Translating Documents into Semantic Documents using Semantic Web and Web2.0

Hak Lae Kim
Digital Enterprise Research Institute, National University of Ireland,Galway
IDA Business Park, Galway, Ireland
+353-91- 495016
haklae.kim@deri.org

Hong Gee Kim
Seoul National University
28-22 Yeonkun-dong
Jongro-gu
Seoul, Korea
+82-2-7707452
hgkim@snu.ac.kr

Jae Hwa Choi
Dankook University
29, Anseo-Dong
Chonan, Chungnam, Korea
+82-41-550-3368
jchoi@dankook.ac.kr

Stefan Decker
Digital Enterprise Research Institute, National University of Ireland,Galway
IDA Business Park, Galway, Ireland
+353-91- 495016
stefan.decker@deri.org

ABSTRACT

Managing metadata of documents is a difficult and slippery for desktop users. A wide variety of technologies have been applied for supporting requirements of metadata management, ranging from the acquisition, creation, maintenance, retrieval, reuse, and publishing of metadata.

We introduce essential concepts of a semantic document and implement the necessary functionality of metadata managing process. We also propose that three tasks are required to facilitate unambiguous representation of metadata in documents: using XMP to store metadata with the file itself, using ontologies to represent semantic concepts and using Social Web services to interact with web based resources. So our approach allows a user to interact and share the resources among a Desktop and Web more easily.

Categories and Subject Descriptors

I.7.1 [**Document and Text Editing**]

General Terms

Management, Documentation, Design, Reliability, Human Factors, Languages.

Keywords

Semantic Document, Semantic Desktop, Web2.0, Folksonomy, Semantic Web etc.

1. INTRODUCTION

Managing electronic documents in a Desktop is a more challenging task for end users [5]. There are many kinds of applications or software components to manage electronic documents in a Desktop, but it is very difficult to organize documents in a consistent way and to search expected ones in a precise way.

There have been many efforts [2, 3, 5, 6, 13, 19, and 23] to reduce the complexity of metadata operations by implementing automatic tools for acquisition, extraction, storage, and annotation. The *Social Semantic Desktop* [1] and *Web2.0* are also reliable technologies trying to promise solutions for metadata management.

The *Social Semantic Desktop* is a new computing paradigm that provides an advanced way to create, automate and structure information and "*the technology convergences including the social network and community services, P2P services*" [1, 3]. It could be provided for the transformation of a typical desktop system into a collaborative environment that supports both personal computing and information sharing via social and organizational channels [17]. There are several approaches in this direction such as Haystack[1], Gnowsis[2], IRIS[3] etc.

Web2.0 comprises technologies and services to enable users to collaborate and share social contents. From the technical point of view, it includes social software, content syndication, messaging protocol such as weblogs, wikis, podcasts, RSS feeds etc. Social softwares are not only focused on connecting people, but also on sharing data. Therefore, it plays an important role in building social networking on the web. There exist well-known Web2.0 sites like Flickr[4], del.icio.us[5], Technorati[6] and the majority of such sites are connecting people into communities creating networks of shared experience using folksonomy and RSS [10]. In general terms, a folksonomy represents the set of tags containing one or more keywords. Users create tags using their own knowledge then other people use same terms and the content is

SAAW'06, November 6, 2006, Athens, GA, USA

[1] http://haystack.lcs.mit.edu/

[2] http://nepomuk.semanticdesktop.org/xwiki/bin/Main1/

[3] http://www.openiris.org/

[4] http://www.flickr.com

[5] http://del.icio.us

[6] http://www.technorati.com

linked. Hence the Social Web Services contains all features of web services and social software through a folksonomy.

1.1 Problems

As illustrated by a Semantic documentation of Section 2, desktop environments have critical problems to manage [6]:

Heavyweight cognitive activity. The hierarchical file structure of desktop systems allows users to find the documents easily, but also reminds users of their respective task. There are, however, some critical limitations within the file structure for managing the information resources within a Desktop application. Users regardless of their behavior need to remember their document's name, the directory it was saved in, the saved time amongst other details. Because most activities are doing by human themselves this behavior requires heavyweight cognitive activity.

Multiple semantics. The hierarchy file system doesn't provide multiple semantics for a single directory. How could a user save a paper about a conference and a location? A user could create a "Conference_Location" or "ConferenceLocation" folder as its name. It is a slightly ambiguous approach and doesn't reflect multiple semantics correctly. In other words, a computer cannot process the inter-relationships between file names and directory names if their naming is different.

