Extending MS Office for sharing Document Content Units over the Semantic Web

Saša Nešić[1], Dragan Gašević[2], Mehdi Jazayeri[1]
Faculty of Informatics, University of Lugano, Switzerland
School of Computing and Information Systems, Athabasca University, Canada
sasa.nesic@lu.unisi.ch, dgasevic@sfu.ca, mehdi.jazayeri@unisi.ch

Abstract

In this paper, we present an extension to MS Office that enables users to search and retrieve document content units (e.g., paragraphs, images, tables, slides, etc.) from documents, which are stored on user's individual desktops organized in a peer-to-peer fashion. We first introduce the Semantic Document Model (SDM) that turns MS Office documents (i.e., MS Word and MS PowerPoint) into Semantic Web resources, making document content to be accessible and queryable as RDF data. Then we describe the developed tools, which extend Office applications with support for ontology-based, distributed search of semantic documents stored in local RDF repositories over Semantic Web protocols.

1. Introduction

The Semantic Web as an extension of the Web aims at providing a common framework to allow data and knowledge to be shared and reused across application, enterprise, and community boundaries [1]. One of the key steps in building the Semantic Web is local desktops organized as complete RDF (Resource Description Framework) and ontology-based environments. Local desktops should become the Semantic Web for a single user - the Semantic Desktop [2].

Digital desktop documents (e.g., Word, PowerPoint, PDF, etc.) render a significant part of the knowledge base stored on local desktops, and hence are important resources for the Semantic Web. However, traditional digital documents are characterized by heterogeneous, application-specific document formats, so that data exchange between applications is hardly possible. Moreover, a document packs a set of content units (CUs) together, establishing a context for them, but the CUs are not uniquely identified and can be hardly accessed and retrieved from outside the document. Finally, knowledge modeled by traditional digital documents can be used by humans (i.e., human understandable knowledge) but not by machines. In order to overcome these shortcomings and turn digital documents into Semantic Web resources, we have developed a new document model (Semantic Document Model– SDM) as an RDF and ontology-based solution. The model integrates existing digital documents as a human readable (HR) component with a newly generated, machine processable (MP) component. The MP component attempts to model the same knowledge as it is modeled in the HR component, but now that knowledge can be used and processed by machines. The model also enables the unique identification and semantic annotation of document CUs and their search and retrieval via Semantic Web protocols [4]. In order to illustrate benefits, which steam from SDM, we have developed a set of tools and integrated them into MS Office (i.e., MS Word and MS PowerPoint). In this way, we turned MS Office applications into Semantic Web applications that enable the exchange and sharing of document CUs (e.g., paragraphs, graphics, sections, tables, etc.) over the Semantic Web. In section 2, we first briefly explain the introduced model and then continue (section 3) with the description of the MS Office extension. Discussion of the related work (section 4) and final remarks (section 5) conclude the paper.

2. Semantic Document Model

So far, there have been many attempts to adapt digital documents for the needs of the Semantic Web, but these attempts have mainly focused on extending documents with meta-level descriptions (annotations), which are stored together with document content [3]. This has improved discoverability of document contents, but knowledge modeled within documents is not yet machine readable and understandable. The model that we have developed combines two components: HR and MP. The two components are stored separately without restricting each other, but well linked in order to ensure consistency and synchronous evolution of knowledge modeled within them. The model's main characteristics are as follows: 1) the use of existing document formats as a HR component; 2) the universal platform/tool independent MP component; 3) the unique identification of a document and its CUs; and 4) the semantic annotations are moved from the HR to the MP component.

Designing the MP component and establishing links between the two components was the main guiding

978-0-7695-3261-5/08 $25.00 © 2008 IEEE
DOI 10.1109/ICWE.2008.17

factor in developing the model. In general, we can define the MP component as a set of information atoms, linked by directed, typed relations. The information atom is represented with a set of conceptualized phenomena and has a link to the document CU (e.g., sentence, paragraph, image, and slide) from the HR component. In this case, the document CU can be seen as a HR description of the conceptualized phenomena from the information atom. We have used the Semantic Web technologies, in particular ontologies and RDF as the basis of the MP component.

The core of the solution is a document ontology [5] that captures the internal structure of document content by providing definitions of document content units (CU) as well as structural elements. The MP component is an RDF graph, whose nodes are instances of CUs defined by document ontology (e.g., *Paragraph*, *Table*, *Image*, and *Slide*) and to which concepts from the domain ontologies are linked. The concepts from the domain ontologies conceptualize the same phenomena as those described by the document CUs. Each node of the RDF graph has an URI, which is embedded in the document (i.e., its HR component) as a marker, thus uniquely identifying the CU and forming the link between the HR and MP components. The majority of existing document formats has some support for hidden bookmarks or simple types of annotation (e.g., PDF annotation element for PDF documents and custom XML markup and hidden bookmarks for MS Office documents) and we take advantage of this for embedding the markers.

