Computing and Informatics, Vol. 28, 2009, 781–793

ONTOCS: A WEB-BASED SYSTEM FOR COLLABORATIVE ONTOLOGY CONSTRUCTION

Dosam Hwang, In Keun Lee, Jason J. Jung

Yeungnam University
Dae-Dong, Gyeongsan, Korea 712-749
e-mail: dshwang@yu.ac.kr, iklee@ynu.ac.kr, j2jung@gmail.com

Revised manuscript received 19 January 2009

Abstract. A number of studies on ontology editing tools and ontology-based applications have been proposed for automatically processing knowledge and information. However, the existing methodologies and tools for dealing with ontologies have assumed that the system is restricted to a single user. Main motivation of this paper is to foster collaborations between users, because ontology building is an expensive task. Thereby, in this paper, we present a web-based ontology construction and integration system, which is called OntoCS, to support collaborative interactions between people during creating ontologies. Particularly, inexpert users can collect available language resources from the web to describe concepts in a (even unfamiliar) domain. We believe that this collaborative process is implementing collective intelligence. In conclusion, we have shown that the proposed OntoCS system can efficiently edit and manage multiple ontologies over time.

Keywords: Ontology engineering, collaborative works web-based systems

1 INTRODUCTION

Ontology has been playing an important role of developing domain-specific intelligent systems [1, 2]. Such intelligent systems can acquire knowledge about resources on heterogeneous information systems by referring to the corresponding semantic information which describes the resources, and efficiently understanding which context is involved to them [3, 4]. More importantly, ontology-based systems have to be semantically interoperable with other systems for automatically exchanging useful information and knowledge with each other [5].

However, we can easily realize that it is very difficult to practically implement such ontology-based systems in real world. The main problem is that high cost and too much effort is required for ontology construction. It means that a number of experienced domain experts need to be involved in providing and formalizing their own knowledge for quite a long time. In other words, similar to traditional expert systems, there is a significant gap between knowledge engineers who can build ontologies and domain experts who have knowledge. Consequently, most of the existing methodologies for ontology construction have been focusing on single specific domains. Well known domains are represented by biomedical engineering [6, 7, 8] and chemical engineering [9, 10].

Differently from such domain-specfic ontology developments, we want to investigate a general-purpose ontology construction methodology. Moreover, by using the system based on this methodology, non-expert users (who are neither well trained nor aware of the domain knowledge) will be able to take part in the ontology construction tasks.

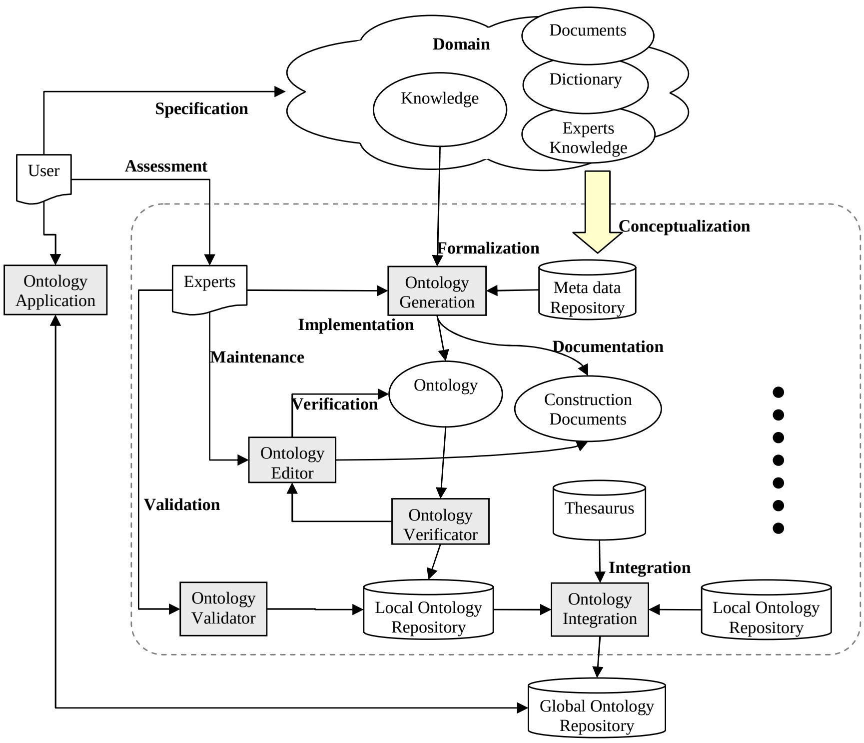

Fig. 1. Ontology development process

To do so, we need to describe the whole process of ontology development, in terms of ontology engineering. As shown in Figure 1, there are several issues for taking care of ontologies and their life cycles:

- Ontology population by instance collection
- Ontology generalization by conceptualization and formalization
- Ontology integration by mapping and alignment
- Ontology validation and assessment
- Ontology maintenance and evolution.

In this work, we are mainly focusing on two issues, which are

i) ontology population by instance collection, and

ii) ontology integration by mapping and alignment.

The first issue of ontology population is regarded as the most laborious work. It means that most manpower and time are required for dealing with the issue. For the second issue of ontology integration, we introduce an *ontological relaxation* process which is capable of taking into account candidate ontological elements of a given ontology.

Thereby, in this paper, we present a web-based ontology construction and integration system, which is called *OntoCS*, to support collaborative interactions between people when creating ontologies. More particularly, in terms of collective intelligence, most of public users can participate the ontology building process, and share relevant semantics for realizing and capturing emergent semantics and knowledge.

The outline of this paper is as follows. Section 2 shows main system architecture of the proposed OntoCS system. In Section 3, we describe collaborative ontology construction and ontology integration methodologies by using OntoCS system. For evaluating the proposed methodologies and system, Section 4 mentions experimental results that we have collected during applying OntoCS system to the real ontology development. Section 5 addresses our experiences on implementing and exploiting OntoCS system, and compares it with some existing work. Finally, in Section 6, we draw a conclusion of this paper.

2 SYSTEM ARCHITECTURE OF ONTOCS

In order for non-expert users to build domain ontologies, we want to support them to refer to technical documents and dictionaries. We have considered two design points;

i) multi-working space for exploiting multiple language resources all together, and

ii) sharing platform to connecting the references and ongoing ontologies.

Thereby, in this paper, we have developed OntoCS system meeting the following requirements; multiple users efficiently have to be able to

1. share language resources and metadata with each other
2. extract and manage the language resources
3. manage the relaxed ontologies, and
4. automatically transform the relaxed ontologies to OWL ontologies.

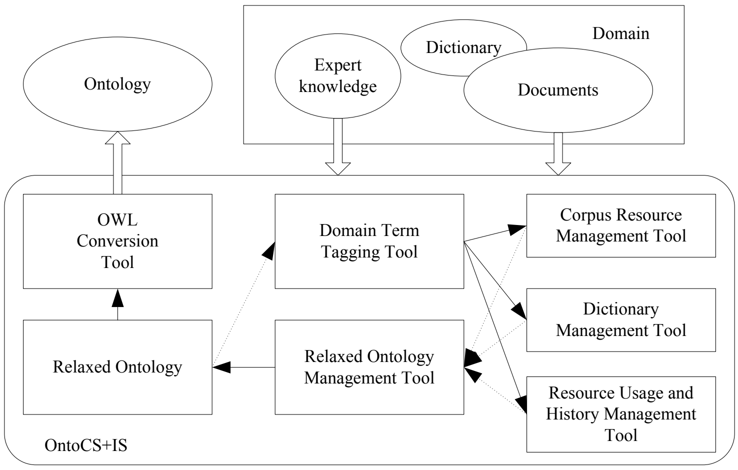

Fig. 2. System architecture of OntoCS

As shown in Figure 2, the whole system is composed of six tools. Corpus resource management tool and dictionary management tool are able to manage language resources for ontology building. Additionally, resource usage and history management tool can present and manage a list of language resources applied by individual users. Relaxed ontology management tool can generate and edit "relaxed" ontologies by using conceptualized knowledge. Domain term tagging tool can automatically indicate specific terms from given documents (e.g., either by changing colors or by changing fonts) as referring to the relaxed ontologies. This tool assists users by improving the understandability of documents. Finally, the relaxed ontologies can be transformed to OWL ontologies by OWL conversion tool.