Poor updatability and interoperability. Compared with web content, Desktop content is difficult to modify without an owner's intervention. If the users spend a significant amount of time adding and/or modifying documents, the updatability of desktop content might be high. However, the majority of people don't spend their time adding additional information to the document. Also it is hard to share documents with other users despite P2P or instant messenger, both of which are supposed to provide file sharing services.

Editing problem. The metadata-oriented approaches provide enriched functionalities such as managing, searching and even sharing information in information systems. There exist a variety of metadata schemes as de facto standards such as RDF, Dublin core, vCard. But these approaches are not a panacea. The operations over metadata are complex and time-consuming. Moreover, a metadata is stored separately from the document and is connected by external references or links like XPointers. When a document are edited, deleted, or copied, however, it is the maintenance of the links that become a problem. This problem has been termed the editing problem by the Open Hypermedia community. A straightforward solution to *"editing problem"* [4] is to embed the metadata in the document itself.

1.2 Contributions

We present three contributions. (*i*) We propose the architecture and implement the tool to interact between a Dekstop and Web. It bootstraps the management of metadata and stimulates a user to participate in information management activity. (*ii*) We propose how desktop documents can be enriched using existed technologies like Semantic Web and Web2.0. Ontology and Folksonomy based metadata are important part of our system. A generated metadata by a user can be saved in document itself as XMP. It is possible to reuse and share for other users easily. (*iii*) We provide a user-friendly interface to extract or create metadata and efficient navigation through ontology and tags.

1.3 Outline of the paper

The main part of this paper is about how desktop systems can use resources to enrich metadata in document. So we decide to use the Social Semantic Desktop and Web2.0 technologies for making semantic documents in a Desktop. Especially we focus on PDF (Potable Document Format) which is the most well-known document format and on XMP which represents embedded metadata in PDF.

The remaining of this paper is structured as follows: Section 2 defines a Semantic Documentation and proposes the Semantic Document Model for our research. Section 3 then explains the design principles. Section 4 describes the system architecture and the metadata managing process for a semantic document. Finally, the paper concludes with Section 5.

2. Semantic Documentation

2.1 Semantic Document

Lawrence (Lawrence et al., 2004) defines that a semantic annotation is "*the process of mapping instance data*" to a semantic structure such as an ontology. A semantic document includes any information regarding the document and its relationship with other documents [27]. A semantic annotation of documents formally identifies concepts and relations between concepts in documents, and is intended primarily for use by machines [28]. Therefore, a semantic annotation is a key notion and a basic technology for the realization of a semantic annotation. It is augmentation of data to facilitate automatic recognition of the underlying semantic structure such as document structure (title, section, paragraph, etc.), linguistic structure (dependency, coordination, thematic role, conference, etc.), and so forth. Basically it is based on the semantically links between information stored within a document and the ontology. Ontologies are conceptualizations of a domain that typically are represented using domain vocabulary.

2.2 PDF and XMP

PDF is an open document format developed by Adobe. Most authors and publishers use it to store and to view documents. There are some advantages of using PDF format as the basis for semantic documents. PDF supports on-line viewing and printing while containing semantic information linked to the document itself [26] and provides extensible ways to add new information inside document using XMP.

In a nutshell, XMP (eXtensible Metadata Platform) is a format for embedding metadata in documents. It is a labeling technology that allows users to embed data about a file, known as metadata, into the file itself [10, 11, and 15]. It consists of a data model, a storage model, and schemas. A data model is a useful and flexible way of describing metadata in documents. It defines the kinds of

metadata values and concepts that can be represented. A storage model, as the implementation of the data model, includes the serialization of the metadata as a stream of XML and *XMP Packets*, a means of packaging the data in files [10]. Also schemas are predefined sets of metadata property definitions that are relevant for a wide range of applications, including all of Adobe's editing and publishing products, as well as for applications from a wide variety of vendors.

The specific serialization syntax is important. As long as the mapping to the data model is well defined, it is reasonably easy to convert between different ways to write the metadata [11]. XMP makes use of the Resource Description Framework (RDF), which is based on XML. By adopting the RDF standard, XMP benefits from the documentation, tools, and shared implementation experience that come with an open W3C standard [7-10].

