INTEGRATION OF ONTOLOGY RESOURCES INTO OPEN FORMAT DOCUMENTS FOR SEMANTIC INDEXING

Viacheslav Lanin

Abstract: *The article describes the development of a software library for ontological metadata inclusion into modern office documents formats. The model of the document used for indexing its content by ontology concepts is given. Existing projects addressed for similar problems are overviewed.*

Keywords: *ontology; semantic indexing, document formats.*

ACM Classification Keywords: *I.2 Artificial Intelligence: I.2.11 Distributed Artificial Intelligence; I.7 Document and Text Processing: I.7.2 Document Preparation; I.7.3 Index Generation.*

Introduction

In modern information systems (IS) there is a shift from the processing of structured data to unstructured data handling. For definiteness, we mean that unstructured data is the traditional electronic documents in different formats. This trend is noticeable in the corporate sector and among private users. Specialized software and formats for storing documents were developed and used throughout the history of information technologies for the processing of documents. Nowadays new classes of systems (social networks, corporate portals, wiki-resources, etc.) become the only important part of the information space. The key element of those classes is the concept of "content", which can be generalized to the "electronic paper".

Across-the-board applicable technology WYSYWIG becomes a "time bomb" for electronic documents. Most of modern technologies that used for working with documents (text editors, language HTML) focus on organization of convenient work with information for the person because often ways to work with electronic information just copy the methods of work with a "paper" information. Text editor contains wide opportunities for text formatting (presentation in human readable format), but there are practically no opportunities for the transfer of the semantic content of the text, i.e. there are no tools for semantic indexing. Automatic intelligent processing of text is extremely difficult, because we usually have to deal with the "document for the people" and not to "document for the person and the system".

The modern approach to the definition of an electronic document requires metadata that describes the structure and semantics of the data presented in the document. Due to this approach, the processing of electronic documents can be organized in a qualitatively different level, since it is possible to

automatically intelligent analysis of information. This concept is laid in the project Semantic Web, but the status of the "Semantic Web" for a variety of reasons is still far from implementation. However, the ideas of the Semantic Web [Berners-Lee, 2001] can be implemented within a single information system due to the smaller scale of its domain. Currently, the data required for processing the documents dispersed (stored in the document as well as in databases of IS for processing documents), and is specific to each of the tasks accomplished during the life cycle of the document in the IS. Therefore, it is necessary to use a single mechanism to provide information about the document. Another solution is the ontological resource that describes the various aspects of electronic documents that exist throughout its life cycle. This resource can be the basis for a wide range of tasks associated with the processing of electronic documents in the IS.

For the complex decision of tasks is necessary to develop a model of the electronic document that allows to include the meta and ontological resource (the basis for semantic indexing document contents). Also it's needed to develop technology for the introduction of metadata in the document and to propose a mechanism for processing documents. This paper is devoted to discussion of possible approaches to solving the above problems.

Document Model

The electronic document is a set of structural elements called fragments in this paper. For example, they are table, header, details form, etc. Thus the document can be represented by four species:

$$d = (S(F, R), C, o, M).$$

Here *S (F, R)* is oriented hypergraph. Nodes of that hypergraph are compared elements of *F* (*F* is a set of document fragments, and *R* is the set of edges of corresponding relations between fragments); elements of *C* represent the information content of the document (its contents); o is document ontology, *M* is mapping of *F* on the ontology's concepts. Let us examine the described components.

Hypergraph *S (F, R)* defines the relationship between the portions of the document. Direction of graph is needed, for example, to keep track of links "part-whole" between fragments. The nodes belonging to the edge are numbered that allows to set the order of the fragments in the text. Obviously the edge that includes all nodes corresponds to the document entirely.

There two types of fragments. The first type is basic simple fragments that are indivisible elements, such as the title or date of the document creating. The second type is compound fragments that contain other fragments.