The introduced semantic document model has effects on both humans and machines. Humans can continue to work with documents as before, but now they can use ontology-based software agents to locate document CUs based on knowledge modeled within them rather then relaying on simple content based search. Moreover, document CUs become uniquely identified, queryable resources, which humans can access and retrieved without affecting the document as a whole. On the other side, intelligent software agents (machines) can understand knowledge modeled within documents and can perform more of the tedious work involved in finding, sharing and combining knowledge on the Semantic Web.

3. MS Office Extension

The real use and success of the introduced model strongly depends on the cost of increased document management effort. Therefore, we argue for a semantic document management system that can be easily integrated into existing document authoring environments. In order to illustrate the benefits of the model, we have chosen MS Office document format (OpenXml) and extended MS Office (i.e., MS Word and MS PowerPoint) to support functionalities, which are enabled by the model. We have developed a set of modules that we have integrated into MS Office. The modules support two processes: 1) transformation of MS Office documents into semantic documents, that is, generation of the MP document components and their store in RDF repositories; and 2) search and retrieval of document CUs from distant documents repositories via Semantic Web protocols. The modules are seamlessly integrated into MS Office through the ***Transformer add-in*** and the ***Authoring Recommender add-in***. The GUI of the add-ins follows the design approach of MS Office GUI and does not alter user workflow. Now we explain both of the aforementioned processes. Further information and demos can be found on the project's web page [5].

3.1. Transformation Process

In order to have documents represented by the introduced model, we need to enable the transformation of regular MS Office documents. The transformation process is almost completely automated. Prior to the transformation, the user only needs to select a set of domain ontologies that conceptualize tentative phenomena described in the document to be transformed (e.g., active document in Word or PowerPoint) and then starts the transformation through the ***Transformer add-in***'s GUI. During the transformation, the add-in deploys four modules: 1) *core transformation module*, 2) *annotation module,* and 3) *indexing module.*

The ***core transformation module*** scans the structure of a document to be transformed, extracts and stores all media CUs (e.g., images, audios, and videos) into the document media repository, and generates the MP component (i.e., RDF representation of the document). For recognized CUs (e.g., paragraph, image, and table), the MP component contains instances of appropriate concepts from the document ontology (e.g., *Paragraph*, *Image*, and *Table*). The ontological instances are uniquely identified with URIs, which copies the module embeds into the source MS Office document as hidden-bookmarks. In this way the link between document CUs and their ontological representations is established (i.e., between the HR and MP components). Also, the names of the extracted media CUs encode the URIs of their ontological instances. Moreover, the module does ontology-based information extraction from the document CUs and identifies concepts from the set of the user selected, ontologies that conceptualize the same domains as those described by the CUs. Ontological concepts, which are found, are then related to the ontological representation of the CUs and the process of generating MP component is finished.

The ***annotation module*** does semantic annotation of CUs by relating annotations to their ontological instances within the MP component. The annotation process is fully automated. One part of the annotations is derived from the document's metadata and the formatting styles of the document's content. The other part comes from capturing the interaction between users and the CUs [5]. Every time the user visits, reuses or modifies a CU, information about that is automatically added to the CU (i.e., to its ontological instance).

The ***indexing module*** does text indexing for all textual data from the document. The document repository has a single index, which is updated every time a new document is transformed. Text indexing is included to support text-based search as a secondary type of search, which is used if the ontology-based search does not return any results.

After a successfully completed transformation, the MP component of the document is created and stored in the semantic document repository (i.e., RDF repository). The MS Office document (i.e., HR component) stays in the same location of the file system with embedded links to the MP component.

3.2. Document Content Units Sharing

In order to enable sharing of document CUs over the Semantic Web (Figure 1), the semantic document repository needs to be a part of the RDF repository, which supports remote SPARQL [4] queries of RDF data. To achieve this, we use the RDF repository of the NEPOMUK platform [6]. NEPOMUK (the Social Semantic Desktop) platform is made up by the user's individual desktops, which are organized in a peer-topeer (P2P) fashion. By integrating the semantic document repository in the NEPOMUK platform, semantic documents become part of a collaborative environment, which enables sharing and exchanging of document CUs across social and organizational relationships. In order to enable users to search the semantic document repositories of their 'friends' for document CUs while working in MS Office applications (i.e., MS Word and MS PowerPoint) we have developed the ***Authoring Recommender add-in*** (Figure 2). The add-in uses two modules: 1) a *search module* and 2) a *raning module*.

