International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 12 (2015) pp. 31441-31448

http://www.ripublication.com

Use Of Natural Language Texts To Create Ontologies During The Development Of Applied Interface For User Requests In Analytical Systems With High Performance

A.V. Prokofyev[1], M.A. Khlamov[2], R.G. Chesov[1]

[1]LLC "Flexby Solutions", Dolgoprudny, Moscow region, Zhukovskogo street, 3.
[2] Moscow Physical-Technical Institute (State University), Dolgoprudny, Institutsky lane, 9.

Abstract

The present work describes the ways of ontology creation for the development of applied interfaces for analytical systems (BI). Some shortcomings of this approach and the ways of their solutions are described as the system of ontology automatic creation. This approach uses the opportunity of ontology creation using the phrases of contracted natural language. The use of Russian language is offered as an example of API creation. This approach may be extended to other natural languages.

Keywords: ontology, formalization, text, restricted Russian language, BI, Big Data, API.

1. INTRODUCTION

One of the key issues of the accumulated data processing speed and quality improvement is the creation of convenient interfaces for the development of applied logic and the writing of user queries in industrial and enterprise analytical systems. The program interface should allow the creation of applied logic for data processing, the writing of queries without the involvement of system developers, including the use of engineering knowledge principles in the analyzed subject domain [1]. Using only the domain experts will increase the speed of information processing and analysis by reducing the time of the data logical description preparation. The main purpose of the applied logic development is the transition of business process management from the level of information technologies to the level of logic and information [2]. One possible approach during the creation of user queries interface

and the applied logic development of a high-performance analytical system is the use of ontology [3,4].
Ontology is the conceptual modeling of a subject domain. Today the task of ontology creation for different subject domains (SD) is more than ever relevant, for example, at the optimization of large and extra large volumes of data processing [5].

2. BASIC APPROACHES FOR ONTOLOGY DEVELOPMENT

During the development of ontologies [6,7], which often may be regarded also as a knowledge base one has to face the following challenges:

1. At least two men are necessary to create an ontology - an expert in this field and a knowledge engineer. Sometimes only one expert is enough if this expert has cognitive knowledge and the knowledge of technologies. But the opposite happens more often - an expert does not take part in this process for whatever reason and a knowledge engineer often has to deal with an unfamiliar area. Therefore, the ontology creation technology is an actual one. This ontology should be based on available information provided in texts, different types of dictionaries and other materials.
2. There is no only one right way of domain modeling - there are always viable alternatives. The best solution is almost always depends on an intended application and expected extensions.
3. The ontology development is a necessary iterative process.
4. The ontology concepts should be close to objects (physical or logical ones) and the relations in a subject domain of interest.

Thus, there is a need to provide a convenient system of ontology automatic creation and their use as an application interface for the creation of business logic in respect of high analytical systems. There are several approaches to solve this problem:

1. The visual approach which allows the experts to "draw" ontologies directly that helps clearly formulate and explain the nature and structure of phenomena.
2. The creation of ontologies from texts in a simplified natural language.

The second approach seems the most appropriate, since many organizations accumulated a lot of knowledge in the form of texts: documents, e-mails, internal expert systems and wiki-pages.
The possibility of ontology creation from texts in a "pure" natural language would solve the problem of ontology creation and their subsequent use completely. However, at this moment the use of so-called "contracted" language seems to be possible [8]. The "contracted" Russian language (CRL) is the formal language with the syntax similar to the ordinary Russian language which supports the ability of transfer to the first-order logic. Anyone who is able to read in Russian, may read CRL without additional training. However, some practice is necessary in order to write in CRL and take into account the syntactic and semantic constraints. The flexibility of ordinary Russian language and any of its imagery is not supported. Despite these

limitations, CRL is similar to Russian language, which is used in the software specifications, mathematical books and definitions and in the axioms of formal ontology.