Formally define fragment as a pair of the form:

$$f = (stat, inf),\ inf = \begin{bmatrix} F^*, F^* \subseteq F; \\ c, c \in C \end{bmatrix}.$$

There *stat* is a static part of the fragment (it can be represented as a text, images, links, any special symbol. In addition, there information for representing the fragment may be contained); *inf* is a part of fragment that indicates the location for placement of element content c ($c \in C$), or contains a set of fragments F^*.

Traditionally, common graphs are used for document presentation. Usually trees are used (e.g., format XML). Tree structure of description is much easier to work with the document but at the same time, it has significant limitations. Selecting of hypergraph to represent document structure is substantiated of possibility of hypergraphs to present arbitrary connections between fragments of documents and their sets.

In the above notation, the document template can be defined as $t = (S(F, R), C_0)$, where C_0 is the *primary content* (for instance, standard headers that are included in the template, etc.).

In view of the specificity of solved problems in this paper, we specify the notion of ontology:

$$o = (C, R, A),$$

where C is a set of ontology concepts, R is the set of relations between concepts, A is a set of axioms, that are determined on ontology. Both classes and instances of these classes can be concepts. Axioms are used to set limits and rules that cannot be expressed in terms of the relationship.

For documents processing it's necessary to implement the operation of allocation an arbitrary part of the document (let us call it *operation of range getting*). The input parameter of that operation is arbitrary set of nodes, and the result is the subgraph generated by this set of nodes. *Operation of decoding* is "imposition" of the structure on the fragment (node of graph). In the majority of applications *visual layout of the document* and its presentation in a certain format are very important, so the *operation of document presentation in specific format* is necessary. This operation represents the function that sets a mapping between document fragments and some set of formats, the elements of which set the rules of fragments displaying. *The search operation* is applied to the various components of the document: the structure, content and presentation, and the result of the operation will be parts of the document matching search criteria.

Project Semantic Assistants

Semantic Assistants is an open source research project [Witte, 2008] developed by Canadian laboratory Semantic Software Lab. Semantic Assistants helps users to extract, analyze and development of content providing contextual services of NLP (Natural Language Processing). It directly integrates with desktop applications (word processors, email clients, web browsers), web information systems (e.g., wiki) and mobile applications based on Android. Semantic Assistants has an open service-oriented architecture and uses OWL ontologies Semantic Web.

Semantic Assistants architecture consists of four levels. On the first level, there are the client applications. On the second level, there are Web services and NLP Service Connector, which now wraps GATE framework for NLP and it is responsible for communication with customers, read requests, and the creating of the responses. The third level is NLP subsystem that is responsible for extracting, compiling and indexing of information as well as search. The fourth level is resource. It contains all the necessary external documents to that subsystem NLP should have access.

This work is of interest claiming a large number of supported client applications and offering good architecture. But at the moment the project is under construction and only three clients are implemented. Only one of them is word processor OpenOffice.org Writer. For introduction of semantic information in ODF documents Semantic Assistants does not use all the possibilities provided by the specification of ODF 1.2 and uses peer review mechanism adding to the document notes. Thereby it keeps the information in unstructured form and accessible for editing by the user, which is not always convenient.

Word Add-in for Ontology Recognition

Word Add-in for Ontology Recognition (Word Add-in) [Fink, 2010] is tool for the manual annotation of documents in Microsoft Word. Word Add-in is an application layer add-in for Microsoft Office and it is built on the .NET platform with using of VSTO technology. Word Add-in is an open source project that allows adapting it easily to needs of any interested user.

Using the Word Add-in begins with the selection of the base ontology. It's an electronic catalog that contains ontology related to one subject domain. The user can select one of the ontology of the database and then starts working with Word Add-in in the background to analyze the input text. If the word matches one of the selected ontology's concepts it will be specially marked (smart tags or custom actions). When the smart tag is activated or custom actions are selected in the menu there is a special context menu with which this concept can be viewed in browser of ontologies.

