" as domain and "image" as range.

Consistency Check Function. Checking whether some ontology parts are inconsistency depends on consistency condition. The function will return inconsistency part with solution for that issue.

Validation Function. People can have their own way to construct ontology or maybe they have different background knowledge. As shown in Fig.1, every action that is going to change data in ontology, needs to be approved by two types of user group which are "validator" and "publisher" (ontology expert). The validation function will perform this issue before releasing the updates to the public.

Import Function. It enables to import external ontology in OWL format that has the same schema compared to the system. In case of duplication, system will alert to user.

Export Function. It enables to export ontology from in OWL format to RDF, XML, TBX, SKOS ,OWL (simple format) and RDBMS (SQL, UTF8) format.

Scheme Management. This module is used for grouping concept.

3.2 Ontological *Knowledge Authoring Tools*

One of necessary parts of this workbench is the ontological knowledge authoring tools, (semi-) automatic ontology acquisition component, which supports the users for acquiring the complete and up-to-date ontology (Imsombut A. *et al*, 2007) .This component allows extracting ontological terms, their lexicon information and their relations from different resources, i.e. texts and dictionaries, and integrating them into the core ontology. This component is divided to 3 sub-processes: ontology acquisition process and ontology integration.

3.2.1 Ontology Acquisition Process

The process of (semi-)automatic ontology acquisition (Imsombut A. *et al*, 2007) from texts is composed of two main processes. The first one is the morphological analysis and the phrase chunking and the second is the ontology learning process.

Morphological Analysis and Phrase Chunking. These processes are preprocessing module. The execution of these modules is language dependence so the grammatical rules are

changed to process those various languages. The first step (if needed) is that the printed books are scanned in order to make them to be electronic text. After that, a shallow parser, based on grammatical rules and statistical approach, is applied for identifying the boundary of words and morphological information, e.g. part-of-speech. Next, the outputs are chunked into phrases by using grammatical rules.

Multi-Algorithms for Ontology learning. The multi-algorithms applying for extracting the complete ontological terms and relationships: concept acquisition, NP analysis-based taxonomic and cue-based taxonomic relation acquisition composed of

Concept acquisition module. Concept can be acquired by using term frequencies in texts. The terms that are more frequently used in a domain-specific corpus than in general corpus will be identified as ontological concept and proposed the user to verify.

NP analysis-based taxonomic relation acquisition. The noun phrase analysis technique is used to analyze the surface form of a compound term's head word. If the head word of a term has the same surface form as other terms, the system will apply the IS-A relationship to them. For example, the head word of cow milk is milk which has the same surface form as milk. Then, the system will identify cow milk is a subclass of milk.

Cue-based taxonomic relation acquisition. To identify the intended relationships of the ontological terms, we use explicit cues, i.e. lexico-syntactic patterns (e.g. NP such as NP1, NP2, …) and an item list (i.e. bullet list and numbered list). The main advantage of this approach is that it simplifies the task of concept and relation labeling since the cues can be used to identify the ontological concept and to hint their relations. However, this technique poses certain problems, i.e. cue words ambiguity, item list identification ambiguity, and numerous candidate terms ambiguity. The last problem is very important, especially for the sentence that head word has several modifiers.

The corpus used to test these methodologies deals with the domain of agriculture (Imsombut A. *et al*, 2007). It is the 302,640 words plain text in Thai from 90 documents. By testing with these documents, the system is able to extract about 2,228 concepts and 2,325 taxonomic relations when using multi-algorithms techniques. The performances of the system are 0.74 of the precision, 0.78 of the recall and 0.76 of the F-measure. The important errors of pattern approach are caused by some ambiguities of the cue words.

3.2.2 Ontology Integration and Reorganization

At this step, the related word/phrase pairs are collected from the two types of sources, texts and Dictionaries, and integrated to the existed core ontology by applying two heuristics techniques: If the separated ontological trees have the same label nodes, then merge them. If the terms' head words match partially, then merge them. For example, *Fruit* has head word matching with *Tropical Fruit*. At the current state, there are two operations involved in this process:

- *Addition*: A child node will be added to the core tree, if the parent node has the same label.

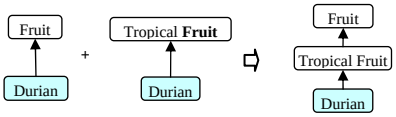

Fig. 3 Example of the insertion operations for ontology integration

- *Insertion:* If the child nodes have the same label as the head word of the parent nodes then the new term that more specific is inserted between two existing ontological terms. Fig. 3 points out an example of the ontology operation for inserting a new ontological tree (right-hand-side tree) into a core-tree (left-hand-side tree).

The process of ontology integration is iteratively occurred when the system adds each extracted concepts and relationships to the core tree (Mukda *et al*, 2008). The system integrated 1,544 relationships that extracted from corpus to the core tree with term matching technique and 595 relationships with partially terms' head words matching technique. The accuracies of these techniques are 0.82 and 0.91, respectively.

4. Experiment and performance testing

To evaluate the performance of the ACSW, several tests with different ontology sizes were carried out. Furthermore, the tests were conducted using different Internet browser available. Table 2 shows the loading time of ontology model by the ACSW.

