MULTIDIMENSIONAL ONTOLOGY OF ELECTRONIC DOCUMENT AS A BASE OF INFORMATION SYSTEM

Viacheslav Lanin

Abstract: *This work presents an approach based on unstructured information retrieving from electronic documents to design and organize information system functioning. Document processing is based on semantic indexing and additional meta-information including. Document model allows formalize intelligent document processing algorithms and integrate documents containing ontological resources. Semantic indexing is based on multidimensional ontology describing document semantics and structure. To process documents agent-based approach allowing to resolve task of including business logic into documents is supposed. A systems built on such principles will be more flexible, intelligent and adjustable to changing environment. The proposed approach lets to solve a wide range of tasks, assuming usage of electronic documents at all lifecycle steps, which in turn allows implement document-oriented paradigm of information system life cycle maintenance.*

Keywords: *ontology, electronic document, document model, information system, semantic indexing, intellectual agents.*

ACM Classification Keywords: *I. Computing Methodologies. I.2 ARTIFICIAL INTELLIGENCE: I.2.11 Distributed Artificial Intelligence – Intelligent agents; Multiagent systems. I.7 DOCUMENT AND TEXT PROCESSING: I.7.2 Document Preparation – Index generation.*

Introduction

There is a tendency in modern information systems to handle documents, i.e. unstructured data rather than structured ones. New system categories, e.g. social networks, enterprise information portals or wiki-resources, have become a pivotal part of modern information space. «Content», or «electronic document» as more general term, is a key component of such systems.

Nowadays it is supposed that an electronic document has metadata describing data structure and data semantics apart from the content itself. Due to this approach, electronic document processing can be organized in a brand new way, because fully-automated intelligent information analysis can be applied. One of the base components of the Semantic Web [Segaran, 2009] was developed using such

approach, nonetheless this project in its current state is far from being fully implemented. However «Semantic Web» ideas can be realized within a framework of a single information system because of less scaling domains of interest. Now data needed to process documents are widely diluted, that means it is kept in documents itself as well as in databases of information systems created to handle that documents, and it appropriates only to a range of specific tasks being solved during the document life cycle. That is why there is a necessity to apply a uniform mechanism used to represent document information. One of the possible solutions is an ontological resource describing different aspects of an electronic document at different stages of its life cycle. This resource can be applied to resolve a variety of tasks connected with processing electronic documents by the means of information systems.

Document Model

To find a solution means to develop a model of electronic document, including metainformation, an ontological resource, which is a fundament for document content semantic indexing [Lukashevich, 2003], and document processing mechanism.

An electronic document is a set of structure elements further called fragments in the article. Most obvious examples of fragments are tables, headers, disposition form requisites etc. An electronic document can be represented as a set of four:

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

$S(F, R)$ – oriented hypergraph, which nodes (set F – a set of document fragments and edges set R – a set of relations between fragments), set C – document content, o – document ontology, M – mapping of set F into the o ontology concepts.

Hypergraph $S(F, R)$ assigns relations among documents fragments. Graph directivity is required to monitor such relations as *«is-a-part»* among fragments. Nodes are numbered, so it is clear how to determine a right order of fragments. Obviously, a graph edge including all nodes correspond to a whole document.

There can be two types of fragments: *simple fragments* are primary atomic elements like header or creation date and *compound fragments* consisting of simple ones. Formally, *fragment* is a set of two: *stat* – static fragment part including text, images, references, special symbols and some information

required for fragment representation, *inf* – fragment part indicating its content place or a set of fragments.

Conventional document representation takes advantage of usual graphs, trees, for example, the XML tree-like description structure considerably facilitates document processing, but at the same time, it implies many notable restrictions. Hypergraph allows store arbitrary relations among fragments and sets of fragments.

Using notion described above, document template can be defined as $t = (S(F, R), C_0)$ where C_0 – primary content (standard template headers etc.).

To give the particular characteristics of problems solved here, let us make more specific o ontology definition:

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

where C – a set of ontology concepts, R – a set of relations among concepts, A – a set of axiomatic statements made upon this ontology.

Concepts can include both classes and its instances; axiomatic statements are used to determine rules and restrictions, which can not be expressed by means of relations.

