Ontalk: Ontology-Based Personal Document Management System*

Hak-Lae Kim
Dankook University
Anseo-Dong
ChonAn, Chungnam, Korea
82-41-550-1812
hlkim@dku.edu

Hong-Gee Kim
Dankook University
Anseo-Dong
ChonAn, Chungnam, Korea
82-42-550-3363
hgkim@dku.edu

Kyung-Mo Park
KyungHee University
Giheung, Sochen-ri
Yongin, KyungGi, Korea
82-31-201-2979
saenim@khu.ac.kr

ABSTRACT
In this paper, we present our development of a document management and retrieval tool, which is named *Ontalk*. Our system provides a semi-automatic metadata generator and an ontology-based search engine for electronic documents. *Ontalk* can create or import various ontologies in RDFS or OWL for describing the metadata. Our system that is built upon .NET technology is easily communicated with or flexibly plugged into many different programs.

Categories and Subject Descriptors
I.1.7 [**Document and Text Processing**]: Document and Text Editing – *Document Management , Languages*

General Terms
Management, Documentation, Performance, Design, Experimentation, Languages, Theory.

Keywords
Ontology, Document Management, Knowledge Management, Inference etc.

1. INTRODUCTION
Managing electronic data becomes a more challenging task for end users as the personal data storage capacity increases. There are many kinds of applications or software components to manage files in a local computer, but it is very difficult to organize personal documents in a consistent way and to search expected ones in a precise way. When users store documents in their computers, they have to remember file names or locations to retrieve them. A file-searching tool such as *Windows Explorer* usually relies on information about the physical features of the file (i.e. format, file name, path, size etc). Although we remember the names and the paths of the files stored in our computer, it would be almost impossible to find the right ones without knowing their contents. Suppose that we try to find the telephone number of a reasonable Italian restaurant, we will have to refer to the yellow page rather than the white page since the restaurant names and addresses are not useful enough. The yellow page that contains richer metadata could provide richer and more useful information of the place we want to find.

* This study was supported by a grant of the Korea Health 21 R&D Project, Ministry of Health & Welfare, Republic of Korea(03-PJ1-PG10-51300-0003)

WWW 2004, May 17-22, 2004, New York, NY, USA.
ACM 1-58113-912-8/04/0005.

We have developed a system, named *Ontalk* that provides a semi-automatic metadata generator and an ontology-based search engine for electronic documents. The system supports management and retrieval mechanism focusing on the contents of electronic documents. *Ontalk* can create or import various ontologies in RDFS or OWL for describing the metadata. In the following sections we present a brief overview of the system architecture.

2. The System Architecture
Figure 1 shows the architecture of the *Ontalk* system. *Ontalk* consists of three major parts: User Interface Module, Storage Module, and Inference Module.

The Interface Module consists of five components: namely, the Ontology Browser, the Query Editor, the Ontology Creator, the Metadata Generator, and the Author Info Viewer. The Ontology Browser and the Ontology Creator are much simpler than a full-fledged ontology development tool such as Protégé. A user can import any external ontology created by any other ontology builder. The Metadata Generator provides a semi-automatic mechanism to create the metadata summarizing the electronic document in question that may be in any format including text, image, and multimedia data. Any metadata can be automatically extracted from file profiles such as the ones created by Windows File Property Descriptor if there exists such a thing and it is needed. Each individual file or a part of file can be semantically described in XML metadata, and the metadata is then defined in terms of an ontology. The Metadata Generator also provides a widget to flexibly create a metadata input template. The Storage Module is a repository that stores the metadata files and ontologies. The Inference Module based on Jena API provides an ontology-based search or inference mechanism for the most

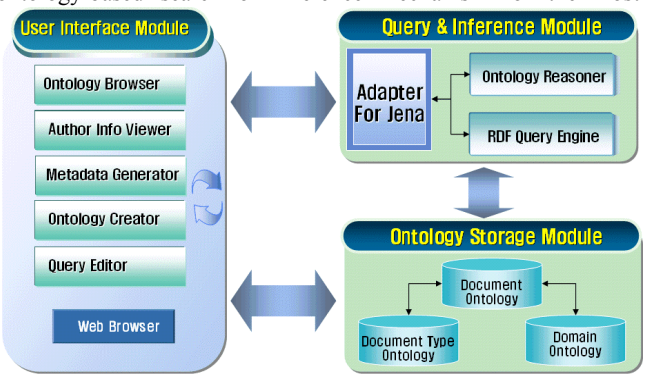

Figure 1. Ontalk Architecture

appropriate documents. Our system that is built upon .NET technology is easily communicated with or flexibly plugged into many different programs. We have developed an adapter to communicate between .NET components and Jena[1].

3. Document Management

Ontalk uses the three types of ontologies: namely, document schema ontology, document type ontology, and domain (or user) ontology. The document Schema and the Document Type ontologies are built in the system. On the other hand, the domain ontology can be created and modified by users. Figure 2 shows the relationship among ontologies. That is, an instance level metadata that is generated by a user is linked to the document type and the specific domain ontology.

