# Rapid Ontology Generation for Use in Semantic Web


| 郭瑞陽 | 許育銓 | 楊晴涵 | 翁照閔 |
|---|---|---|---|
| 逢甲大學資訊四乙:學生 | 逢甲大學資訊四乙:學生 | 逢甲大學資訊四乙:學生 | 逢甲大學資訊四乙:學生 |
| e-mail：super007boss@hotmail.com | e-mail：d9329053@fcu.edu.tw | e-mail：potato-angies@yahoo.com.tw | e-mail：a3127409@yhaoo.com.tw |

*Chyi-Ren Dow, Kuang-Ho Chen, Jui-Yang Kuo, Ching-Han Yang, Chao-Min Weng, and Yu-Chuan Hsu*
*Feng Chia University, R. O. C.*
*{crdow@fcu.edu.tw, cyne@pluto.iecs.fcu.edu.tw, super007boss@hotmail.com, potato-angies@yahoo.com.tw, a3127409@hotmail.com, yuchanhs@hotmail.com}*



## Abstract

*The idea of Semantic Web has been proposed by W3C in recent years. In such a network, individual information pieces are annotated according to ontology, the knowledge base of the domain of the application. While such knowledge-reusing architecture brings about automation as well as compatibilities, the process of the producing of the ontology in question is usually difficult and time-consuming. When processing large body of data, it is usually impossible to produce ontology manually.*

*This work designs and implements an Ontology Generator by using lattice and clustering algorithms. The generator can be used either through the web or as a standalone executable application where datasets and various thresholds can be dynamically imported and configured. Through this generator, ontology can be automatically constructed from entries in database, XML, or even directly from Web pages. To evaluation its performance, it has been used to construct the Computer Science Research Trends Ontology (CSRTO), and outputted it into Web Ontology Language (OWL) format, which is the standing schema of ontology backed by W3C.*


**Keywords:** Ontology.OWL.Ontology Generator

## 1. Introduction

The Internet has become an indispensable technology for the past ten years, from the previous Web 1.0 to the age of user-friendly, content-democratic Web 2.0. Semantic Web has been proposed and pushed by W3C in recent years [9-10]. Under its guideline, documents are no longer linked together simply because their title/content *look* alike, but are established basing on the similarity of their *meaning* (semantics) behind their title/content [3]. Thus computers would be able to find the definition of keywords through hyperlinks, logical reasoning for using keywords, in order to obtain semantic significance of information. These are done through the use of ontology. In [14] it was stated that: "*Ontology can be used as the basis of knowledge, avoid duplication of knowledge of the field, and because of uniform terminology and concepts, making the purpose of sharing knowledge be possible.*" Here, the term *ontology* is a description and definition of the information architecture, resources and the knowledge content for any website [11]. The document is usually defined using ontology languages [12] such as RDF and OWL. Definition of the relationship is between the rule of logic and reasoning for Semantic Web of the knowledge base.

Ontologies are usually manually created and maintained by a single user or an interest group. There are numerous ontologies published on the Internet such as *Friend-of-a-Friend project* [23], *Music Ontology* [24], *Traditional Chinese Medicine Ontology* [25], or those found in [26]. Yet establishing ontology requires significant assistance from experts [15]; manual construction consumes time and human resources. Therefore it is usually hard for a single user or non-experts to maintain their own ontology.

This work designs and implements an Ontology Generator which can be used either through the web or as a standalone executable application where datasets

and various thresholds can be dynamically imported and configured. By applying concept analysis and clustering, ontology can be automatically constructed from entries in database, XML, or even directly from Web pages. Support threshold and similarity threshold can be dynamically adjusted in order to acquire desired complexion of the ontology structure. The generated ontology can be stored in Web Ontology Language (OWL) [20, 22] format, the standing schema from W3C, thus enabling either publishing or future use in semantic web applications. A project of developing an ontology covering computer science research trends was conducted and the preliminary results presented.

The rest of this paper is organized as follows: related work is described in section 2, system architecture in section 3, system implementation in section 4, system prototype in section 5, experiments in section 6, and finally conclusion in section 7.

## 2. . Related Work

Technologies related to this research are described in this section, including ontology defining, ontology construction, and concept lattice.