To process documents it is necessary to implement an operation of an arbitrary part of a document detaching (range extracting operation), a set of graph nodes will be an input to this operation, and a subgraph induced by this set of nodes will be an output. Decipher operation is a mapping structure onto a fragment (graph node). Apart from document structure and content, visual and format document representation take an integral part, that is why, operation of a format document representation, i.e. a function determining correspondence among document fragments and a set of other fragments assigning representation rules is needed. The search operation is applied to different document elements: structure, content and representation, and search result will be document fragments relevant to search criteria.

Multidimensional Ontology of Electronic Documents

It is required to have consolidated knowledge about document structure and content (electronic document format and type, its structure) to resolve document processing tasks. All of these three aspects are presented in multidimensional ontology, but concepts from different aspects are

interconnected. As a result, a single ontology of electronic documents is created. This resource has to maintain extension and enhancement configurable capabilities to solve newly appeared tasks at all life cycle stages of electronic documents.

Document Processing Logic

A problem of including data processing logic into electronic documents is being solved for many years (including commercial and research projects being implemented). There are commercial solutions such as office programming tools, macro programming languages; an example of context-based logic including are form filling automation tools (InfoPath, Google).

One of the main functions of all information systems is electronic document processing, however document is often considered just as «information containers» for users. Yet, if electronic documents are somehow «active», a range of conventions tasks for information systems will be solved more effectively. Document activation problem is supposed to be solved under the agent-based approach. Specific intelligent agents representing so-called «document interests» can be used to make documents more «dynamic». This approach allows not only make the intensity of tasks, being solved at all information system life cycle stages, lower, but also proposes new capabilities for developers and users.

The main idea of the approach supposes that each document has its intelligent agent containing all information about it and able to «Be of its interest» when solving diverse tasks. To implement intelligent document management it is necessary for an agent to have knowledge on document semantics and properly interpret document content. This in turn requires resolving semantic indexing task. Within the framework of the described approach basic domain knowledge is supposed to be stored in the ontology and it is accessible by the agents and can be interpreted by them.

Document life cycle maintenance

The proposed approach assumes to automate document life cycle in different information systems as well as to lower the labour intensity of domain analysis and domain changes during designing and maintaining information systems.

Based on semantic indexing and intelligent analysis of documents, a list of domain concepts, their attributes, restrictions, relations and operations can be derived. System analysts can take advantage of this information when developing a domain model:

– Information about changes in the content of documents in the process of information system exploitation can be applied while updating domain model,
– Information about changes in the document templates can be applied while updating document representation,
– Information about changes in the information system objects can be applied when generating and updating document content,
– Information about changes in business logic can be applied to change document content (for instance, manuals and instructions).

Solving tasks described above are based on the proposed approach of document activation by the means of intelligent processing tools.

Multi-agent semantic indexing algorithm

Simplifying the problem we assume that the first step of text analysis process was made (for instance using Yandex Mystem [Mystem, 2012]), i.e. a set of *morphological descriptors* for each word have been obtained. All other steps are performed by an agent-based semantic indexing. As it could be seen in fig. 1 syntax analysis is not used because it has high time complexity. Instead words order in a sentence is considered.

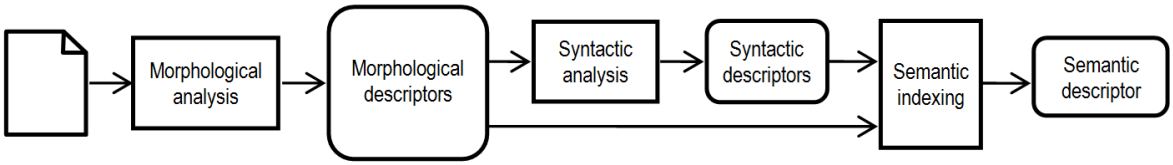

Figure 1. Steps of document analyses

The next step is a *semantic analysis*. The result of the semantic analysis is a semantic descriptor of plain text that binds the morphological descriptors to the elements of the domain ontology. Stop words are skipped.

The next step is a *structural analysis*. The structural analysis uses document's structure, ontology that describes structure and semantic descriptors of plain text. At this step every concept of structural

ontology tries to bind to the corresponding structural document element. The result of structural analysis is a semantic descriptor of whole text.

Descriptors (morphological, semantic) are a set of tags, which mark each word in the text.