3. BASIC PRINCIPLES OF CRL USE

In contrast to Russian language, the basics of which require the authorization of many uncertainties, CRL has the syntax that allows you to solve any uncertainty during the parsing of a sentence.

CRL sentences are similar to the subset of usual Russian language sentences. Like the sentences in Russian each CRL sentence consists of one or more statements, and each statement has one main verb. The sentences may be divided into two principles: depending on the structure of a sentence the sentences may be simple, composite or complex; depending on the use of a sentence the sentences may be declarative, interrogative, or imperative. Let's consider a few examples:

Simple sentence. A simple sentence consists of only one statement.
Example: "Tom is a cat".

Composite sentence. A composite sentence has one major statement and one or more subordinate statements, depending on a major statement. CRL has two types of subordinate sentences: an attributive clause beginning with the relative pronoun 'that'; and a conditional expression, joining the main expression by the phrases only "if", "then" and only "then" or a separate combination of "if" and "then".
Example: "If the number is even, then the number is not odd."

Complex sentence. A complex sentence is composed of two or more independent sentences, which may be simple or composite. The sentences are joined by the word "and" or the word "one of two" as well as by one or more cases of the word "or".
Example: "There is some cat on the mat, the cat is red and the mat is blue".

"There is a bonus in the sales department and the sales department is within the organization".

Declarative sentence. A declarative sentence announces a judgment, which may be a simple or a compound statement of fact with a plurality of conditions and opportunities.
Example: "Petrov gets a bonus in the accounting department".

Interrogative sentence. An interrogative sentences poses a question. It usually follows the sequence of declarative sentences, stating some facts or axioms which may answer a question. For the further question about numbers, an answer may be obtained from the question itself and the preceding definitions of the words 'even', 'number' and 'less' semantics.
Example: "What is the even number less than 3?"

Imperative sentence. An imperative sentence specifies a command that a computer system expects to perform. Each imperative sentence starts with a verb whose implicit subject is a computer system. The basic CRL version may use only three verbs in an imperative sentence: import, declare and note.
Instead the transfer command into the logic of implementation CRL performs them directly: for the verb 'to import' they import the named file into the current workspace; for the verb 'to declare' they transfer the declaration into a format used by CRL analyzer; for the verb 'to note' they ignore the comment.
CRL words are divided into a small number of reserved words for an unlimited number of declared words. All reserved words are written as Russian words, but their values are limited to only one meaning or a small number of meanings, differentiated by syntax. There are ten types of reserved words:

Boolean operators: not, and, one of the two, neither of the two, also not, if, then.
Quantifiers: some, something, someone, anyone, everything, everyone, no, nothing, no one.
Special verbs: there is, it is.
Questions: who, what, when, where, what.
Relative pronoun: who.
List connector: and.
Special lists: blank, the rest, nothing else, no one else.
Argument markers: from, than, how.
Special phrases: exist; like this; if only; then and only then; false that; true that.

Some reserved words have more than one meaning, but always the correct meaning may always be obtained from syntax.
The announced words may denote names, nouns, verbs, adjectives, adverbs or prepositions. These may be the words with the same spelling as the Russian language words, the words, written by underlining - "hard_disk" or arbitrary sequences of letters, numbers and underlinings.
Thus, CRL may be used for ontology description language and the development of user inquiries and the development of data processing applied logic.

4. ONTOLOGY DEVELOPMENT EXAMPLES IN ANALYTICAL SYSTEMS

The primary data which are analyzed by analytical systems include the financial data, various transactions, employee data, the results of user behavior analysis and other structured and unstructured data, the analysis of which helps to make decisions.
The example of ontology editor FluentEditor shows the applicability of the approach in practice for English language [9]. This paper describes an attempt to develop an automatic ontology editor for Russian language on CRL basis.
The basis of any ontology development is the determination of ontology classes and the relationships between the sets of objects and the parts of objects. The most important semantic relations is the concept of "genus - species" and "part - whole".