Another important feature of OntoCS is to monitor and manage user activities. Social collaborations with remote users and asynchronous moments might show inconsistency problems on the ontologies. Although OntoCS can not support automatic consistency checking of the ontologies, we have to take into account how to manage user activities. Thereby, all of the user activities and behaviors are recorded on the system. The results can be visualized and reported to system administrators. For example, Figures 3 and 4 show us two types of statistical information on OntoCS;

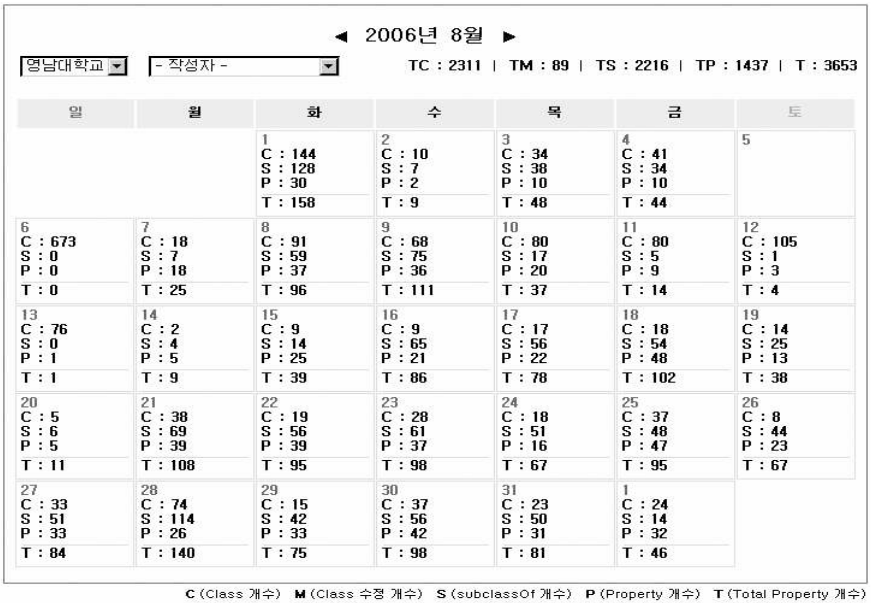

Fig. 3. Statistics of Ontology Construction on OntoCS

i) amounts of ontologies (e.g., numbers of concepts, properties, and instances) generated by each user and each day, and

ii) working time taken during ontology construction.

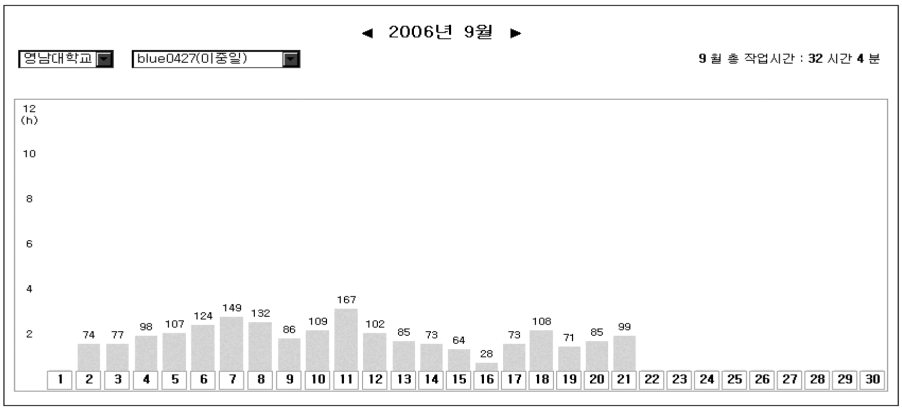

Fig. 4. Working time of Ontology Construction on OntoCS

3 COLLABORATIVE ONTOLOGY CONSTRUCTION

As a matter of fact, there have been many software tools for editing ontologies. Such ontology editor tools are Protégé [11], OilED [12], OntoEdit [13], and SWOOP [14]. Most of them are appropriate to single users. More seriously, users can not share their ontologies, until they have finished ontology development tasks.