One of the main problems of Word Add-in is synonymous. It is also one of the problems is that a word may correspond to several concepts of different ontologies. In this case, the user has to select one of

ontology the most satisfying sense of the text.

In spite of the above-mentioned disadvantages, Word Add-in is a completely finished product. Its main advantage is the high level of integration with one of the most popular office suite Microsoft Office, which allows using it of wide range of users and does not have specific requirements for their preparation.

An Infrastructure for Managing Semantic Documents

Infrastructure for Managing Semantic Documents (ISDM) is specialized industrial product [Lucas, 2010]. Main functions of Infrastructure for Managing Semantic Documents (ISDM) are:

- semi-automatic annotation of electronic documents based on ontologies using markup templates;
- version control of electronic documents;
- semantic search;
- notification of changes.

ISDM consists of two main modules. The first is a semantic document repository (Semantic Document Repository – SDR) for storing electronic documents. The second module is so-called *"main module*" which in turn also has a complicated structure and can be divided into several sub-modules. They are module of semantic markup (Semantic Annotation Module – SAM), data extraction module and version control (Data Extraction and Versioning Module – DEVM), Search module (Search and Traceability Interface Module – STIM).

Semantic markup Module (SAM) allows you to add metadata to the corresponding subject ontology electronic document. Version ISDM described in provides a single electronic document format ODF 1.0. This format version is still lacking convenient and flexible metadata model and so the authors were forced to use the most appropriate means provided by the format. Instead of using the manual annotation of documents an approach is promoted based on the use of templates that allows reusing metadata.

Metadata is used to represent the so-called "instructions". They are instance and property. To specify ontologies that are used in the annotation another hidden field with the name «Ontologies» is used in the sense of that, the URL of the ontology is indicated.

Although this project is still relevant to this day, its main part, namely the mechanism of semantic markup is significantly out of date, because it is designed in accordance with ODF 1.0, while the new ODF 1.2 specification provides substantial tools to add metadata to ODF documents.

The architecture of the OfficeMetadataLib component

In this section, we will describe the requirements for software library OfficeMetadataLib and will show the architecture designed to solve the problem.

The OfficeMetadataLib component should be implemented as a software library that provides a set of functions:

- creating new and opening existing textual electronic document of Office Open XML and OpenDocument formats;
- providing access (control) of textual content of documents;
- providing access (control) of preinstalled and user metadata documents;
- inculcation of ontologies in OWL format to the document metadata and providing access to them (management);
- automated search and binding of the document's fragments of text with ontology concepts;
- possibility of algorithm's expanding and replacement of implemented basic search algorithms and lemmatization.

The OfficeMetadataLib software library should have a modular architecture (schematically shown in Fig. 1) for providing unified access to the electronic document format Office Open XML and OpenDocument, and the possibility of expanding the basic search algorithms and lemmatization.

DocumentModel describes the generalized model of textual office document that consists of two levels:

- ContentModel is model of document content.
- MetadataModel is model of document metadata.

This model should be designed in accordance with ISO/IEC 29500 standard that is described in [ISO/IEC 29500-1; ISO/IEC 29500-2; Open, 2011] and OASIS ODF 1.2 specification.

LemmatiserModel describes a generalized model of lemmatizer.

SearchModel describes a generalized model of search engine

OOXMLPlugin and *ODFPlugin* implement a generalized model of document in accordance with specifics of Office Open XML and OpenDocument formats. Selection of the feature's implementation for each document format into a single plug-in allows to refine and modify the code for each plugin individually (e.g., in case of changing the specification of document format) without changing the overall model and without touching the source code of the main library and other plug-ins.

LemmatiserPlugin is a concrete implementation of lemmatizer.

Selection of lemmatizer implementation as a plugin will allow connecting to the library third-party lemmatizers that implement appropriate interfaces.

BaseSearchPlugin implements a basic search engine. Like a lemmatizer it can be changed by third-party developers.