2.3 Semantic Document Model

In this section, we describe the Semantic Document Model where users are managing metadata of their documents. Most users are doing their information management activity with both desktop and web applications; here, we describe a conceptual model for managing metadata using desktop resources and resources of social web sites. Firstly, the Semantic Document Model consists of a number of ontologies to define a metadata structure. Basically we propose the *document schema ontology* [7] for describing metadata of document. It can be locally maintained, interlinked and highly structured semantic information of each document. We propose the *document type ontology* to describe publication's type of research communities and relevant concepts - proceedings, thesis, article, technical reports etc. *Domain ontology* describes a certain subject which is closely related to a content of document. It might be extended by users as they need. Furthermore, users are able to get valuable piece of tags from various roots like the social web sites, user's blogs.

Figure 1 shows the Semantic Document Model which defines types of information. Basically it contains a physical information and basic content metadata of a document which supports by conventional file systems. Also a semantic document consists of social information and ontological information.

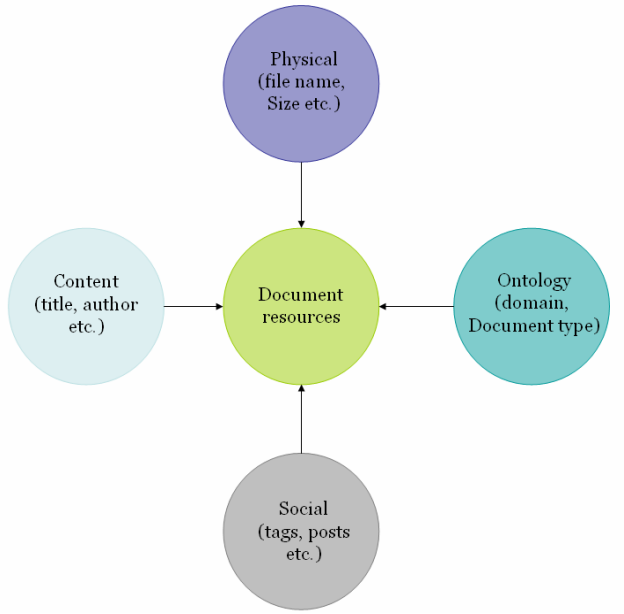

Figure 1 Semantic Document Model

3. Design Principles

In this section, we describe basic design principles, which are founded on the general problems sketched in the introduction above. Table 1 depicts simple processes for semantic document and requirements for solving problems. The key functions or process are extraction, creation, storage, index, and search. An overview of the matrix is given in Table 1. It shows functions are mainly used to answer challenges set forth in the introduction.

Table 1 Design Principles

Processes / Problems	Extraction	Creation	Storage & Index	Search
Heavyweight cognitive activity	X	X		X
Poor updatability	X		X	X
Multiple semantics		X		X
Editing problem			X	X

Extraction. In order to reduce heavyweight cognitive activities of a user, the extraction process allows semi-automatic or automatic methods. Basically, the results of the this process can involve with a metadata of documents, physical information such as file name, size, and date etc. In addition, this process should extract a metadata from weblogs or social web services.

