ISSN 2278-3091
Volume 9, No.3, May - June 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse172932020.pdf
https://doi.org/10.30534/ijatcse/2020/172932020

The Technology for Semantic Interoperability based on a Cognitive Approach

Dmitry Gennadievich Korneev, Mikhail Samuilovich Gasparian, Andrey Aleksandrovich Mikryukov, Elena Valerievna Yaroshenko, Galina Evgenyevna Golkina
Plekhanov Russian University of Economics, Russia

ABSTRACT

The article presents the results of research conducted to select the means of software implementation (construction) of an ontology used to ensure semantic interoperability of information systems (IS).

Previously, the authors of the article have identified several specific features of this type of ontology. Based on the analysis of structures and methods for constructing ontologies, it has been concluded that it is advisable to include concepts that allow describing both the static condition and dynamic changes in the state of objects in the data domain, into the ontology used to ensure semantic interoperability of the IS. The authors have identified the main types of concepts used and have determined a set of mandatory attributes (properties) that characterize each of the objects and possible types of relationships with other concepts.

The present article analyzes the possibilities of using existing ontology description languages, metadata representation models, system design languages, algorithms, and software implementations of ontology editors, as well as the possibilities of using relational and graph database management system (DBMS) to build ontologies used to ensure semantic interoperability. The article substantiates the choice of a tool for software implementation (construction) of an ontology used to ensure semantic interoperability of the IS. Using the selected software tools, a fragment of ontology is constructed, which can be applied to ensure semantic interoperability of the IS. In conclusion, the authors present a unified algorithm for constructing ontology to ensure semantic interoperability.

The practical significance of the obtained scientific results lies in the development of approaches to the creation of the IS. The results obtained will allow creating IS that would implement the cognitive functions of understanding the content of requests when processing them in the means of the interaction of the IS. The research results in the field of semantic interoperability will allow improving the intelligent IS performance quality that will enable IS to solve a wider range of intelligent tasks.

Key words: open systems, semantic interoperability, software-bearing systems, ontological engineering, ontology description languages, relational and graph database management systems (DBMS).

1. INTRODUCTION

To ensure the ability to interact with each other, open systems must have the interoperability property. In ISO/IEC 24765-Systems and Software Engineering-Vocabulary [1], interoperability refers to "the ability of two or more systems or elements to exchange information and to use the information resulting from the exchange". Standards and research on the interoperability issues consider various levels of the IS interactions.

Currently, the solution to problems of ensuring semantic interoperability of the IS has the greatest scientific and practical significance. This is also because in recent years, intelligence of IS, including those that operate using IoT technology, has been increasing dramatically. IS are being created that can replace a person in many cases, including those in the field of intelligent decision-making. Understanding not only the syntax but also the meaning of the request that comes to the IS from another system will allow giving more accurate response, which, in turn, should be maximally understood by the IS that generated the request. Providing semantic interoperability is associated with the need to apply ontologies of concepts used in processes and describing the processes of the IS functioning. The need to apply an ontological approach in the problems of intelligent interaction of IS is justified, for example, in the works [2, 3] dealing with the issues of semantic interoperability.

The principles and methods of semantic structuring of the data domain of the IS functioning based on the ontological approach are described, for example, in [4-8]. The present article considers an approach to ensuring semantic interoperability of the IS using concepts that allow describing both static and dynamic changes in the conditions of objects in the data domain [9]. Solving the problems of ensuring semantic interoperability gives a new impetus to the development of the IS towards improving the ways of applying the cognitive approach to the implementation of the IS.

2. CHOOSING A TOOL FOR SOFTWARE IMPLEMENTATION OF ONTOLOGIES USED TO ENSURE THE IS INTEROPERABILITY

The ontology description languages are discussed below to select the most optimal language for the creation and application of ontologies to ensure semantic interoperability of the IS.

There are several languages used to write semantic models. Among them, RDF, RDFS, and OWL are the main languages. The RDF (Resource Description Framework) is the metadata description language that forms the basis of the Semantic Web. It represents the family of languages that are used to describe relational data models, whose specificity is that resources and properties are identified using global identifiers, such as URI. The RDF describes the data domain in terms of resources, resource properties, and property values. The RDF data can be considered as a set of the following statements (triplets): *subject*, *predicate (property)*, and *statement object*, and can be represented as a directed graph formed by these statements. The RDF is a universal metadata description language and requires configuration for specific specialized tasks. The way of such configuration is to extend RDF using dictionaries, one of which is RDFS (Resource Description Framework Schema), which is used to describe ontologies.