### 2.1 Ontology Defining

Ontology engineering has been an issue in the field of either knowledge engineering or artificial intelligence for some time [1]. Because of the development of the Semantic Web in recently, the number of related applications grows by the day. The use of ontology and semantic web reduces the world of computer-aided business back to the domain of the business itself, thus users would spend more time solving the domain problems in the real world at hand then dealing with computer-related technology issues [2]. According to [7-8], a body of ontology usually contains several components, namely *concepts*, *attributes*, interconnecting *relationships*, and *axioms*. Standards have also been proposed to describe ontology in more formal and reusable forms, such as Resource Description Framework (RDF) [18]. RDF proposed a simple model to express any condition resources that is called Triple. This model is based on the idea of identifying things using Web identifiers. Points show resources and curve show the relationship of those resources. OWL (Web Ontology Language) [18] W3C Web Ontology Working Group extends from XML, RDF and RDFS. The main of function is defining the basic of the related knowledge of Web Ontologies and Ontologies. The OWL language provides three increasingly expressive sublanguages (Comparison of OWL Full, OWL DL, and OWL Lite)

### 2.2 Ontology Construction

Ontology can be generated from various data types such as text data, knowledge base, semi-structured and/or relational schema [13]. Compared to other types of data, ontology generation from textual data has attracted the most attention. Among techniques used for processing textual data, clustering is one of the most effective techniques for ontology learning. Conceptual clustering techniques such as COBWEB [4] are powerful clustering techniques that can conceptualize clusters for ontology generation.

### 2.3. Concept Lattice

Formal Concept Analysis [5] is a mathematic method by the basic of Lattice Theory that can not only find data's analytic theory of conceptive structure from data's set, but also cluster by basing on object's attribute [4]. This data structure records of the relationship between parent nodes and child nodes. As a result, we can express the relationship of many-to-one or one-to-many in lattice.

## 3. System Architecture

System architecture of the ontology generator will be detailed in this section, including system overview, lattice producing, and clustering.

### 3.1. Overview

As shown in Fig. 1, first input data either from database, web page, or pure text files are extracted and preprocessed, and used to produce a lattice, which will create all possible set of all attributes. The lattice is then clustered using a similarity threshold so that concepts alike to a certain level are merged. The result is similar to ontology. The final encapsulation converts it into sharable OWL document format.

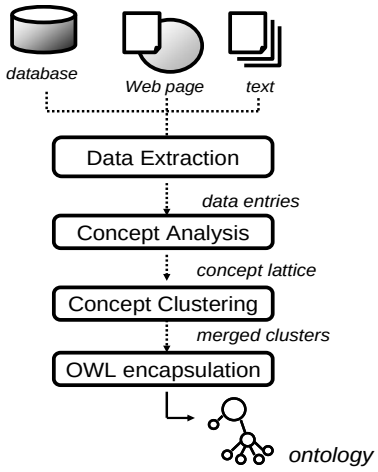



**Fig.1 System flow sheet**

## 3.2. Lattice Producing

In the context of lattice, it is defined that a set of objects as *G*, a set of attribute as *M*, and a relationship as *I* which denotes the on/off status of each objects verse each attributes. The concept lattice is defined as $G \times M \times I$, where as a concept is defined to be a pair ($G_i$, $M_i$) such that:

1. $G_i \subseteq G$, the object set is in the context of the ontology.
2. $M_i \subseteq M$, the attribute set is in the context of the ontology.
3. All object in $G_i$ has all attribute in $M_i$.
4. For every object in *G* that is not in $G_i$, there is an attribute in $M_i$ that object does not have.
5. For every attribute in *M* that is not in $M_i$, there is an object in $G_i$ that does not have that attribute.

| objects / attributes | $D_1$ | $D_2$ | $D_3$ |
|---|---|---|---|
| $m_1$ | 0.5 | 0.02 | 0.1 |
| $m_2$ | 0.318 | 0.8 | 0.56 |
| $m_3$ | 0.004 | 0.42 | 0.3 |

**Fig. 2 Data table for lattice producing**

Shown in Fig. 2 is an example, where $m_1$, $m_2$, $m_3$ denote attributes, and $D_1$, $D_2$, and $D_3$ denote objects. The values represented by $I:\{d_i, m_j\}$ are *membership* values indicating $D_i$'s and the other as relationship *I*. While there is no zero value here, it can be observed that some objects' certain attributes are significantly less then average. To lower the overhead of lattice producing, *support threshold* ($\sigma_s$) on each (or a single bound for all) attribute could be set to filter these insignificants. For example in this example if 0.3 is used, the resulting lattice is shown in Fig. 3.