[7] http://www.blogweb.co.kr/research/ontology

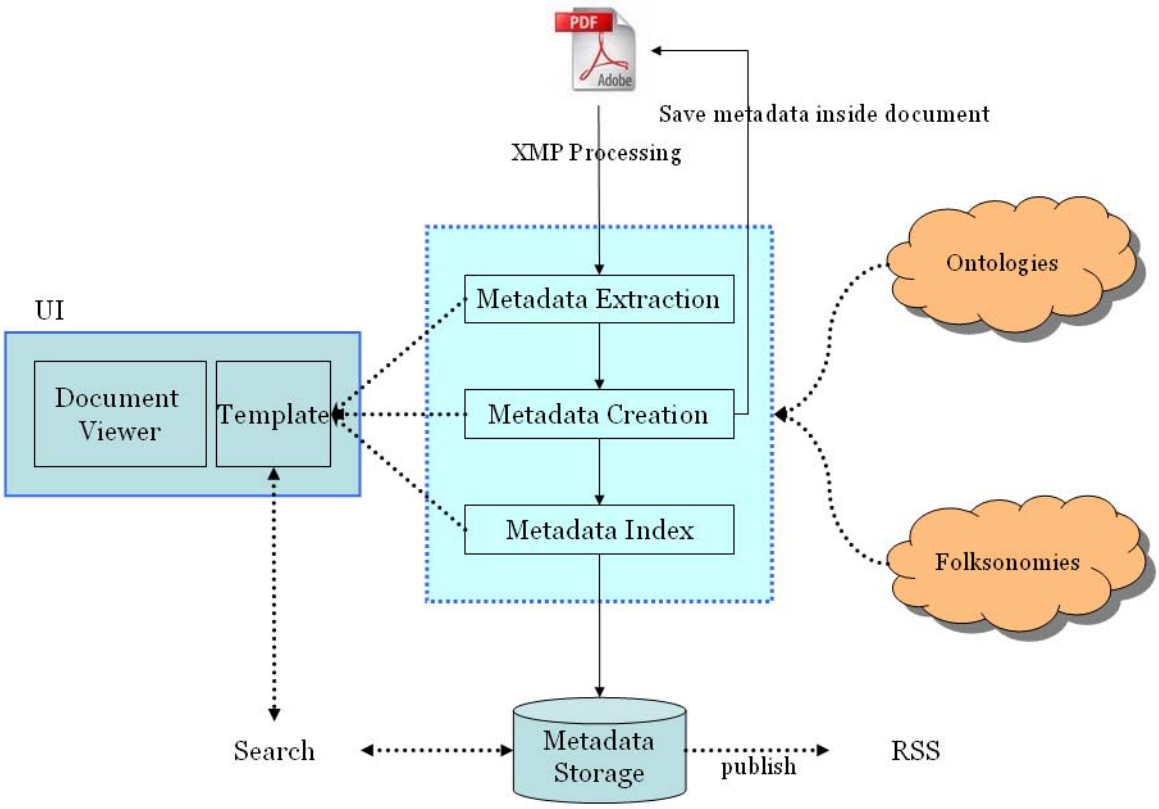

Figure 2 Architecture

Creation. To generate or modify metadata users can use various sources such as ontologies, tags, and even physical information. Users can define their own knowledge structures which are called domain ontology. Also tagging is one of new approaches to create metadata. In order to allow for the creating this metadata, the process must be supported by tools.

Storage & Index. A document metadata must be existed in the document itself to avoid the editing problem. And the metadata should have URIs of web resources. It becomes a starting point to connect on the Web.

Search. This process must cover ontology-based and tag-based search. The search results must be connected other resources as URIs. For example, a user identified the tags at a particular time, with URIs of web resources. But when they search, they can get unintended results with the tags because tags or folksonomies are self-evolutionary. It can be solved the problems of *Poor updatability and interoperability* in a Desktop**.**

4. Implementation

Figure 2 illustrates our architecture designed in response to the opportunities for functionality identified in the previous section. In this architecture, metadata of documents is created by two different sources, based on the ontologies and folksonomies. The idea behind the methods is based on the following observations. Ontologies are "intentional models" of information models of information contents with a well-defined logical basis which can be used for reasoning [13]. A folksonomy provides a shared meaning through collaborative work on the Web. Although ontology and folksonomy have different approaches to make meanings, they can both supplement each other in the process of creating metadata and searching it.

Basically metadata of a document is extracted from the document itself. The Metadata Extractor can parse and deliver metadata inside the document to the Metadata Explorer. Also users are able to get valuable piece of information from various roots like folksonomies, user's blogs, or even ontologies when they would create metadata. Then all kinds of metadata should be saved in certain PDF file itself as XMP.

Each document including metadata is built and is stored the index automatically. It allows user to search using the domain ontology or tags. Search results would contain relevant data such as raw file information, ontology concepts, and tags from embed metadata. If users want to see web resources with relevant results, they may be getting all lists of the terms from specific blogs or social web services sites.

In order to solve general problems and support the processes mentioned the introduction above, we provide core UIs such as the Metadata Explorer, Ontology Editor, and Tag Generator etc. tool support is essential component of the semantic document approach.

- *Metadata Extractor* : extract metadata from a document
- *Metadata Explorer*: view, create, and modify metadata

- *Ontology Editor*: view, edit an ontology
- *Tag Generator*: create, view tags
- *Search* : keyword, ontology based search

In the following subsection we explain the concrete realization and processes.

4.1 Metadata Extraction

Metadata Extraction is an internal process. Users do not need to know how it works since XMP is machine readable metadata. The XMP handler extracts a XMP metadata using Jena RDF API and display each items in the Metadata Explorer (see Figure 3).