The relationship between the subset and superset is the ratio "genus - species". This ratio determines that each element of the first set includes the second one (isA is performed for each element), as well as the logical connection between the subsets: that the first one is not larger than the second one and that the properties of the first set are inherited by the second one. The ratio of "genus - species" is often used for the navigation in an information space. These relations will be described using the keyword "Any" (and its forms depending on the gender of a noun) and the word "is".
For example: "Any award is the reward for results".
Such a proposal is transformed into OWL-format as follows:

<owl:Class rdf:about="reward_for_results"/>,

<owl:Class rdf:about="reward">

<rdfs:subClassOf rdf:resource="reward_for_results"/>

</owl:Class>

Ontologies, depending on the purpose of creation, may be purely abstract or applied. Most often abstract ontologies are a top-level ontologies such as Cyc, DOLCE, SUMO, the ontology of John Sowa (J.Sowa) and others [11, 12, 13]. They contain the most general terms: essence, phenomenon, process, object, role, space, time, matter, event, action, etc. In practice, ontology together with a set of class instances develops the knowledge base. In fact, it is difficult to determine where ontology ends and where a knowledge base begins. The relationship between an object and a plurality indicating that an object belongs to this plurality, is called the classification ratio. They say that a class classifies its instances, and the ratio IsA between an instance and its class is called the classification ratio.
Thus, the creation of ontology class instances is one of the most important functions for an automatic ontology editor. The creation of instances is performed as follows:
"The best employee of the month" is a reward.
Such a sentence is transformed into OWL-format as follows:
<Reward rdf:ID ="Best Employee of the Month">
</Reward>
Another important semantic relation is the relation "part - whole". It expresses the ratio between the set of objects or concepts and the relation that binds these objects and leads to the appearance of integrative properties (new, not inherent to the objects in their disunity) and laws. The full description of such a relation for all objects in the system allows you to continue the obtaining of more consistent and accurate results.
The creation of relationships "part-whole" is as follows:
A reward consists of sales, plan performance success and reporting.
Such a proposal is transformed into OWL-format as follows:
1. The creation of properties "consists of" and "makes part of".
<owl:ObjectProperty rdf:about="consistsOf"/>
<owl:ObjectProperty rdf:about=" makesPartOf">

<owl:inverseOf rdf:resource="/consistsOf"/>
</owl:ObjectProperty>

2. Sentence processing.
<owl:Class rdf:about="reward">
<rdfs:subClassOf>
<owl:Restriction>
<owl:onProperty rdf:resource=" consistsOf"/>
<owl:onClass rdf:resource="sale"/>
</owl:Restriction>
<owl:Restriction>
<owl:onProperty rdf:resource=" consistsOf"/>
<owl:onClass rdf:resource="plan_performance_success"/>
</owl:Restriction>
<owl:Restriction>
<owl:onProperty rdf:resource=" consistsOf"/>
<owl:onClass rdf:resource="reporting"/>
</owl:Restriction>
</rdfs:subClassOf>
</owl:Class>

The example of a query making.
The sequence of sentences separated by a semicolon, is called an extended sentence. The main purpose of an extended sentence is to allow extend the scope of variables and quantifiers beyond a separate sentence. In the following example, the first noun group is transferred into an existence quantifier, such as (x: Rewarded_employee). In the next sentence the referential noun group "this employee" is presented as a link to the same variable x.
CRL helps you to make a request to the ontology. For example, we need to know the SNP of the employees whose rewards need to be paid.
Then, the query will be as follows: A rewarded_employee is the employee who has the performed_plan_of sales at the amount of 20,000 - 50,000; Who is this fellow?
After its performance we get a list of employees that correspond to the condition of "performed sales plan at the amount of 20,000 - 50,000".