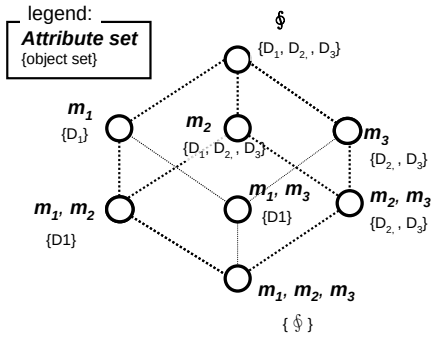



**Fig. 3 Resulting lattice from above example**

From Fig. 3 it could be seen that the upper most concept (node) represents the object set with the constraining attribute that is §, thus all objects are covered. This concept thus maps to many ontology's top *everything* node. In the $m_1$ concept, all objects with its *membership* ($m_1$) $\geqq$ *threshold* are covered, thus covering $D_1$.

## 3.3. Clustering

Although concept latticing yields a rough model for ontology draft, it would be prudent to say that the result is satisfying. For a dataset with *n* attributes, there would be as much as $2^n$ concepts. Simple threshold filtering is not enough a refining. Similar concepts must be able to be *merged*.

The similarity between two fuzzy sets of A and B are defined such as:

$$E(A, B) = \left| \frac{A \cap B}{A \cup B} \right|.$$

The fuzzy similarity function could be replaced by other domain based similarity function since the problem of deciding of how similar certain two entities are is usually a subjective matter. Same at lattice producing, a *similarity threshold* ($\sigma_t$) could be used. If the similarity between two concepts is qualified, they would be merged.

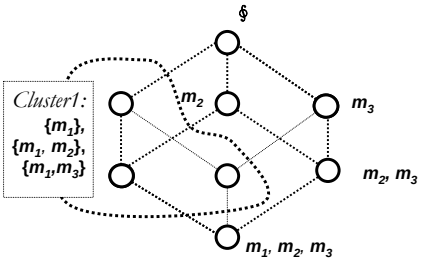



**Fig. 4 Similar concepts in a lattice**

In Fig. 4, for example, the concepts $\{m_1\}$, $\{m_1, m_2\}$, and $\{m_1, m_3\}$ are similar, thus will be merged, resulting in clustered hierarchy as shown in Fig. 5.

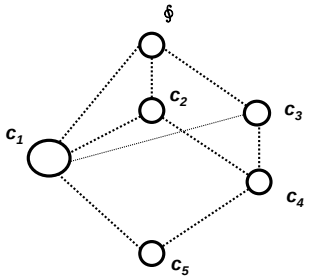



**Fig. 5 Clustered results**

## 4. System implement

In this section, the implementation of the generator is explained.

### 4.1. Data Extraction

The generator would take *G*, *M*, and *I* from three files. This work first aimed at providing a CS research trends ontology generated from literature portal. Thus necessary information should be arranged from the portal's database, thus documents are collected as *objects*, and keywords as *attributes*, and the document v.s. keyword flags as *relationship*, the resulting schema should be as shown in Fig. 6.

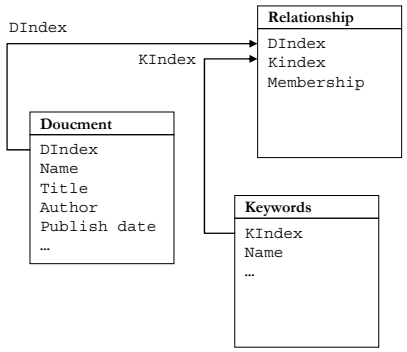



**Fig. 6 Converted Data Schema**

### 4.2. Lattice

In the first version, full lattice with no threshold is first created using the data in Table 1, generating all possible combination of keywords, as shown in Fig. 7. Then the combinations occurred less then 2 times are removed, resulting the lattice shown in Fig. 8

**Table 1 Documents and its Keywords**

| Document | Keyword |
|---|---|
| D1 | K1,K2,K3 |
| D2 | K1,K3 |
| D3 | K2 |
| D4 | K1,K2,K4 |

But it uses this way has one problem. When the number of keywords rises, the overhead would be horrible. 1000 keywords would generate $2^{1000}$ possibilities. Thus method in [17], indicated in Fig. 9, is adapted instead. First it will get a single keyword set. If not qualified, it will be removed. The left result is then used to create the set by calculating all combination, and then remove the unqualified sets until all sets are qualified. Much time could be saved in this fashion.