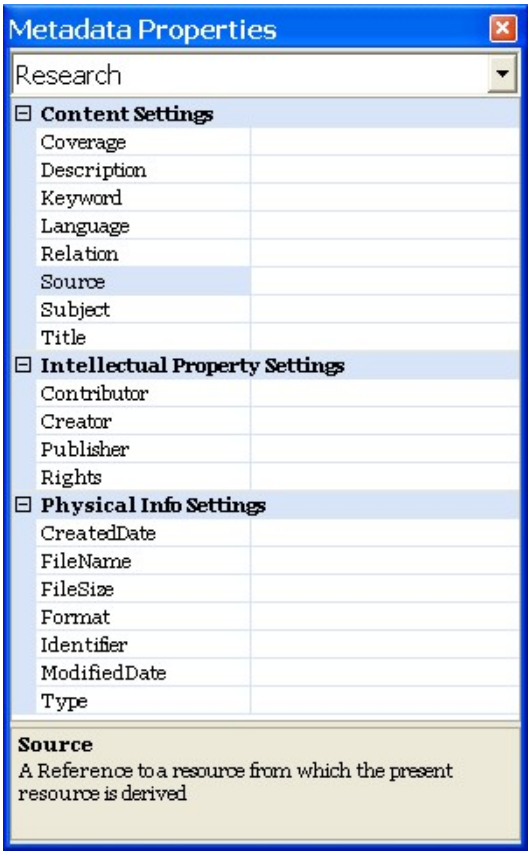

Figure 3 Metadata Explorer

The Metadata Extractor can automatically extract embedded metadata if documents have pieces of information and the Metadata Explorer shows the items of metadata. It allows users to add or modify metadata directly in the fields as it allows editing items. Unfortunately some items (subject, tags etc) should be added manually. In following section, we describe two kinds of a way to add metadata in document. Since it provides user-friendly interface, a user would be saved their time and effort to create metadata.

4.2 Metadata Creation

Insert ontology concepts. Users can define their own ontology using the Ontology Editor. It provides functionalities for editing and browsing ontology and allows users to define and update ontology in a tree structures. The ***Subject*** item which describes [dc:subject] in Dublin Core, related to a specific domain ontology in our system. The ***Type*** item which describes [dc:type] in the document type ontology concerns a document type. Users select a node to insert it into the ***subject or type*** item in the Metadata Explorer from the Ontology Editor.

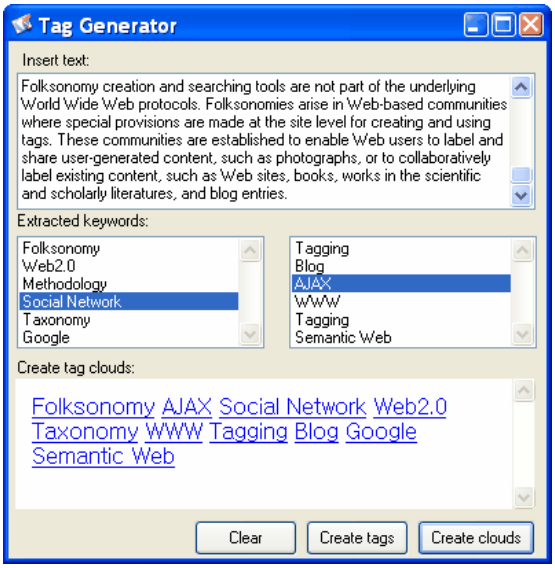

Figure 4 Tag Generator

Insert tags. To add certain tags we provide several functions. Users can add tags from social web services using the TagCloud[8] interface. It shows folksonomy from Flickr or Del.icio.us etc. In addition, if users want to create tags automatically, they would create tags using the Tag Generator (see Figure 4). It is based on the Yahoo's Content Analysis web service[9] which is a context extraction web service. This service allows retrieval of terms that were extracted from a given text [13]. Tags which users selected will be added in ***Keyword*** item in the Metadata Explorer.

After inserting relevant items, it can be saved in the file as well-defined data in RDF format. One of the main advantages of serializing XMP as RDF is that this has potential possibility for reaching ubiquity as the cross-platform container for machine readable/processible metadata [20].

Ontological concepts and tags can be assigned to a document; the document in desktop no longer has to be in a single folder. Eventually it can be solve the restriction of multiple semantics in desktop. In addition, the tags contain relevant URIs or feeds on the Web. It can be evolved itself without any human interruption. It means desktop documents can be evolved through connecting the Social Web services.