Agent-based solution

Further, let us consider the process of building a semantic index based on multi-agent approach (see Fig. 2).

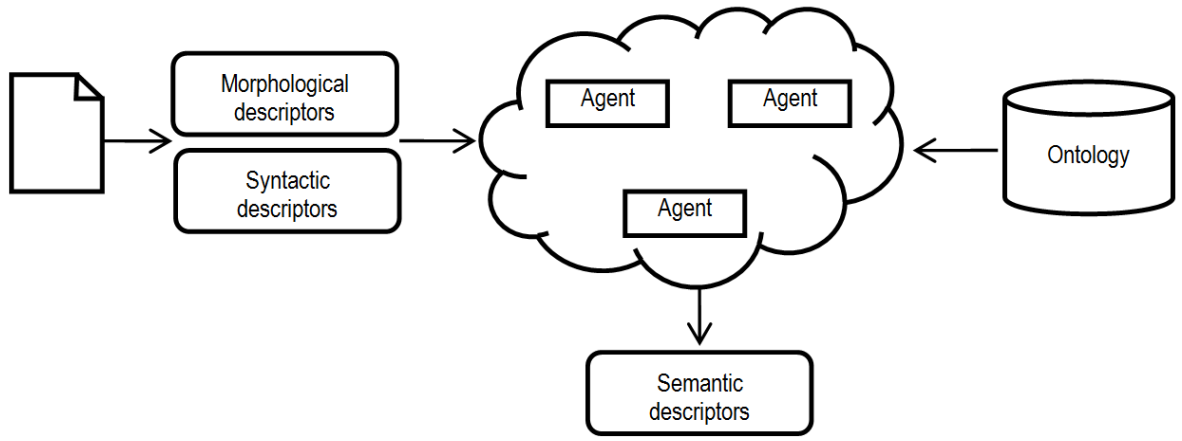

Figure 2. Architecture of agent platform

Agents have access to a domain ontology, structural ontology, morphological descriptors and electronic indexed documents. Indexing process is produced on the sentences in the text. Agents process sentences sequentially. The agents form a "team" to index the particular sentence. Thus, agents in the system are divided into teams after the start of the indexing.

Agent types

The following types of agents are identified in the system, according to the functional separation:

- Team Lead First Level Agent – TLFL agent,
- Team Lead Second Level Agent – TLSL agent,
- Word Indexer Agent – WI agent,

– Index Writer Agent – IW agent.

The task of WI agent is accessing the domain ontology and obtaining the set of possible semantic tags for the indexed word. An input word is passed to the WI agent for indexing with the parameters obtained at the stage of morphological analysis. The resulting set of possible semantic tags is passed to the TLSL agent.

TLSL agent binds to sentence morphological descriptors and distributes words to all available WI agents. TLSL agent finishes its work on the sentence when the consistent semantic descriptor is formed and written to the document. TLSL agent plans actions for the WI agents and participates in the auction for the resolution of contradictions. After building a consistent semantic descriptor TLSL agent transmits the generated semantic descriptor of the sentence to IW agent who writes semantic tags to the document.

TLFL agent binds to morphological descriptors of the document and distributes descriptors of the sentences to all available TLSL agents. TLFL agent monitors the TLSL agents work. If the work on the sentence is completed TLSL agent gives TLFL agent a new sentence. In addition, TLFL agent conducts an auction among TLSL agents to resolve ambiguity in the descriptors (see details in section «Agent negotiation»). Besides TLSL agents perform structural analysis. They distribute parts of structural ontology to TLSL agents.

Agent communication

Agents communicate through language FIPA ACL (Agent Communication Language developed by FIPA). Two types of actions are used. They inform (inform about anything) and perform (execution of an action).

Inform action type is implemented in the following cases:

– WI agent informs the TLSL agent about the completion of indexing word and give it the set of possible semantic tags; the content of the communication is as follows: (*id*, *tags*), where the *id* is the identifier word that come to be indexed, *tags* are returned set of possible semantic tags;

– TLSL agent informs the TLFL agent about a completion of an indexical sentence with a specific identifier; the content of this message contains an identifier of indexed sentence.