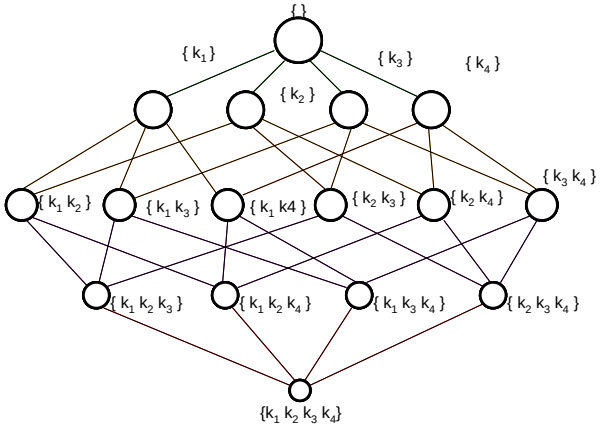



**Fig. 7 Exemplary lattice with threshold = 0**

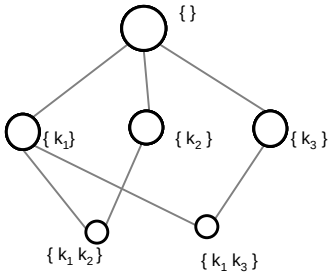



**Fig. 8 Exemplary lattice with threshold = 2**

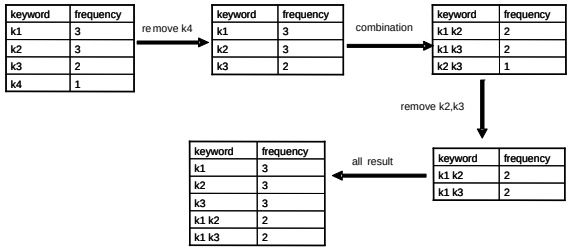
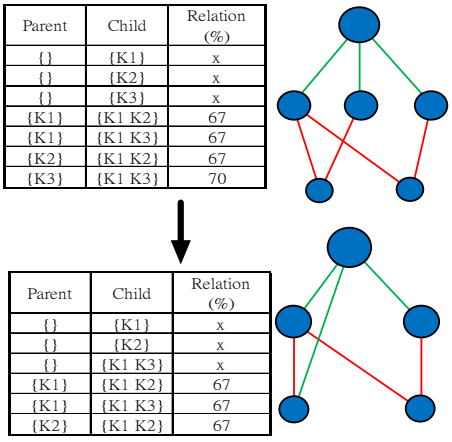



**Fig. 9 Latticing Method (D. L. Yang, 2002)**

### 4.3. Clustering

To clustering, first the relation between parent and child should be calculated. The relation is all numbers of parent of document divide its own numbers of child of document. Then merge in top-down and left-right matter when the relationship exceeds the threshold. But when the child of parent has other parent, the parent with highest relation would be picked.

However, when we take the data to the rule, we found the result gives no denotation. So we have to change the rule. We employed the method in [6]. If relation is higher than threshold, merge parent and child. The different is we merge parent and child when their relation is the highest between the child and its own parent. For example, which uses Table1 data then use the method shown in Fig. 10. The pseudo code is shown in Fig. 11.



| Parent | Child | Relation (%) |
|---|---|---|
| {} | {K1} | x |
| {} | {K2} | x |
| {} | {K3} | x |
| {K1} | {K1 K2} | 67 |
| {K1} | {K1 K3} | 67 |
| {K2} | {K1 K2} | 67 |
| {K3} | {K1 K3} | 70 |

| Parent | Child | Relation (%) |
|---|---|---|
| {} | {K1} | x |
| {} | {K2} | x |
| {} | {K1 K3} | x |
| {K1} | {K1 K2} | 67 |
| {K1} | {K1 K3} | 67 |
| {K2} | {K1 K2} | 67 |

**Fig. 10 Clustering Implementation**

```
cluster
INPUT threshold, data[,]
FOR  i ← 0 to data.length
  IF the relation between child and
parent higher than threshold
    mix(data[i].child,
data[i].parent)
  ENDIF
END
```

**Fig. 11 Clustering Pseudo Code**

### 4.4. OWL transform

According to the relation line to write code, we show an example in Fig. 11.