[8] http://www.tagcloud.com

[9] http://developer.yahoo.net/search/content/V1/

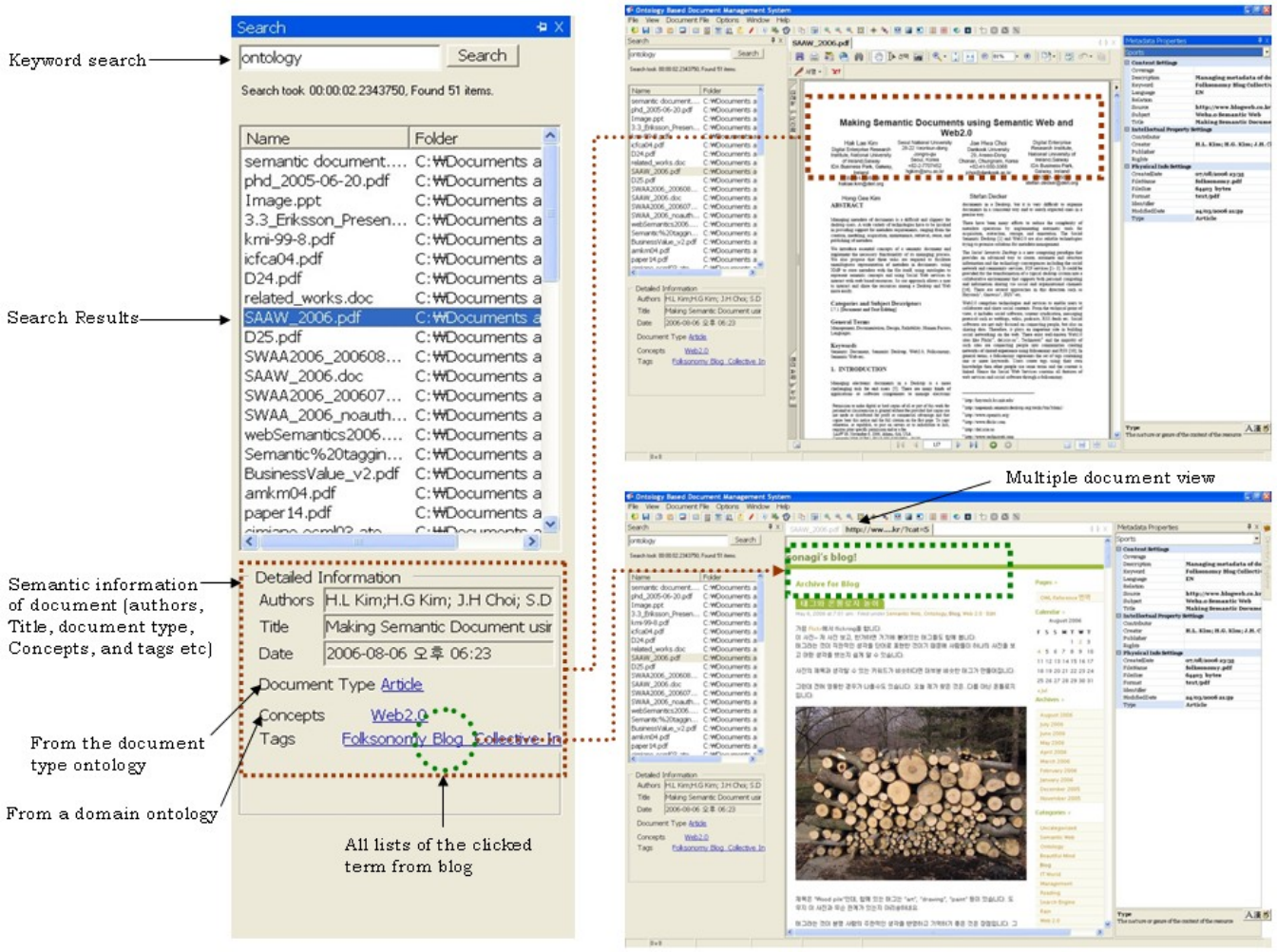

Figure 5 Unified Search View

4.3 Indexing and Search

We build an index using XMP which already embedded in PDF file. We use Jena[10] to parse the XMP data and Jakarta Lucene[11] to index metadata. This is the most popular document indexing and search library available for Java and .Net. Since Lucene by itself will accept and process only plain text, some kind of adapter must be used that can extract plain text from PDF files in order for those files' content to be added to a Lucene index. This process is done using the XMP Parser class module in Jena. With Jena/Jakarta Lucene user can select a folder they want to build an index. This is quiet simple. User clicks the Browser button, and then chooses the folder. But we don't provide multiple indexes in one computer at this moment. So if users want to make multiple one, they should select upper level folder.