Differently from them, we propose a novel system, called *OntoCS*, to collaboratively build ontologies on the web. It means any users who can access to web will be able to participate in ontology development process. All of the domain decisions are in keeping with the OWL nature and specifications. Thus, multiple ontologies are supported simply, a variety of OWL presentation formats are used to render ontologies, OWL reasoners can be integrated for supporting consistency checking, and open-world semantics are assumed while users are editing the ontologies.

3.1 OntoProcess and OntoMProcess

Ontology building process, called *OntoProcess*, is employed into the OntoCS system. Here, we want to briefly describe the steps of OntoProcess as follows.

1. Given a certain seed term, we have to search for a set of relevant documents and sentences in a certain specific domain.
2. From the retrieved document (or sentence) set, candidate domain terms are chosen according to the user's subjective opinions.
3. One domain term is selected from the candidate terms.
4. Ontology is scanned by using the selected term.
   (a) If the selected term is already in the ontology, then go to Step 3.
5. The selected term is looked up in dictionaries and on the web.
6. Referring to the definitions from the dictionaries and web, the user has to determine whether the selected term should be inserted into the ontology or not.
   (a) If definition of the term is clearly understandable and the term is in the domain, then the term should be inserted into the ontology.
   (b) If definition of the term is not clearly understandable and the term is in the domain, then the term should be inserted into the ontology as an undefined term.
   (c) If the term is not in the domain, then the term should be removed from the candidate terms.
7. Until determination process of the candidate term set chosen in Step 2 is completed, Steps 2 to 6 are repeated.
8. The relationship between terms should be extracted.

(a) If relationship between the selected terms does not exist in the ontology, then the relationship should be inserted.
(b) Relationship between the selected terms should be defined by choosing the relationship from the ontology.

9. Step 8 should be repeated, until the selection process of relationships between the terms is finished.
10. Step 1 should be repeated to Step. 9.

(a) If there is any mistake in selecting terms and relationships, the correction should be made from Step 1 to Step 9.

As shown in Figure 2, OntoProcess can represent concepts by using dictionaries. Given a certain seed term, we have to search for a set of relevant documents and sentences in a certain specific domain.

In addition, OntoMProcess is needed for multiple users. The proposed system allows them to manipulate ontologies. However, there might be some problems such as redundancy and conflicts between labels of concept and relationships generated by different users.

This problem can be dealt with by monitoring other users' activities, because all activities are recorded on the server. Also, for better understandability of others' ontology results, the meta-ontologies are represented as a table, instead of formal language (cf. Tables 1–4).

Concept name	digital_television
Definition	Digital television (DTV) is sending and receiving of moving images and sound by discrete (digital) signals, in contrast to the analog signals used by analog TV. Introduced in late 1990s, this technology appealed to the television broadcasting business and consumer electronics industries because it offers new financial opportunities.
Synonym	digital TV

Table 1. An example of concepts in a meta-ontology

4 IMPLEMENTATION AND EVALUATION

In this project, we have implemented a web-based system called *OntoCS*[1]. Figure 5 shows an user interface of OntoCS.

To evaluate this web-based system, we have invited two groups of users (G_A and G_B). User group G_A was asked to exploit OntoCS system to build ontologies, while user group G_B has done ontology building tasks in normal single applications.

[1] OntoCS, `http://ontology.yu.ac.kr`

Domain	digital television
Property	isSubclassOf
Restriction	someValuesFrom
Range	television
Source	"In order for digital television to be broadcast, it must initially interoperate with analog television. When analog television ceases to exist, digital television signals must not interfere with each other. Propagation research carried out by several important digital television regulators has derived a table of acceptable parameters for tolerable interference margins. This table provides all the important acceptable interference margins."

Table 2. An example of concepts in a meta-ontology

Instance	Pavv SVP-42Q2HL1
SuperClass	digital television

Users in G_A have been trained not only what ontologies are, and how they are represented, but also how to use OntoCS.

As shown in Figures 3 and 4, we can keep track on ontology building activities on OntoCS. Table 4 shows temporal duration of five sampled users.

Figure 6 also presents the temporal duration at each time of experimentation. We found out that all of the users spent less and less time for building ontologies, as repeating the experimentation. The users are usually getting adjusted to efficiently use the OntoCS system (except user A).