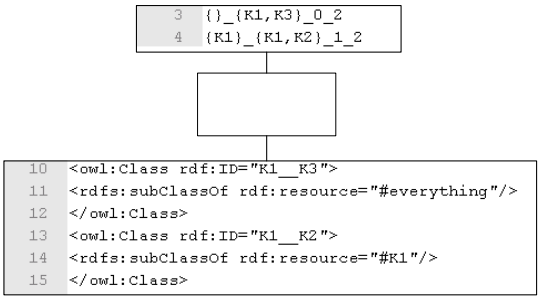



**Fig. 12 Convert relation line data to OWL code**

Note that underlines are used to divide the word into two elements since many OWL editors are particularly problematic with special symbol such as space, comma, etc, but are fine with underlines. And some keywords or documents have two or more letters, so single underline is used to divide a phrase into two words, and two underlines are used to separate two entities. We gathered from Fig. 12 that the relation of the concept {*everything*} and the concept {$k_1$,$k_3$} is *subClassOf*. The relations of {$k_1$} and {$k_1$, $k_2$} are *classes* and *subClassOf*.

## 5. System Prototype

The usages and screenshots of the implemented system are described in this section.

The application accepts txt files as inputs only that they should be in the right format. In such file, object and Keyword from underline. The system would then execute lattice producing. At first users need input the threshold of lattice (The number of times of keywords came out will be leaved out below the threshold). It can show the result in default program in your computers, as shown in Fig. 13. Threshold can be

dynamically selected using a slide bar, and the graph would change accordingly.

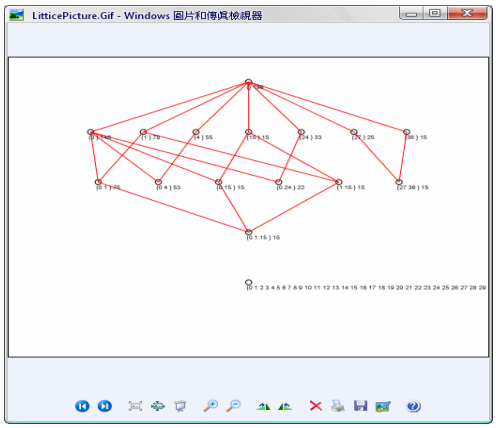



**Fig. 13 Resulting Lattice Graph**

For clustering, we need to input the threshold (The value is the lowest relation between set and set). There is also side bar that can adjust the threshold, as shown in Fig. 14. The result of cluster can be saved to another file (*.owl).

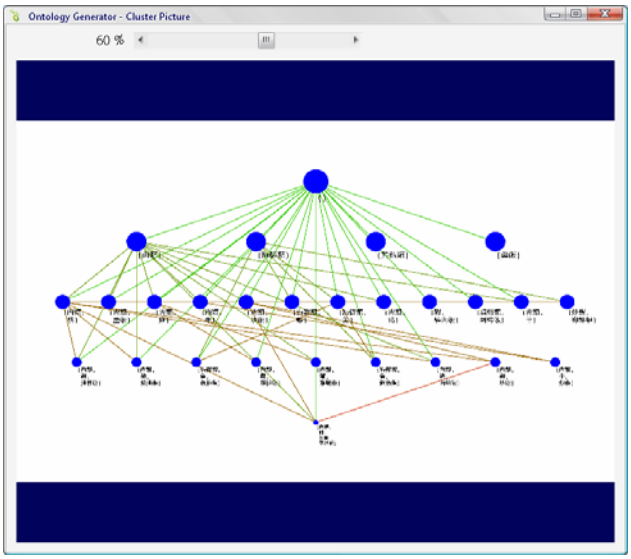



**Fig. 14 A Clustered Result**

## 6. Experiments

To test the proposed generator, we aimed at building an ontology that covers research trends in the field of computer science.