Users may search for more specific information regarding the topics or keywords, but are not sure how to narrow their search. Although they are typing in several terms, they cannot sure results. Our tools are able to help users in narrowing down their search range using the Ontology Editor and to search related items using the results.

Ontology-Based Search. The search component executes a search across the 'subject', 'title', 'keyword' and 'description' metadata fields as well as the text of PDF files. If a user cannot find a start term, he or she can use the Ontology Editor. The search results display the 'file name', 'title', 'description', 'date', 'format' and 'weighted score' and 'format' metadata fields. The weighted score is a weighted primary according to the subject filed in the metadata. The Ontology Viewer is used for a refined searching. If user chooses several terms in the Ontology Editor,

[10] http://jena.sourceforge.net

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

then results change automatically. It allows user to combine any fields such as subject, title, description.

Tag-Based Search. This function gathers RSS feeds from a set of selected remote tags. When a user chooses a keyword in their results, it collects the related feeds with the selected keyword from the remote web blog. The data is collected simultaneously when the search executes. Currently we selected a list of RSS feeds consisting of several web blog sites. The tag-based search interacts with the information published in user's blog. It tries to enrich users' metadata with associated information in web.

Figure 5 shows the search results which includes file information, ontology, and folksonomy. That is, our tool provides unified search views. Firstly, a user can see physical information of files. Even though the Window Explorer already provides this function, it is useful because the Result View includes not only a file name, folder, but also content's title, keywords, concepts. Secondly, if a user want to see more detail metadata information, they click each list in results, and then it opens the Metadata Explorer. Finally, a user is able to reuse keywords, which attach raw files as metadata, of the clouds in blog. If a user wants to see blog entries with relevant results, she clicks the term of keywords in results and then she can get all list of the term – "clicked term".

5. Conclusions and Future Work

This paper describes a means for managing a semantic document by leveraging two kinds of metadata: ontology based and tag-based. In order to enable documents to be unambiguously used by human and machine, metadata should be represented with explicit part of documents. The *document schema ontology* contains ontological concepts as well as social collective tags. Furthermore metadata could be existed embedded object in the document rather than being separated with it. An embedding metadata could be stayed with file content itself regardless of moving, modifying the file. The documents would then be indexed and be searched by semantic tools. Hence making semantic documentation an explicit and embed part of the document makes the metadata managing process easier to support. We have focused mainly on PDF format. But we have plan to process different format like JPEG, GIF, Microsoft Office formats etc. Our future work plans include a more detailed focused on the mechanisms to interact and feedback between Desktop and Web. The approach, model, and techniques of this research will be explored in our future work.

6. ACKNOWLEDGMENTS

We also thank our colleague Dr. Handschuh for his continued guidance and his assistance with information for this paper.

7. REFERENCES

[1] Decker, S., Frank, M. The networked semantic desktop, *In: Workshop on application design, development and implementation issues in the semantic web*. 2004.

[2] D.Quan, D.Huynh, and D.R. Karger., Haystack: A Platform for Authoring End User Semantic Web Applications, *In International Semantic Web Conference 2003*, 2003

[3] Sauermann, L., The gnowsis semantic desktop for information integration, *In: 1st Workshop on Intelligent office appliances*, 2005

[4] Leslie. C, Timothy. M.B, and Arouna. W, The Case for Explicit Knowledge in Documents", DocEng'04, 2004.

[5] H.L. Kim, H.G. Kim, and K.M. Park, Ontalk:Ontology-Based Personal Document Management System, WWW2004.

[6] H.L. Kim, H.G. Kim, and Decker,S., Semantic Documentation using Semantic Web Technologies and Social Web Services, *In:Proc. International Conference on Next Generation Web Services Practices (NWeSP'06), 2006*

[7] Jenneke. F, Johan. P, Wray. B, Tag-Based Navigation for Peer-to-Peer Wikipedia, WWW2006, 2006.

[8] Adobe, XMP SDK Overview, 2001.