5. CONCLUSIONS

Thus, we may conclude the following:

1. The development of a rather complex object - such as the ontology encoding the information of a general software in various configurations of service systems - is possible to implement through a series of simple sentences by a limited native language NL, built during the semantization from the original pool of texts.
2. The quality of the performed semantization, expressed in a set of simple sentences in a limited NL, despite its apparent simplicity, essentially depends

on the completeness and other qualities of the used pool of texts. And, as in the case of direct encoding through the editor of ontologies, it depends on knowledge engineer experience.

3. The developed technology, built on a set of simple sentences using a limited and specialized ~~native~~ NL, was an easy and qualitative one accurately for the fast development of a basic ontology. The integration of knowledge based on such a technology is performed faster and better.

At the moment, we investigate the possibility of a software implementation for an applied development interface, which will allow to develop ontologies automatically from the texts by contracted Russian language. At the moment, there is already a possibility to develop a hierarchy of simple sentence concepts, to reflect the relationship "part-whole". The resulting ontology is stored in xml format and may be used further in an analytic system.

6. SUMMARY

The performed study of ontologies developed using a natural language, showed the applicability of this approach at the creation of an application interface for the writing of queries and data processing in analytical systems. Russian language was used as an example, but this approach may be extended to other natural languages, which will create an application interface of analytical system developer, which will allow to process the requests, made in a usual language of an average system user.

In the near future an automated processing of other properties important for ontology creation is planned - the setting of user properties and relations, and their application to the classes and ontology instances, the setting of constraints on properties and relationships.

CONFLICT

The author confirms that the presented data do not contain any conflict of interest.

ACKNOWLEDGEMENTS

This work was supported by ~~RF~~ Ministry of Education and Science of Russian Federation (unique work number ~~the~~ ~~article~~ ~~number~~ RFMEFI57914X0069).

REFERENCES

[1] Wang, Hai; Wang, Shouhong Ontological map of service-oriented architecture for shared services management // Expert Systems with Applications, Volume 41 (5) – Apr 1, 2014

[2] Rykov V.V., Hlamov M.A. Creation of ontologies from text // International Conference "Physical and technological computer science (CPT2013). Cyprus, Larnaca, 12-19 of May 2013 - M.: IFTI 2013.
[3] Pathak, Bimalendu; Sinha, Madhavi //Analytical Study of Agent Based Distributed Data Mining and its Ontology // Institute of Electrical and Electronics Engineers — Mar 5, 2014
[4] Kotenko, Igor; Polubelova, Olga; Saenko, Igor // The Ontological Approach for SIEM Data Repository Implementation, Institute of Electrical and Electronics Engineers — Nov 20, 2012
[5] Rykov V.V. Processing of non-numeric data. Knowledge management. - M .: MIPT, 2007.
[6] Klimenko S.V., Rykov V.V., Hlamov M.A. Technologies of complex technical objects construction ontology (according to AUV example - an autonomous underwater vehicle) // International Conference "Physical and technological computer science (CPT2013). Cyprus, Larnaca, 12-19 of May 2013 - M.: IFTI 2013.
[7] Gavrilova T.A. [et al.]. Visualization of ontologies as the tool of knowledge acquiring // Proceedings of the 4th International Workshop on applied semiotics, semiotic and intelligent control ASC/IC'99. Moscow, 1999.
[8] Dolganov A.A. Problems of accumulated knowledge formalization and reuse. Thesis. - M.: STANKIN, 2010.
[9] Ontology Editor: Fluent Editor // http://www.cognitum.eu/semantics/FluentEditor2/Default.aspx).
[10] Mike Dean, Guus Schreiber OWL Web Ontology Language Reference // W3C Recommendation – 2004 - http://www.w3.org/TR/owl-ref/
[11] http://www.jfsowa.com/ontology/index.htm
[12] Ramachandran, Deepak, P. Reagan, K. Goolsbey. First-Orderized ResearchCyc: Expressivity and Efficiency in a Common-Sense Ontology. In Papers from the AAAI Workshop on Contexts and Ontologies: Theory, Practice and Applications. Pittsburgh, Pennsylvania, July 2005.
[13] http://www.adampease.org/OP/