Perform action type is implemented in the following cases:

– TLFL agent gives to the TLSL agent a task to index a sentence with a specific descriptor; content will be like this: (*id*, *descriptor*), where the *id* is the identifier of the sentence,

descriptor is descriptor of the sentence received as a result of syntactic and semantic analysis;

– TLSL agent gives a task to the WI agent to index a word with specific *id*; content will look like this: (*id*, *word*, *parameters*), where *id* is *ID* of the word, *word* is the word for indexing, *parameters* are parameters obtained at the stage of morphological and syntactic analysis;

– TLSL agent gives a task to the IW agent to write semantic tag of specific word; content is as follows: (*word*, *tag*), where *word* is an indexed word, *tag* is just a semantic tag of indexed word.

Planning

The planning is dynamic. TLSL agents themselves form a team of agents from the available WI agents. A count of needed WI agents depends on a sentence structure. If there is a lack of WI agents at the team formation time the TLSL agent may designate to perform indexing of few words at once to the same WI agent. TLFL agent monitors the performance of TLSL agents work and if they are released it assigns them new sentences for indexing. Completing of agents (WI and TLSL) work is monitored not only by sending their corresponding messages of inform type, but also by changing their states (agent states) in the meaning of "vacant."

Agent knowledge bases

WI agents and IW agents are primitive reflex agents working in the mode of stimulus-response. Their main function is a simple, no inference, execution of work. There are only procedural steps in the knowledge bases of these agents.

Knowledge bases of TLFL and TLSL agents represent productions with embedded procedural actions. In fact, the script actions are necessary for the distribution of work between agents. Accordingly TLSL agent knowledge base contains a script for word distribution among WI agents, and TLFL agent knowledge base includes a script for sentences distribution between agents TLSL.

Agent negotiation

TLFL agent conducts an auction among agents TLSL, each of which has a contextual memory (training component). Every TLSL agent using the contextual memory votes for a one option of sematic descriptor of the sentence. Option of semantic descriptor of the sentence with the highest number of

votes will be considered as a true semantic descriptor of the sentence. The set of all consistent semantic descriptors of the sentences form the document semantic descriptor.

RELATED WORKS

Existing Document Ontology

Dublin core [Dublin core] is a set of metadata used to describe documents of various types (publications, audio records, video records). This set specification has status of official international standard (ISO: 15836 2003). The standard has two levels: Simple, comprising 15 elements and Qualified having three additional elements and element refinements (or qualifiers), which refine semantics of the elements. The main feature of Dublin Core is that every element is optional and might be repeated. Dublin Core is a powerful instrument used to describe resources of various types. The fact that it is widespread and flexible is its overwhelming advantage. However, it describes documents tags, i.e. information having indirect correlation with the document content. In this case it is impossible to describe other aspects of the electronic document.

Project ontologies «docOnto» [CNXML] developed by German research group KWARC (Knowledge Adaptation and Reasoning for Content) differ from other projects oriented on formal structure description development (CNXML document ontology) and document semantics (OMDoc document ontology). Members of this group also develop mechanisms of semantic document indexing and tools for document processing. CNXML document ontology (Connexions Markup Language) describes such terms as paragraph, section, reference etc. Ontology is formalized on UML. It gives detailed description of the document. Unfortunately, work in this direction is frozen, last changes date back 2007. One more direction in document ontologies creation is semantics description of documents for narrow subjects, where documents are well formalized, for example mathematical OMDoc documents. Mathematical Terms, theorems and several other terms are included in ontology.

Document ontology SHOE [SHOE] describes most types of documents. Academic papers are given particular emphasis. Dublin Core reference books and Document Classifier PubMed were the resource.

Document Ontology of Research Centre Linked Data DERI is developed by scholars of Irish Institute DERI (Digital Enterprise Research Institute) and is described in RDFS and OWL-DL. Terms referring project activity documentation are given in the ontology. Developers purposefully refused modelling structure and document content to accommodate flexibility and interoperability.

Muninn project document ontology became the result of processing archive documents of the First World War within the project Muninn WW1 [Muninn]. The Ontology describes bibliography, origins and storage description of the digital item. Most ontology classes are child classes of FOAF. That decision was compatibility possible, on the other hand, make adding additional features of document processing possible, i.e. features for representation document pages, copyright description, etc. One of the main ontology classes is Document, which is integrate class of FOAF Document and Creative Commons Works. Page class describes document pages, in its turn, Image class describes digital page image. Description of different document aspects, document structure in particular, is a significant benefit of this ontology. However, structure description is initially oriented on digital images of archive documents.