### 6.1 DSpace Project

The test data is from the DSpace portal of FCU [19], in which stored compounds of academic literatures. Including in these collections are proceedings of International Computing Symposium (ICS, 國際計算機會議) and National Computing Symposium (NCS, 全國計算機會議), both held by the Computer Society of the Republic of China [18] (CSROC, 中華民國電腦學會). Enclosed in them are papers which are input into the generator. The data from DSpace has about 2,400 documents, more than 10,000 keywords. It was expected that the resulting ontology can provide a quick glimpse of the research trends of computer science in Taiwan in the past ten years. Since the data are Web-based, a Web crawler agent was designed to mine the document information into the generator.

### 6.2 Problems

Although with a 10,000-keyword input, the keyword repetition frequency was not adequate, thus the ensuing lattice was noised. Thus phrases were cut in order in increase relationships. For example, the term *data mining* would occupy three tokens: *data*, *mining*, and *data mining*. By doing its relation with either *Web mining* or *database* would increase. Now the keyword repeat is better. The new most appear times are network. It appears 177 times. But the new way has a problem. The way to solve problem is drop the illegal keyword. The illegal keyword such as “of”, it appears about 50 times. We remove the illegal keyword. We cut the illegal keyword, but the number of keyword is huge. When we run the program ask appear times need more than 10 times and relation is 50%, it spent us 10 hours. It really runs too long. We decide cut the keyword which is appear times less than 5 times. To do this we find less than 5 times keywords are above keywords of 10,000. When we cut the keyword which is less than 5 times, and it run the program only need 120 seconds. It really saves lots of times. Some keywords appear times a lot, so its subclass we can see. But for the other appears times not enough could be only show single keyword set. There is no subset behind. We write a program name subclass to let us can see the difference ontology model. We not only can see the DSpace ontology model, also can see the ontology of keyword such as database ontology.

### 6.3 Result

After several testing with both $\sigma_s$ (support threshold) and $\sigma_f$ (similarity threshold), it was deemed that the result with $\sigma_s$=35 and $\sigma_f$ =50, as shown in Fig. 15, is acceptable as the macro view of the ontology, with most concepts being major research trends.

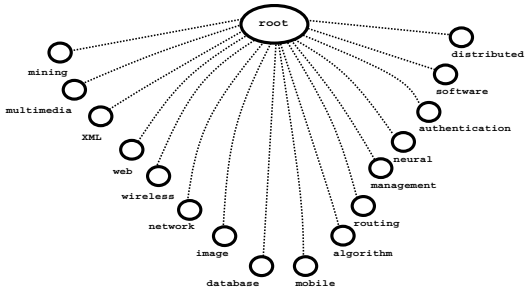



**Fig. 15 CS-Research Trend Ontology,**

However, it is plain that with this high a threshold, it is impossible to have second or third level concepts, since the frequency of lesser keywords would not pass $\sigma_s$, and a lower $\sigma_s$ would produce many inconsequential concepts, thus it become necessary to look at the subontology of each trend with thresholds lower then the macro view. Shown in Fig. 16 is the subontology centered on the cluster *mobile*, with $\sigma_s$ =3, $\sigma_f$ =50.

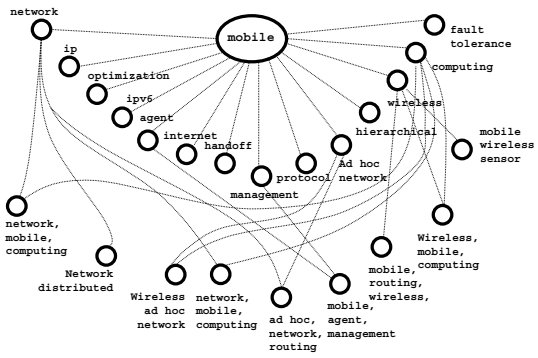



**Fig. 16 subontology of trends *mobile***

## 7. Conclusions

This work design and implemented an automatic ontology generator, which enabled quick construction of ontology body from raw data either form database, or from input text file.

In the process, It must go through the analysis of possibilities, some inappropriate keyword deleted, retained the needs of users, converted into a suitable lattice, and use of our clustering merged into a group of the higher the higher relation, we can further naming it.