5 DISCUSSION

In this section, we want to make some discussions which we have practically experienced during implementing and applying OntoCS for building real world ontologies. Of cause, there are many technical and usage problems on collaborations. We can summarize them into two major problems, which are

i) semantic inconsistency between multiple users, and

ii) redundancies of similar knowledge.

Domain	Samsung
Property	isProducerOf
Range	Pavv SVP-42Q2HL1
Source	`http://www.naver.com/ecoms?Redirect=Log`

Table 3. An example of relationship between instances in a meta-ontology

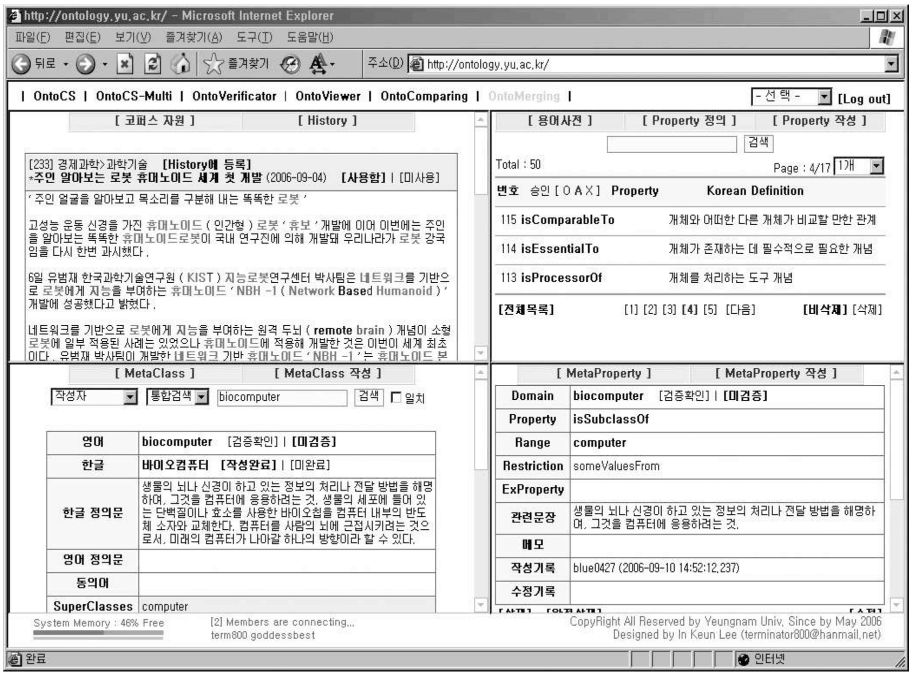

Fig. 5. User interface of OntoCS

Users	A	B	C	D	E
Number of concepts	505	564	814	674	1 066
Working time (minutes)	1 122	786	1 061	1 134	1 451
Average time per a concept (minutes)	2.2	1.4	1.3	1.6	1.3

Table 4. Temporal duration for ontology development

5.1 Practical Problems of Collaborations

We have realized that inconsistency problems are caused by collaborations. Now, we want to address the practical problems and how to deal with them.

5.1.1 Preprocessing for Notation Unification

First problem is in mismatches between language resources written by multiple users. We have solved this problem by establishing a unified notation protocol for all participants.

Case sensitivity. OWL ontologies are case sensible. Consequently, for example, concept "Digital_Television" can not be identical with "digital_television". In OntoCS, all users have to use lower case for writing concept labels.