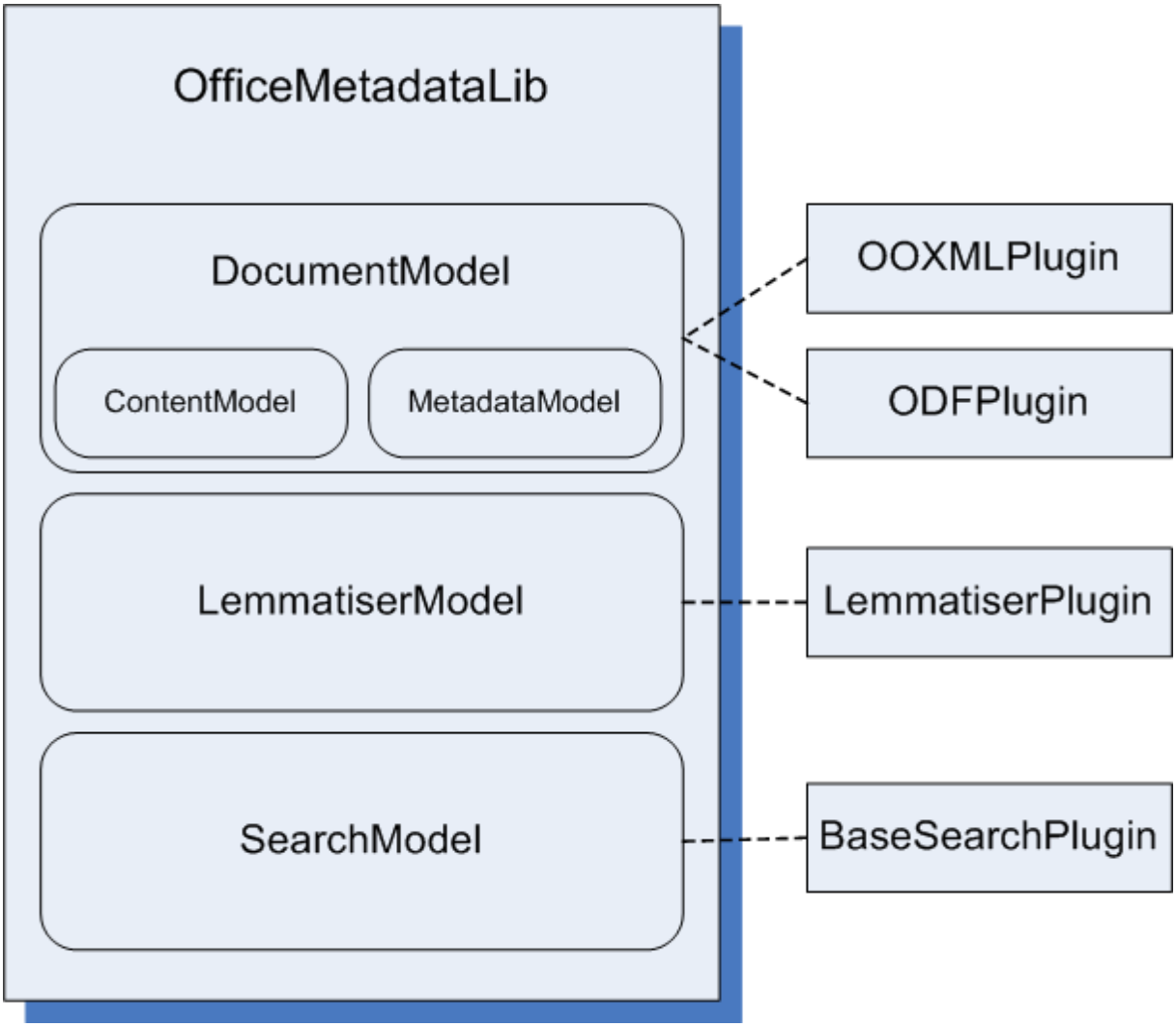

Fig. 1. The architecture of OfficeMetadataLib

Conclusion

In this work a software library providing unified access (control) to metadata of office document of Office Open XML format and OpenDocument format was developed. The main component of the library is *OfficeMetadataLib.DocumentModel*. It is a model of an electronic document and metadata based on the models of ISO / IEC 29500 (Office Open XML) and OASIS ODF 1.2. This model adequately reflects the characteristics of both formats and allows working with documents that use these formats in unified way. It is also worth noting that despite the fact that *OfficeMetadataLib.DocumentModel* was originally