The OWL (Web Ontology Language) [10] extends the RDFS dictionary (mainly by defining classes and subclasses), introducing class comparability relations (sameAs, differentFrom, equivalentClasses, etc.), property characteristics (inverseOf, TransitiveProperty, SymmetricProperty, etc.), property restrictions (for example, which classes they belong to, or the cardinality of properties), class intersection, etc. Three OWL language dialects are distinguished depending on their expressive properties.

- The OWL Full which has the maximum expressive power, but does not guarantee the computability of logical conclusions in the ontology created using this language. For example, in the OWL Full language, classes can simultaneously act as both a class and an instance, which can lead to contradictions in the ontology description and the inability to make logical inferences based on existing rules.
- The OWL DL (Description Logic) guarantees computational completeness (logical output is computable) and solvability (calculations are performed in a finite time). The OWL DL contains all OWL Full constructs but their use is limited.
- The OWL Lite has the least expressive properties, but can be used as an intermediate link in the transition from simple taxonomies to ontologies.

The simplest form of storing ontologies is an OWL file. When reading the OWL file, a model (a set of statements) is created in RAM, which is used in further work. The obvious drawback of this approach is the increase in the amount of RAM required, as well as a significant increase in the loading time of OWL files as the volume of metadata contained in the ontology increases. The need to use special language tools for extracting metadata stored in ontologies makes it necessary to build ontology repositories based on DBMS.

From the standpoint of structural features, graph databases are best suited for storing ontologies. In this case, graph vertices can be used for storing ontology concepts, and graph edges can be used for storing relationships between concepts. Vertexes and edges can contain any set of attributes. For oriented graphs, an edge always has a start node, end node, type, and direction. Graph DBMS support methods for Creating, Reading, Updating, and Deleting data (abbreviated as CRUD) based on the graph data model.

Considering the capabilities of relational DBMS as a mechanism for creating and storing ontologies, the following can be noted. Traditionally, the disadvantage of relational DBMS, when working with graph data models, was the lack of tools for implementing hierarchical queries. For example, searching for vertexes connected to a given vertex via any other vertexes required performing numerous JOIN operations, which significantly slowed down query execution. However, most contemporary DBMS have overcome this limitation.

The authors of [11] describe an approach for organizing the semantic data search that takes into account the advantages of both graph and relational DBMS. In this case, a graph stores a set of interrelated concepts that reflect the semantics of the data domain. Data are stored in relational tables that contain a significant number of records. Initially, the query selects graph vertices based on the specified conditions imposed on the relationships between concepts. The selected vertexes contain attribute values that are used to search for records in relational tables. The graph model search is organized by the Neo4j DBMS, while the search for records in relational tables is organized by the MySQL DBMS.

Analyzing the capabilities of using graph and relational DBMS for working with ontologies, it should be noted that they do not have specialized mechanisms that support axioms and rules for inferring statements based on the relationships of concepts. To implement these mechanisms, it is necessary to write software packages in procedural languages, which are inefficient to use for large amounts of stored data.

In this case, special consideration should be given to the ORACLE DBMS which today can be attributed to the class of object-relational DBMS. The ORACLE 11g operates based on implemented mechanisms that are united by the term of Semantic Technologies [12, 13]. Version 11g provides the ability to export and import OWL structures and supports the OWLPrime ontology description language, which is a subset of the above-mentioned OWL DL language. The latter provides opportunities for the following:

- creating an ontology structure (class, subclass, property, subproperty, domain, range, and type);
- specifying property characteristics (transitive, symmetrical, functional, inverse functional, and inverse);
- comparing classes (equivalence and disjointness);
- comparing the properties (equivalence);
- comparing essences (same, different);

• setting restrictions on properties (hasValue, someValuesFrom, allValuesFrom).

To support OWLPrime, more than 50 rules are implemented that are used in the course of logical output. The rule consists of a condition (if), filter (conditions), and output (then) [14]. The ORACLE 11g implements the ability to configure custom rules using the OWLIF language (IF-THEN constructs). One can impose certain restrictions on the rules which the user can create. For example, one can specify that a user can only create logical output in the system within the subClassOf hierarchy, and at that, one can limit the number of output steps. Requests to extract information from ontologies in ORACLE 11g are made using the SPARQL language. To connect output rules created by the user, the SEM_RULEBASES construct is used in the SPARQL queries.