Each listed above document ontology has its advantages and disadvantages. We create our own ontology specialized on academic paper description.

System for creating text-based ontology

Nowadays there are some information systems that let you create text-based ontology models of documents or let you define correspondence of ontology models thereby transform one model into another one. We found two web-resources that let you create ontologies: OwlExporter and OntoGrid.

The core idea of OwlExporter is to take the annotations generated by an NLP pipeline and provide for a simple means of establishing a mapping between NLP (Natural Language Processing) and domain annotations on one hand and the concepts and relations of an existing NLP and domain-specific ontology on the other hand. Than the former can be automatically exported to the ontology in form of individuals and latter as data type or object properties [Witte] .

The resulting, populated ontology can be used then within any ontology-enabled tool for further querying, reasoning, visualization, or other processing.

OntoGrid is an instrumental system for automation of creating domain ontology using Grid-technologies and text analysis in natural language.

This system has bilingual linguistic processor for retrieving data from text in natural language. Worth derivational dictionary is used as a base for morphological analysis. It contains more than 3.2 million word forms. The index-linking process consists of 200 rules. “Key dictionary” is determined by words allocation analysis in text. The developers came up with new approach of revealing super phrase unities that consist of specific lexical units. The building of semantic net is carried out this way: the text is analyzed using text analysis system, semantic Q-nets are used as formal description of text meaning. The linguistic knowledge base of text analysis system is set of simple and complex word-groups of the

domain. This base can be divided into simple-relation-realization base and critical-fragment-set, that let you determine which ontology elements are considered in this text. The next step is to create and develop the ontology in the context of GRID-net. A well-known OWL-standard is used to draw the ontology structure.

Conclusion

The proposed model allows giving a formal definition of intelligent processing of electronic documents, providing a wide range of opportunities to integrate documents with ontological resources. The agent-based approach enables to resolve a problem of adding business logic into documents. In contrast to other approaches, in this case no additional programming code or attributes should be added to documents. So, it is possible to abstract from technological processing particularities and format specifications. Such a system will be more flexible, intelligent and adjustable to changing environment. A proposed approach lets solve a wide range of tasks connected with usage of electronic documents at all lifecycle steps, that in turn will allow to implement document-oriented paradigm of information system life cycle maintenance.

Acknowledgement

The reported study was supported by RFBR, research project No. 14-07-31273.

Bibliography

[Segaran, 2009] Segaran T., Evans C., Taylor J. Programming the Semantic Web, O'Reilly Media, 2009.

[Lukashevich, 2003] Lukashevich N.V., Dobrov B.V. Bilingual information retrieval based on the automatic conceptual indexing // Computational linguistics and intelligent technologies. Proceedings of the International Conference “Dialogue-2003”. Protvino. June 11-16 2003y. / Ed. by I.M.Kobozevoy, N.I.Laufer, V.P.Selegeya - M.: Science, 2003. - pp.425-432.

[CNXML] CNXML/DocumentOntology http://mathweb.org/wiki/CNXML/DocumentOntology

[Dublin] Dublin Core Metadata Element Set, Version 1.1 http://dublincore.org/documents/dces/

[SHOE] Document Ontology (draft) http://www.cs.umd.edu/projects/plus/SHOE/ onts/docmnt1.0.html

[Muninn] Muninn Documents Ontology http://rdf.muninn-project.org/ontologies/documents.html

[Bessonov, 2012] Bessonov V., Lanin V.A, Sokolov G. A semantic indexing of electronic documents in open formats/INFORMATION THEORIES & APPLICATIONS, 2012, P. 139-148

[Mystem, 2012] Program for morphological analysis of text in Russian “Mystem”. [Electronic resource] [Mode of access:http://company.yandex.ru/technologies/mystem/] [Checked at: 24.06.12]

[Witte] Witte R., Khamis N., Rilling J., Flexible Ontology Population from Text: The OwlExporter Dept. of Comp. Science and Software Eng. Concordia University, Montreal, Canada. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2010/pdf/932_Paper.pdf

Authors' Information

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

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