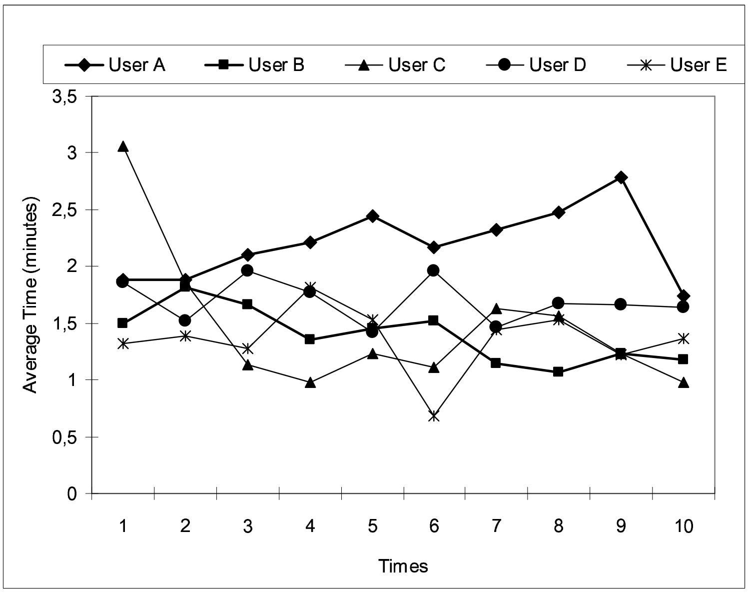

Fig. 6. Experimental results on ontology construction

Spacing words. As to spacing, compound words can not be identical, even though they have same semantics and alphabets (for example, the concepts "biotechnology" and "bio_technology"). To solve this problem, users are asked to check whether the same compound word has been inserted or not.

Abbreviated words. Abbreviation is another problem for making ontologies consistent. For example, "LCD" stands for "liquid_crystal_displays". We do not allow users to use any abbreviations without administrative exceptions, and if the abbreviated word or concept is well-known, users can attach it to the concept.

Singular/plural words. For example, the concept "amplifiers" is not the same as "amplifier". We do not allow users to use plural form without administrative exceptions.

Symbols. Some mathematical and logical symbols can result in technical errors on parsing OWL ontologies (e.g., encoding and decoding). For example, concepts "C#" and "C++" make the system confused. Thus, users have to write down full spelling instead of the symbols (i.e., "C_sharp" and "C_plus_plus").

Polysemy. Some concepts have multiple semantics and meaning. Thus, for disambiguating the concepts, users have to consider exact contexts of the concepts, and represent them more specifically. For example, the concept "Jaguar" should be replaced either by "jaguar_vehicle" or "jaguar_animal" according to the relevant contexts.

5.1.2 Defining Relationships Between Concepts

Originally, OntoProcess was supporting users to freely define relationships between concepts. Whenever users need to describe a relationship between concepts, they simply defined the relationships, without checking whether there exist any similar relationships in the same ontologies. For example, relationships "isUsedAs", "isUsedIn" and "isUsedTo" are generated by different users, even though the relationships are derived from the relationship "Use" and are similar with each other.

To deal with this problem, the system administrator has to standardize the relationships between concepts. It means that the relationship hierarchy should be designed, similar to the concept hierarchy.

5.1.3 Unstructured Concept Hierarchy Structure

Due to lack of domain-specific knowledge, non-expert users may make some mistakes on establishing subsumption relationships between two concepts. (Simply we can consider the concept hierarchy structures of ontologies.) Intuitively, the solution of this problem might be background knowledge, e.g., dictionaries and other references.

Similarly, to deal with this problem, the system administrator and domain experts have to standardize the subsumption relationships between concepts, and validate the topological patterns on concept hierarchy.

6 CONCLUDING REMARKS AND FUTURE WORK

According to the characteristics and purposes of domain ontologies, it is very difficult and expensive for a small group of domain experts to design and build the ontologies. Thus, some existing studies have proposed several approaches, e.g.,

i) collaborative framework among end-users and

ii) ontology reuse.

In conclusion, this OntoCS system is designed and developed to provide web-based collaborations between multiple users. More particularly, in terms of collective intelligence, most public users can participate the ontology building process, and share relevant semantics for realizing and capturing emergent semantics and knowledge.

Acknowledgement

This work was supported by the Yeungnam University research grants in 2007. Also, this work was supported in part by MKE & IITA through IT Leading R&D Support Project.

REFERENCES

[1] Regoczei, S.—Plantinga, E. P.: Creating the Domain of Discourse: Ontology and Inventory. International Journal of Man-Machine Studies, Vol. 27, 1987, No. 3, pp. 235–250.

[2] Faro, A.—Giordano, D.: Ontology, Esthetics and Creativity at the Crossroads in Information System Design. Knowledge-Based Systems, Vol. 13, 2000, No. 7–8, pp. 515–525.