In addition, the generator is able to plot the ontology generated, in the form of either classes or subclasses. The result can also be converted into OWL files, enable later editing using tools such as Protégé and OWL VE, or later uses for Semantic Web.

Implemented using C#.net, currently it can be run from most desktop computers. In the future, we hope it could be constructed as Web service based applications, where data could be submitted through browsers.

## 8. References

[1] B. Chandrasekaran, J. R. Josephson, and V. R. Benjamins, "What Are Ontologies, and Why Do We Need Them?" IEEE Intelligent Systems, 14(1):20-26, 1999.

[2] M. Crubezy, M. O'Connor, Z. Pincus, and M.A. Musen, "Ontology-Centered Syndromic Surveillance for Bioterrorism," IEEE Intelligent Systems, vol. 20, no. 5, pp. 26-35, Sept/Oct, 2005.

[3] M. C. Daconta, L. J. Obrst, K. T. Smith, "The Semantic Web: A Guide to the Future of XML, Web Services and Knowledge Management", Wiley Publish, May 2003.

[4] D. Fisher, "Knowledge Acquisition via Incremental Conceptual Clustering," Machine Learning, vol. 2, pp. 139-172, 1987.

[5] B. Ganter et al, "Formal Concept Analysis: Foundations and Applications," Lecture Notes in Artificial Intelligence, no. 3626, 2005.

[6] T. R. Gruber, "A Translation Approach to Portable Ontology Specifications".

[7] C. S. Lee et al, "Recent Research on Ontology Applications," Proceedings of the 8th International Symposium on Advanced Intelligent Systems (ISIS 2007), 2007.

[8] C. S. Lee et al, "Fuzzy Ontology and Its Use on News Summarization," IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, Vol. 35, No. 5, 2005.

[9] K. Moller and S. Decker, "Harvesting Desktop Data for Semantic Blogging",in Proceedings of ISWC 2005 Workshop, 2005.

[10] T. O`Reilly, "Unleashing Web 2.0: From Concepts to Creativity, " 2006

[11] E. Ruckhaus, M. E. Vidal, "An Ontology Language to Describe and Query Web Sources",WIDM'03, New Orleans, Louisiana, USA, November 7–8, 2003.

[12] Y. Sure, R. Studer, A Methodology for Ontology-based Knowledge Management, In J.Davies (Ed.), Towards the Semantic Web: Ontology-driven Knowledge Management, Wiley, November 2002.

[13] Q.T. Tho, S.C. Hui, A.C.M. Fong and T.H. Cao, "Automatic Fuzzy Ontology Generation for Semantic Web," [17] D. L. Yang et al, "An Efficient Closure-Based Method for Discovering Maximal Frequent Itemsets, " Master Thesis, Dept. of Information Engineering and Computer Science, Feng Chia University, 2002.

[18] "Computer Society of the Republic of China", http://www.csroc.org.tw

[19] "DSPACE at FCUniversity," http://dspace.lib.fcu.edu.tw:8080/dspace/

[20] OWL Web Ontology Language Guide", W3C Recommendation 10 February 2004, http://www.w3.org/TR/2004/REC-owl-guide-20040210/

[21] Resource Description Framework (RDF)", http://www.w3.org/RDF/

[22] "W3C", http://www.w3.org/TR/owl-features/

[23]"Friend-of-a-Friend project", http://www.foaf-project.org/, retrieved 2007.

[24] Music Ontology Specification, " http://musicontology.com/, retrieved on 2007.

[25] " Semantic-based Search and Query System for the Traditional Chinese Medicine Community," http://www.w3.org/2001/sw/sweo/public/UseCases/UniZheijang/, retrieved on 2007.

[26] "KBS/Ontology Projects World Wide", http://www.cs.utexas.edu/users/mfkb/related.html, retrieved on 2007.

IEEE Transactions on Data and Knowledge Engineering, 2006.

[14] S. William T. Austin, "Ontology", IEEE Intelligent Systems, 1999.

[15] S. S. Weng, H. J. Tsai, S. C. Liu, and C. H. Hsu, "Ontology Construction for Information Classification," Expert Systems with Applications, 2005.

[16] T. C. Wesley, and Jihoon Yang, "Extracting Sentence Segment for Text Summarization： A Machine Learning Approach", In ACM SIGIR 2000, 152-159, 2000.