Through the GUI of the add-in, the user provides the ***search module*** with the set of necessary information (Figure 2a): 1) a set of ontologies that conceptualize the domain of interest; 2) a set of tentative terms; and 3) the type of the CU (e.g. paragraph, image, and audio). The module then searches the repository(s) of semantic documents for document CUs by combining ontology-based and content/text-based search. First, the module queries the set of specified ontologies for ontological concepts whose labels contain some of the specified terms. The retrieved set of ontological concepts is then combined with the specified CU type and internally transformed into a query in the SPARQL query language [4]. Before the execution of the query, the module needs to obtain information about available semantic documents repositories against which the query will be executed. This information is kept as a part of the user profile, which is formally described by a user-model ontology [5]. The user is a part of a network of people who want to share their documents and her/his profile keeps information about the user's 'friends' and their semantic document repositories. Therefore, the search module first queries the user profile to find out the list of the user's 'friends' and then executes the SPARQL query against their semantic document repositories (i.e., RDF repositories) using the SPARQL protocol over HTTP. If the user searches for the CU of the text media type, the query result contains a list of CUs together with their metadata sets. If the user searches for the CU of the image, audio, or video media type, the query result does not contain the CUs themselves but a list of CUs' URIs together with metadata sets. In the later case, the search module performs one step more to obtain the CUs. Based on the URIs found and the URLs of the media repositories, the module forms an URL for each CU from the query result and retrieves them over the FTP protocol.

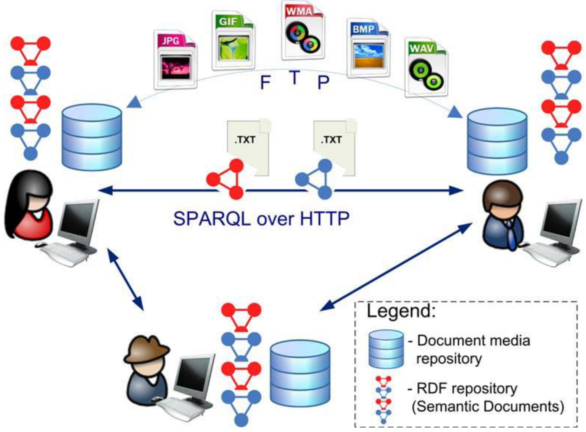
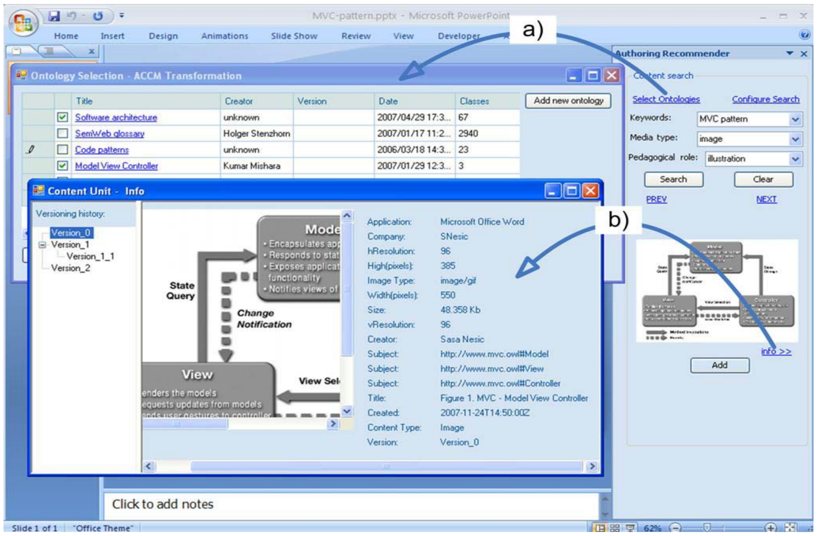

JPG
GIF
WMA
BMP
WAV
F T P
.TXT
.TXT
SPARQL over HTTP
Legend:
- Document media repository
- RDF repository (Semantic Documents)

Figure 1. Sharing of document CUs over SW

Once the *search module* completes the search, the add-in calls the ***ranking module***, which ranks the retrieved set of CUs. The ranking algorithm is based on the user's preferences regarding CUs (e.g., the number of CU's versions and occurrences in different documents), which are specified in the user's profile. For each preference we have developed weighting schema [7], on the basis of which the module first calculates the weight of each CU and than ranks them.