Table 2: Times Comparison

Ontologies	Sizes	Internet Explorer 7	Safari 3.1.1	Mozilla/Firefox 3.0
Blank model	245KB	2 secs	1 sec	1 sec
Rice Level 2	9.5MB	11 secs	6 secs	5 secs
Rice Level 3	13.5MB	19 secs	7 secs	8 secs
Rice Level 4	16MB	-	8 secs	9 secs

The result from table 2 shows that currently ACSW could load the ontology of size approximately 13.5 MB in any browser. When loading the ontology of size 16MB, it fails in Internet Explorer7, but Mozilla/Firefox3 could load it. In conclusion, Safari browser has better performance than Mozilla/Firefox and IE browser loading larger sized ontology. Further investigations and studies are carried out to load larger ontologies. Several Workshops were organized to evaluate the functionality and performance of Workbench. The previous version of workbench developed using sesame API was used in workshop with 20 users from different organization and background. The major issue raised from that workshop was the duplicate key problem when 20 users trying to create new concepts at the same time. This problem was solved in later version of workbench, which uses Protégé OWL API. Another workshop was organized to test this new version of workbench with 25 users. No duplicate key issue longer existed and all the users were able to create, browse, edit and delete concepts simultaneously. During the test of all the functionalities of the workbench, the same participants from previous workshop mentioned that workbench with Protégé API have faster loading time.

5. Future Plan

In the future, we plan to develop Ontology Game to be as another choice of terms acquisition. Fig. 4 shows the Architecture Model of Ontology Game.

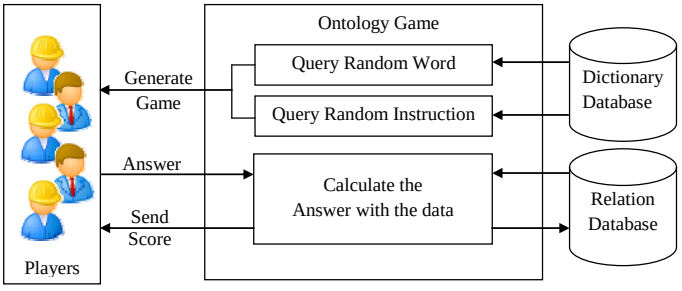

Fig. 4 Ontology Game Architecture Model

- Give **SYNONYMES** for the following word:
- Give **THEMES/DOMAINES** (for example: 'Sports', 'Medicine', 'Cinema', 'Cuisine,' etc.) for the following word:
- Give **PARTS** (for example: 'engine', 'wheel', etc. is the part of 'car' , 'page', 'chapter' etc. is the part of 'book') for the following word:

Ex. 1 Instructions of Game

The Ontology Game will start as following steps (see Fig. 4):

- The program will be random the word and the instruction from the dictionary database. (The example of the instruction see in the Ex. 1)

- The program will generate the game and give the limited time to the players to answer the question.
- The players have to fill the word that related to the given word with the given instruction in limited time.
- After the game finished, the program will calculate the score to the players making the game more attractive and record words statistic to the database.

The Benefit of the Ontology Game

We will gain the many relations between word and word which have verification by the large communities.

6. Conclusion

The workbench, hereby, is originated by The Food and Agriculture Organization of the United Nations (FAO) and has been developed based on web 2.0 by Kasetsart University. ACSW is conceived as a pool of semantically related concepts. All concepts are represented with multiple terms and definitions in many languages. The workbench use protégé as OWL API. We have tested this ACSW with 25 concurrence users. Currently it can support 16 MB maximum size ontology. And safari web browser can provide best loading time. Authoring tool of this version is manual task. We need to improve precision of matching algorithm. Future works would be promoting strategies for this workbench, getting the feedback for tuning of the system, making the system have more robustness and adding the Ontology Game to this AGROVOC workbench.

Acknowledgement

The ACSW has been supported by FAO, NECTEC and KURDI. We would like to give special thanks for Mathieu Lafourcade who created Jeux de mots (Jeux de mots) for giving the idea of Ontology Game. We also thank to Frederic Andres for his suggestion and comment to this paper.

References

M Sini, B Lauser, G Salokhe, J Keizer, S Katz (2007) - Library Review,
The AGROVOC Concept Server: rationale, goals and usage [Online].
Available at http://www.ingentaconnect.com

Jena (2008) “A Semantic Web Framework for Java” [Online].
Available at http://jena.sourceforge.net/

Sesame (2007) [Online]. Available at http://www.openrdf.org/

Protégé (2008) [Online]. Available at http://protege.stanford.edu/

B Liu, B Hu (2005) - An Evaluation of RDF Storage Systems for Large Data Applications in Proceedings of the Joint International Proceedings of the First International Conference on Semantics, Knowledge and Grid (SKG'05) p. 59

Broekstra, J. Kampman (2002) “Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema” in Proceedings of the Joint ISWC2002.

Web Ontology Language (2004) [Online]. Available at (http://www.w3.org/2004/OWL/)

W3C, “Resource Description Framework (2004)” [Online].
Available at http://www.w3.org/RDF/

Mukda S, Dusadee T, Sachit R (2008) Workbench with Authoring Tools for Collaborative Multi-lingual Ontological Knowledge Construction and Maintenance, Proceeding of the LREC'2008

Jeux de mots [Online] Available at http://www.lirmm.fr/jeuxdemots

Imsombut, A., Kawtrakul A. (2007) Automatic building of an ontology on the basis of text corpora in Thai, Proceeding of the Language Resources and Evaluation Journal special issue on Asian Language technology, Springer (2007)