[9] Gray. K, A Manager's Introduction to Adobe eXtensible Metadata Platform, the Adobe XML Metadata Framework, Adobe Whitepaper, 2001

[10] Adobe, XMP Specification, 2005. available at: http://partners.adobe.com/public/developer/en/xmp/sdk/xmpspecification.pdf

[11] Alan. L, Duane. N, OpenDocument metadata and XMP, 2005, available at: http://www.oasis-open.org/archives/office/200512/msg00009.html

[12] Hopkins, I., Vassileva, J, Beyond keywords and hierarchies, .*Journal of Digital Information Management* 3 (2005) 139–145

[13] Stuckenschmidt, H., Harmelen F. V, Ontology-Based Metadata Generation from Semi-Structured Information, *In:Proc. 1st international conference on knowledge capture(K-CAP'01)*, 2001, pp 440-444.

[14] Kraft, R., Maghoul, F., Chang, C. C, Y!Q: Contextual Search at the Point of Inspiration, *In:Proc. CIKM'05* , 2005.

[15] Johnson, A, XMP Blaster: Embedding Metadata into Digital Photographs, http://www.mines.edu/Academic/courses/math_cs/macs370/FS2004/FinalReports/FinalWhite.pdf

[16] Kevin Broccoli, Improving Information Retrieval with Human Indexing, http://www.intranetjournal.com/features/humanindex-1.shtml

[17] Mander. R, Salomon. G, and Wong. Y.Y, A 'pile' metaphor for supporting casual organization of information, *In:Proc. Conf. on Hum. Factors in comp. sys.*, 1992, pp 627-634

[18] James. H, Abby. G, Why can't I manage academic papers like MP3s? The evolution and intent of Metadata standards, 2004

[19] Handschuh, S., Staab, S.: Authoring and Annotation of Web Pages in CREAM. In:Proceedings of the Eleventh International World Wide Web Conference, Honolulu, Hawaii, USA.2002.

[20] Tallis, M.: Semantic Word Processing for Content Authors. In: Proceedings of the Knowledge Markup & Semantic Annotation Workshop, Florida, USA. (2003) Part of the Second International Conference on Knowledge Capture, K-CAP 2003.

[21] Fillies, C., Wood-Albrecht, G., Weichardt, F.: A Pragmatic Application of the Semantic Web using SemTalk. In: Proceedings of the Eleventh International World Wide Web Conference, Honolulu, Hawaii, USA. (2002) 686-692

[22] Ontoprise GmbH: OntoOffice Tutorial. http://www.ontoprise.de/documents/tutorial ontooffice.pdf (2003)

[23] Carr, L., Miles-Board, T., Wills, G., Woukeu, A. and Hall, W. (2004) Towards a Knowledge-Aware Office Environment. In Proceedings of 5th International Conference on Practical Aspects of Knowledge Management (PAKM 2004) LNAI 3336, pp. 129-140, Vienna, Austria. Karagiannis, D. and Reimer, U., Eds.

[24] Martin, P & Eklund, P: Embedding Knowledge in Web Documents, In: Proceedings of the 8th Int. World Wide Web Conf. (WWW'8), Toronto, May 1999, 1403-1419

[25] Anita, D., W., Gerard, T.: The ABCDE Format: Enabling Semantic Conference Proceedings,

[26] Henrik Eriksson: A PDF Storage Backend for Protégé, http://protege.stanford.edu/conference/2006/submissions/abstracts/9.4_Protege-2006-Eriksson.pdf

[27] S. Staab, A. Maedche, and S. Handschuh.: An annotation framework for the semantic web. In Proceedings of the First Workshop on Multimedia Annotation, Tokyo, Japan, January 30-31, 2001.

[28] J. Heflin, J. Hendler and S. Luke: SHOE: A Knowledge Representation Language for Internet Applications, Technical Report CS-TR-4078 (UMIACS TR-99-71), 1999.

[29] Guoren, W., Bin, W., Donghong, H., and Baiyou, Q.: Design and Implementation of a Semantic Document Management System, Information Technology Journal 4 (1): 21-31, 2005

[30] Uren, Victoria; Cimiano, Philipp; Iria, Jose; Handschuh, Siegfried; Vargas-Vera, Maria; Motta, Enrico; Ciravegna, Fabio.; Semantic Annotation for Knowledge Management: Requirements and a Survey of the State of the Art, Journal of Web Semantics 4 (1):14-28, 2006

[31] Lawrence Reeve, Hyoil Han: Technical Report: Semantic Annotation Platforms, http://www.pages.drexel.edu/~lhr24/pubs/2004SemanticAnnotationTechnicalPaper.pdf, 2004.