A sequence of actions that are needed to create ontology and work with it in the ORACLE 11g environment, is shown below. As an example, a fragment of the Finance ontology is created to ensure the interoperability of the financial company's IS.

1) Preparatory activities.

a) Creating a Tablespace.

CREATE TABLESPACE rdf_tblspace;
DATAFILE '/usr/lib/oracle/oradata/orcl/rdf_tblspace.dat' SIZE 1024M
REUSE AUTOEXTEND ON NEXT 512M MAXSIZE 5G
SEGMENT SPACE MANAGEMENT AUTO;

b) Installing the SEM_APIS package in the database:

@$ORACLE_HOME/md/admin/catsem11i.sql

c) Creating a network.

EXECUTE
SEM_APIS.CREATE_SEM_NETWORK('rdf_tblspace');

2) Creating the structure of the Finance ontology.

a) Creating a table for storing the ontology:

CREATE TABLE Finance (id number, triple sdo_rdf_triple_s);

b) Creating a table-based model:

EXECUTE SEM_APIS.CREATE_SEM_MODEL(
'Finance', is the model name
Finance', is the table name
'triple'); is the name of the triplet column.

c) Filling in the ontology facts: the bank can open accounts, accounts may belong to individuals.

INSERT INTO Finance VALUES
(1, sdo_rdf_triple_s('Finance',
'http://foo.com/Bank', 'http://foo.com/rel/Open',
'http://foo.com/Account'));
INSERT INTO Finance VALUES (2, sdo_rdf_triple_s('Finance',
'http://foo.com/Human','http://foo.com/rel/IsOwner',
'http://foo.com/Account'));

d) Filling in user-defined rules. If the concept Bank (Bank, concept type of "Object") is linked by the link "Open" (Opens, link type of "Action") with the Account concept (Account, concept type of "Object"), and if the concept Human (Human, concept type of "Object") is linked by the IsOwner link (is an Owner, link type of "Association") to the Account concept, then the Human is linked to the Bank by the link IsClient (is a Client, link type of "Association").

EXECUTE SEM_APIS.CREATE_RULEBASE ('Finance_rulebase');
INSERT INTO mdsys.semr_user_rulebase VALUES ('sibling_rule1',
--input: '(?Bank<http://foo.com/rel/Open>?Account)
(?Human<http://foo.com/rel/IsOwner> ?Account)', NULL,
--output: '(?Human <http://foo.com/rel/IsClient> ?Bank)', NULL);

3) Extracting knowledge based on the constructed ontology.

Example of a query for sampling implicit facts using predefined (basic) OWLPRIME rules of OWL ORACLE 11g ontologies (for example, symmetry and transitivity of SameAS type relationships (equivalence)) and custom Finance_rulebase rules. All individuals, who are clients of the financial company, will be identified.

SELECT s,p,o
FROM TABLE(SEM_MATCH('(?s ?p ?o)',
SEM_Models('Finance'),
-- set of rules used: SEM_RULEBASES('OWLPRIME',
'Finance_rulebase'),null, null)).

Based on the results of the conducted research, the following algorithm for constructing ontology to ensure semantic interoperability of the IS can be proposed:

1. Identifying ontology concepts and defining the semantics of relationships following the rules set out in [3].
2. Describing the ontology using the OWL DL language using the Protégé 5.0 ontology editor (creating an OWL file) [15].
3. Creating structures for storing ontologies in the ORACLE 11g DBMS environment.
4. Filling in the structures following the description of clause 2 (uploading the OWL file to the ORACLE 11g DBMS).
5. Creating additional user rules in the ORACLE 11g DBMS environment to obtain implicit knowledge.

5. CONCLUSION

The practical significance of the obtained scientific results lies in the development of approaches to the construction of intelligent IS. The results obtained will allow creating IS with more developed means of interaction at the semantic level. The research results in the field of semantic interoperability will allow increasing the intelligence of the IS that will enable IS to solve a wider range of tasks.

In the future, it is planned to test the developed algorithm for implementing ontologies of various data domains, whose business processes will be automated using IS.

ACKNOWLEDGEMENT

The article was prepared with the support of the Russian Foundation for Basic Research (Grants No. 18-07-01053 and No. 20-07-00926).