[3] Eriksson, H.—Puerta, A. R.—Musen, M. A.: Generation of Knowledge-Acquisition Tools from Domain Ontologies. International Journal of Human-Computer Studies, Vol. 41, 1994, No. 3, pp. 425–453.

[4] Jung, J. J.: Exploiting Semantic Annotation to Supporting User Browsing on the Web. Knowledge-Based Systems, Vol. 20, 2007, No. 4, pp. 373–381.

[5] Hori, K.: An Ontology of Strategic Knowledge: Key Concepts and Applications. Knowledge-Based Systems, Vol. 13, 200, No. 6, pp. 369–374.

[6] Baneyx, A.—Charlet, J.—Jaulent, M.-C.: Building an Ontology of Pulmonary Diseases With Natural Language Processing Tools Using Textual Corpora. International Journal of Medical Informatics, Vol. 76, 2007, No. 2-3, pp. 208–215.

[7] Payne, P. R.—Mendonça, E. A.—Johnson, S. B.—Starren, J. B.: Conceptual Knowledge Acquisition in Biomedicine: A Methodological Review. Journal of Biomedical Informatics, Vol. 40, 2007, No. 5, pp. 582–602.

[8] Villanueva-Rosales, N.—Dumontier, M.: yOWL: An Ontology-Driven Knowledge Base for Yeast Biologists. Journal of Biomedical Informatics, Vol. 41, 2008, No. 5, pp. 779–789.

[9] Kokossis, A.—Gao, E.—Kourakis, A.: On the Dynamic Management of Chemical Engineering Knowledge Using an Ontology-Based Approach. Computer Aided Chemical Engineering, Vol. 21, 2006, pp. 1997–2002.

[10] Morbach, J.—Yang, A.—Marquardt, W.: Ontocape – A Large-Scale Ontology for Chemical Process Engineering. Engineering Applications of Artificial Intelligence, Vol. 20, 2007, No. 2, pp. 147–161.

[11] Noy, N. F.—Sintek, M.—Decker, S.—Crubezy, M.—Fergerson, R.—Musen, M. A.: Creating Semantic Web Contents With Protege-2000. IEEE Intelligent Systems.

[12] Bechhofer, S.—Horrocks, I.—Goble, C.—Stevens, R.: Oiled: A Reasonable Ontology Editor for the Semantic Web. In: Proceedings of the KI 2001, Joint German/Austrian Conference on Artificial Intelligence, 2001.

[13] Sure, Y.—Erdmann, M.—Angele, J.—Staab, S.—Studer, R.—Wenke, D.: Ontoedit: Collaborative Ontology Development for the Semantic Web. In: Proceedings of the 2002 International Semantic Web Conference (ISWC), 2002.

[14] Kalyanpur, A.—Parsia, B.—Sirin, E.—Grau, B. C.—Hendler, J.: SWOOP: A Web Ontology Editing Browser. Journal of Web Semantics, Vol. 4, 2006, pp. 2, pp. 144–153.

Dosam Hwang received his Ph. D. degree in natural language processing at Kyoto University. In 1987, he was chosen the best researcher at Korea Institute of Science and Technology. Since 2005 he has been the Professor at Yeungnam University in Korea. Currently he is working on natural language processing, machine translation, ontology, semantic web, and information retrieval. He has served for a number of international conferences and in the technical committee for ISO.

In Keun Lee is currently a Ph. D. candidate in electrical engineering at Yeungnam University. He received his B. Sc. degree in metallurgical engineering from Yeungnam University in 2001 and M. Sc. degree in electrical engineering from Yeungnam University in 2004. His research interests include ontology, autonomous robots, image understanding, and intelligent systems.

Jason J. Jung has been a full-time lecturer at Yeungnam University, Korea, since September 2007. He was a postdoctoral researcher in INRIA Rhône-Alpes, France in 2006, and a visiting scientist in Fraunhofer Institute (FIRST) in Berlin, Germany in 2004. He received the B.Eng. in computer science and mechanical engineering from Inha University in 1999. He received M. Sc. and Ph. D. degrees in computer and information engineering from Inha University in 2002 and 2005, respectively. His research topics are knowledge engineering on social networks by using machine learning, semantic Web mining, and ambient intelligence.