The add-in provides a preview of the retrieved, ranked set of CUs and their metadata (Figure 2b). For each CU the user can also browse the CU's versions from the versioning tree. Once the user selects CU to reuse by clicking on it, the add-in adds the CU to the current cursor position in the active document. Along with the addition of the CU to the document, the add-in also incorporates a hidden-bookmark with the CU's URI.



		Title	Creator	Version	Date	Classes
	☑	Software architecture	unknown		2007/04/29 17:3...	67
	□	SemWeb glossary	Holger Stenzhorn		2007/01/17 11:2...	2940
	□	Code patterns	unknown		2006/03/18 14:3...	23
	☑	Model View Controller	Kumar Mishara		2007/01/29 12:3...	3

Keywords: MVC pattern

Media type: image

Pedagogical role: illustration

Versioning history:
- Version_0
- Version_1
  - Version_1_1
- Version_2

Application:	Microsoft Office Word
Company:	SNesic
hResolution:	96
High(pixels):	385
Image Type:	image/gif
Width(pixels):	550
Size:	48.358 Kb
vResolution:	96
Creator:	Sasa Nesic
Subject:	http://www.mvc.owl#Model
Subject:	http://www.mvc.owl#View
Subject:	http://www.mvc.owl#Controller
Title:	Figure 1. MVC - Model View Controller
Created:	2007-11-24T14:50:00Z
Content Type:	Image
Version:	Version_0

Figure 2. Authoring recommender add-in a) Ontology selection b) Content unit preview

4. Related Work

There are many attempts to convert digital desktop documents to semantic documents, like PDFTab [2], and ActiveDoc [8]. However, most of them are focused only on the addition of meta-level descriptions to document content, which are expressed in machine-processable form. This enhances content discoverability, but the document content is not yet machine-processable and the access and retrieval of the smaller document content units is restricted by the specifics of the document format. Traditionally, document content units are usually reused manually by copy and pasting. ALOCoM [9] tries to automate this process by automatic decomposition of document content and storage of individual components, enriched with metadata, in a centralized repository. However, in this way document CUs become unassociated with source documents and their context-dependent semantics are lost. Also, over time centralized repositories can become too fragile and difficult to maintain. In our solution, we have decentralized repositories of semantic documents, which enable easy access and retrieval of document CUs via Semantic Web protocols [4].

5. Conclusions

The paper presents an extension to MS Office, which enables access and retrieval of document content units from distant document repositories. The basis for the solution is the Semantic Document Model introduced in this paper, which turns MS Office documents (e.g., Word and PowerPoint) into semantic documents with machine processable content and uniquely identified document content units (e.g., paragraphs, tables, images, slides, etc.). By using the developed tools, users can search semantic document repositories of their 'friends' by executing remote SPARQL queries over HTTP. Textual content units are retrieved as binary data encoded in RDF triples. For images, audios, and videos, the retrieved RDF triplets contain URIs for the discovered content units, and their transfer is done over FTP in an additional step.

Acknowledgement: Research reported in this paper has been partially financed by the European Commission in the NEPOMUK (IST-FP6-027705) project.

6. References

[1] Berners-Lee, T., Hendler, J., Lassila, O., "The Semantic Web", Scientific Am., 2001, pp. 34-43.

[2] Eriksson, H., "The semantic-document approach to combining documents and ontologies", *Int'l J. of Human-Computer Studies,* 65(7), 2007, pp. 642-639.

[3] Uren, V., et al., "Semantic annotation for knowledge management: Requirements and a survey of the state of the art", *J of Web Semantics,* 4(1), 2006, pp. 14-28.

[4] http://www.w3.org/TR/rdf-sparql-protocol/

[5] http://www.inf.unisi.ch/phd/nesic/sdms/

[6] http://nepomuk.semanticdesktop.org/

[7] Nešić, S., Gašević, D., Jazayeri, M., "An Ontology-Based Framework for Authoring Assisted by Recommendation", *In Proc. 7th ICALT Conf.*, 2007, pp. 227-231.

[8] Lanfranchi, V., et al., "Semantic Web-based document: editing and browsing in AktiveDoc", *In Proc. of the 2nd European Semantic Web Conf,*, 2005.

[9] Verbert, K. et al., "Ontology-based Learning Content Repurposing: The ALOCoM Framework," *Int'l Journal on E-Learning*, 5(1), 2006, pp. 67-74.