REFERENCES

1. S. Chen, B. Mulgrew, and P. M. Grant. **A clustering technique for digital communications channel equalization using radial basis function network**s, *IEEE Trans. on Neural Networks*, Vol. 4, pp. 570-578, July 1993. https://doi.org/10.1109/72.238312
2. ISO/IEC 24765-Systems and Software Engineering-Vocabulary. Retrieved from https://www.iso.org/standard/50518.html
3. S.N. Prayma. Obespechenie semanticheskoj interoperabel'nosti otkrytyh sistem obrazovaniya vzroslyh posredstvom agento-ontologicheskogo podhoda [Ensuring semantic interoperability of open systems of adult education through an agent-ontological approach]. Perspectives of Science and Education, 1, 2014.
4. V.V. Golenkov, N.A. Gulyakina, I.T. Davydenko, D.V. Shunkevich, A.P. Eremeev. Ontologicheskoe proektirovanie gibridnyh semanticheski sovmestimyh intellektual'nyh sistem na osnove smyslovogo predstavleniya znanij [Ontological design of hybrid semantically compatible intelligent systems based on the semantic representation of knowledge]. Ontology of Design, vol. 9-1, no. 31, 2019.
5. Yu.F. Telnov. Principy i metody semanticheskogo strukturirovaniya informacionno-obrazovatel'nogo prostranstva na osnove realizacii ontologicheskogo podhoda [Principles and methods of semantic structuring of information and educational space based on the implementation of the ontological approach]. Economics, Statistics, and Informatics. Bulletin of the Educational and Methodological Association 1, 2014.
6. Yu.F. Telnov, M.S. Gasparian. Realizaciya processov uchebno-metodicheskogo obespecheniya v integrirovannom informacionno-obrazovatel'nom prostranstve na osnove servisnoj arhitektury [Implementation of educational and methodological support processes in an integrated information and educational space based on service architecture]. Economics, Statistics, and Informatics. Bulletin of the Educational and Methodological Association 1, 2015.
7. M.S. Gasparian, S.A. Lebedev, Yu.F. Telnov. Integraciya elementov informacionno-obrazovatel'nogo prostranstva na osnove ontologicheskogo podhoda [Integration of information and educational space elements based on an ontological approach]. Proceedings of the Anniversary 10th International scientific and practical conference "Electronic Kazan 2018" (Information technologies in the contemporary world). Kazan: Scientific notes of the Institute of Social and Humanitarian Knowledge, vol. 16, no. 1, 2018.
8. D.G. Korneev, M.S. Gasparian, I.A. Kiseleva, A.A. Mikryukov. Ontological engineering of educational programs. Revista Inclusiones, vol. 7, pp. 2-3, 2020.
9. D.G. Korneev, M.S. Gasparian, A.A. Mikryukov. Ontologicheskij podhod k modelirovaniyu innovacionnyh processov na primere raspredelennoj obrazovatel'noj seti vuza [Ontological approach to modeling innovation processes in terms of a distributed educational network of a university]. Open Education, vol. 23, no. 5, 2019. https://doi.org/10.21686/1818-4243-2019-5-4-13
10. A.V. Boychenko, D.G. Korneev, V.A. Kazakov. Razrabotka struktury ontologii dlya obespecheniya semanticheskoj interoperabel'nosti informacionnyh sistem [Development of the ontology structure to ensure semantic interoperability of information systems]. Proceedings of the 21st Russian scientific and practical conference "Enterprise engineering and knowledge management". Moscow: Moscow State University of Economics, 2018.
11. OWL Reference. W3C Recommendation. Retrieved from: http://www.w3.org/TR/owl-ref/
12. A.V. Boychenko, D.G. Korneev, V.A. Kazakov. Poisk dannyh, osnovannyj na semanticheskoj modeli [Data search based on semantic model]. Proceedings of the 18th Russian scientific and practical conference "Engineering of enterprises and knowledge management". Moscow: Moscow State University of Economics, 2015.
13. Oracle Database Semantic Technologies Developer's Guide. Retrieved from: http://download.oracle.com/docs/cd/B28359_01/appdev.111/b28397/toc.htm
14. D.J. Abadi, A. Marcus, S.R. Madden, K. Hollenbach. Scalable semantic web data management using vertical partitioning. Vienna: VLDB, 2007.
15. A.A. Wu. Scalable RDBMS-based inference engine for RDFS/OWL. Retrieved from: http://ontolog.cim3.net/file/work/DatabaseAndOntology/2007-10-18_AlanWu/RDBMS-RDFS-OWL-InferenceEngine--AlanWu_20071018.pdf
16. Protégé 5 Documentation. Retrieved from: http://protegeproject.github.io/protege