designed to work with documents in the Office Open XML Formats and the OpenDocument (thanks to its flexible structure) there is a theoretical possibility of the use of the library for work with other document formats.

OfficeMetadataLib.DocumentModel describes document model and its metadata. Software implementation of model's transformation functions into document of specific format is contained in a special plugins that use special API for this (for example, Open XML SDK, OpenOffice.org SDK). Using of an approach based on the plugins allows eliminating the need for self-realization of all the features of the work with the above formats and allows using existing software solutions. Also worth noting that the use of plugins provides a high degree of flexibility and extensibility. In the case of obsolescence of any library or the appearance of a new more user-friendly library, it is possible to simply replace or add a plugin without changing existing code model.

The use of a unified approach to the development of a model to work with electronic documents is led to the fact that there is no possibility to use some features of formats.

In the future versions of the library it is planned to implement an interface for executing SPARQL queries to metadata document.

Bibliography

[Berners-Lee, 2001] Berners-Lee T., Hendler J., Lassila O. The Semantic Web. In: Scientific American (May 2001). Pp. 28-37.

[Bakalov,] Bakalov F., Sateli B., Witte R., Meurs M.-J., Komg-Ries B. Natural Language Processing for Semantic Assistance in Web Portals. In: IEEE Sixth International Conference on Semantic Computing (ICSC 2012), 2012. Pp. 67–74.

[Witte, 2008] Witte R., Gitzinger T. Semantic Assistants – User-Centric Natural Language Processing Services for Desktop Clients. In: 3rd Asian Semantic Web Conference (ASWC 2008), ser. LNCS, vol. 5367. Bangkok, Thailand: Springer, 2008, p. 360–374. [Online]. Available: http://rene-witte.net/semantic-assistants-aswc08.

[Fink, 2010] Fink JL, Fernicola P, Chandran R, et al. Word add-in for ontology recognition: semantic enrichment of scientific literature. BMC Bioinformatics. 2010;11:103. doi:10.1186/1471-2105-11-103.

[Lucas, 2010] Lucas de Oliveira Arantes, Ricardo de Almeida Falbo. An Infrastructure for Managing Documents. In: 14th IEEE International Enterprise Distributed Object Computing Conference Workshops. 2010.

[ISO/IEC 29500-1] ISO/IEC 29500-1 Third edition, 2012-09-01. Information technology – Document description and processing languages – Office Open XML File Formats. Part 1: Fundamentals and Markup Language Reference.

[ISO/IEC 29500-2] ISO/IEC 29500-2 Third edition, 2012-09-01. Information technology – Document description and processing languages – Office Open XML File Formats. Part 2: Open Packaging Conventions.

[Open, 2011] Open Document Format for Office Applications (OpenDocument) Version 1.2 Part 1: OpenDocument Schema 29 September 2011.

[Lanin, 2014] *Lanin V., Sokolov G.* Using multidimensional ontology of electronic document for solving semantic indexing problem. In: Proceedings of the 8th Spring/Summer Young Researchers' Colloquium on Software Engineering (SYRCoSE 2014). M. : ISP RAS, 2014. Pp. 166–169.

Authors' Information

Viacheslav Lanin *– National Research University Higher School of Economics, Department of Business Informatics; senior teacher; Perm, 614070, Studencheskaya st., 38; e-mail: lanin@perm.ru, vlanin@hse.ru.*

Major Fields of Scientific Research: Intelligent agents, Ontologies, Document processing.