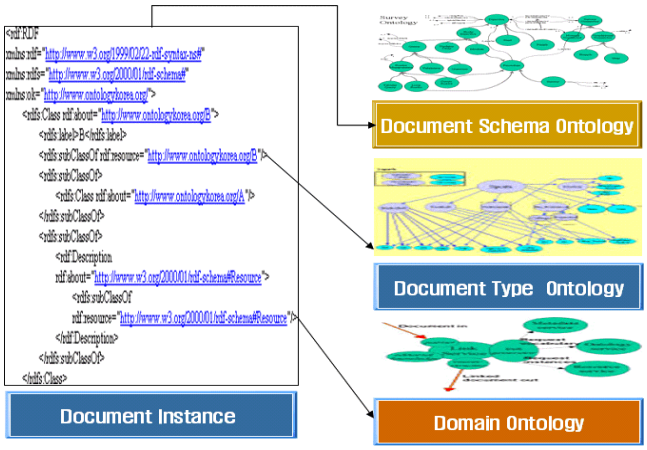

Figure 2. Ontologies of Ontalk

Figure 3 shows the Document Instance Template created by the Metadata Generator. It gathers metadata semi-automatically from a document that is located on the Web or in the local computer. It allows users to use other functional interface to define a metadata. For example, to input information about the author and contributor, we can use the Author info Viewer which defines name information regularly.

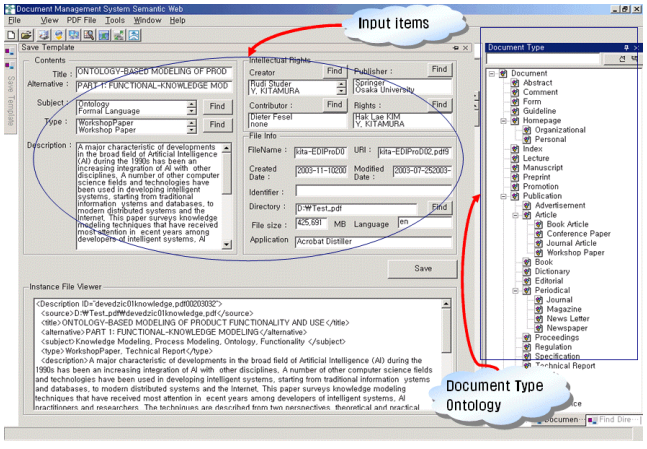

Figure 3. Metadata Generator

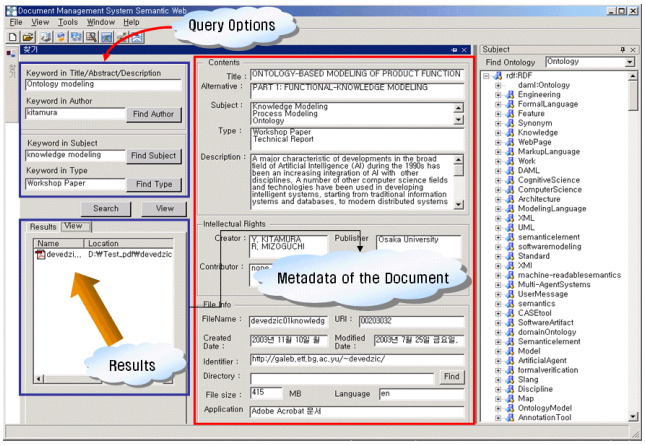

Figure 4. Results

4. Document Retrieval

After defining the ontologies and annotating the metadata of the document, we can query for information. We use the Jena toolkit to query in the KB. In addition to keyword-based retrieval mechanism based the Title and the Author, the system can use more powerful search option including the subject, the type etc. The subject option which describes [dc:subject] in Dublin Core Metadata, related to a specific domain ontology in our system. The Type option which describes [dc:type] in the document type ontology concerns a document type. The system can create and use any kinds of search constraints represented in the form of metadata in order to return more precise results. Using the ontology-based approach to document retrieval, we can make very rich expression for inference. For example, a query would be "Find all the documents whose types are workshop papers and whose subjects are markup languages." After finding the results, we can browser the results with the Listview of the system. In order to view the content of the document, the user can check the metadata before opening the file, which is illustrated in Figure 3.

5. Conclusions

Ontalk is a personalized document management system using ontology-based technologies. It provides a semi-automatic metadata generator and an ontology-based search engine for electronic documents.

6. REFERENCES

[1] HP Labs Semantic Web Research. "Jena-A Semantic Web Framework for Java". http://www.hpl.hp.com/seweb/, 2004.

[2] A.Seaborne. "RDQL:A Data Oriented Query Language for RDF Models". http://www-uk.hpl.hp.com/people/afs/RDQL/. 2001

[3] Ian Horrocks and Sergio Tessaris. "Querying the Semantic Web: a Formal Approach". The 1st International Semantic Web Conference (ISWC2002),. Sardinia, Italy, June 9-12, 2002

[4] D. McGuinness and F van Harmelen (eds)., "OWL Web Ontology Language Overview", http://www.w3.org/TR/2003/WD-owl-features-20030331/., 2003