Warehouse Development of Ontology for Providing Semantic Interoperability[1]

Dmitry Korneev[0000-0001-7260-4768], Alexander Boichenko[0000-0003-3113-9446] and Vasily Kazakov[0000-0001-8939-2087]

[1] Plekhanov Russian University of Economics, Moscow, Russia
Korneev.DG@rea.ru

Abstract. The modern stage of social development is characterized by the ever increasing penetration of information technologies (IT) into various spheres of human activity. Experts in various fields of knowledge associate the development of human civilization at the present stage with the development of the digital economy and the fourth industrial revolution, which involves the mass introduction of cyber-physical systems in production ("Industry 4.0"), serving human needs, including life, work and leisure [1]. Currently, the intellectual component of information systems (IS), including those designed to control devices and mechanisms, is rapidly increasing. A significant part of this intellectual component is the ability of information systems to interact at the semantic level. In this regard, there is a growing need for a more intellectual interaction of IS among themselves. It is assumed that intellectual systems is should understand not only the syntax but the semantics (meaning) of the query addressed to it. So, for example, at present, in connection with the development of the Internet of Things technologies (IoT) there is a need to solve the problems of "communication" devices ("things"), united in information networks. To ensure semantic interoperability in the works [2, 3], an approach was proposed, which consists in the implementation of an integrated knowledge base in each of the interacting IoT, reflecting the integrated ontologies of the subject areas of interacting systems. This article discussions the possibility of using language and software tools to describe and create a repository of ontologies used to ensure the semantic interoperability of IS.

Keywords: semantic interoperability, software intensive systems, ontological engineering

1 Introduction

Currently, the use of the concept of open systems in the creation of software is one of the main trends in the field of information technology. In the documents of the Associ-

[1] The article was prepared with the support of the Russian Foundation for Basic Research (grant № 18-07-01053).

CEUR-WS.org/Vol-2413/paper09.pdf

ation IEEE (Institute of Electrical and Electronics Engineers), under open systems refers to systems that implement "a comprehensive and consistent set of basic international standards of information technology and profiles of functional standards that specify interfaces, services and supporting data formats to ensure the interaction and mobility of applications, data and personnel."

To be able to communicate with each other, open systems (see, for example, the series of POSIX standards developed by IEEE) must have the interoperability property. Interoperability in ISO/IEC 24765-Systems and Software Engineering-Vocabulary [4] is defined as "the ability of two or more systems or elements to exchange information and to use the information obtained from the exchange".

In standards and research on the problems of interoperability, discusses the various levels of interaction systems. It is further proposed to focus on the European interoperability stack (EIF - European Interoperability Framework v2.0) [5] stratifying interoperability functions to the next series of logical levels:

1. Normative – involves the interaction of systems in a single regulatory and legislative environment;
2. Organizational – refers to the organizational aspects of the functioning of information systems and involves common business processes and regulations of their functioning;
3. Semantic – the ability of systems to equally understand the meaning of the information they exchange;
4. Syntactic – the ability to exchange data, the ability of systems to integrate;
5. Technical – the organization of the relationship between the systems.

At the first two levels of the EIF stack, the initial requirements for the design of information systems are set, organizational measures are taken to unify the relevant regulatory documents and business processes. To ensure the fourth and fifth levels of interoperability of the EIF stack in the design and development of information systems, they must include certain software tools. These levels of interoperability are well understood and their practical implementation is not currently causing serious difficulties.

A new stage in the development of solutions in the field of interoperability is associated with the emergence and development of technologies of Software Intensive Systems (SIS) - systems, the functionality of which is mostly or completely determined by the software [6]. The development of SIS and IoT technologies is impossible without solving the problems of interaction between systems. At the same time, it is no longer enough to ensure interoperability only at the technical and syntactic levels. Due to the increasing intelligence of the SIS and IoT, there is a need to ensure interaction at the semantic level of interoperability, assuming that the information system (device) understands the meaning of incoming requests and generates answers, which, in turn, will be correctly perceived by the system (device) that forms the request.

In [2, 3] it was noted that the IoT systems can be considered as SIS and thus the research results and standards related to SIS can be applied to the systems of the IoT. To ensure semantic interoperability, the authors of this article proposed an approach consisting in the implementation in each of the interacting SIS (IoT) using the reference model OSE/RM (Open System Environment/Reference Model) (see, for example,

IEEE P1003.22 Draft Guide for POSIX Open Systems Environment—A Security Framework) integrated knowledge base reflecting integrated ontologies of subject areas of interacting systems [3].

In the course of the study [7], the number of specific features of this type of ontology were determined. On the basis of the analysis of the structures and methods of ontology construction, it was concluded that it is expedient to include in the structure of the ontology used to provide semantic interoperability of is concepts that allow describing both the static state and the dynamic changes in the states of objects of the subject area. The main types of concepts used were identified, for each of which the composition of the required attributes (properties) characterizing it and possible types of relationships with other concepts were determined.

2 Analysis of Ontology Description Languages

Below we consider the languages of ontology description in order to choose the optimal one for ontology creation, which are planned to be used to provide semantic interoperability of is.

There are several "languages" for writing semantic models, the main ones being RDF, RDFS and OWL.

RDF – the metadata description language that forms the basis of Semantic Web, is a representative of the family to describe relational data models, the specificity of which is that resources and properties are identified by global identifiers - URIs. RDF describes the subject area in terms of resources, resource properties, and property values. RDF-data can be regarded as a set of statements (triplets) – "subject", "predicate (property)" and "object of approval", represented as a directed graph formed by such statements. RDF is a universal metadata description language and requires configuration for specific specialized tasks. The way to do this is to extend RDF with dictionaries, one of which is RDFS, which is used to describe ontologies.

RDFS introduces concepts such as classes, subclasses, properties, and subproperties, and allows you to restrict them. RDFS allows you to define resource classes and properties as elements of a data type dictionary, and specify which properties with which classes can be used. RDFS expresses these dictionaries by means of RDF, providing a set of predefined resources and properties with a designated semantic load for them, which can be used to describe new RDF dictionaries.

OWL extends the RDFS vocabulary (basically, the definition of classes and subclasses), by introducing relations of comparability classes (sameAs, differentFrom, equivalentClasses, etc.), characteristics of properties (inverseOf, TransitiveProperty, SymmetricProperty, etc.), restrictions on properties (e.g., what classes they belong to, or cardinality properties), intersection classes, etc. On the expressive properties there are three "dialects" OWL:

- OWL Full, which has the maximum expressive power, but does not guarantee the computability of logical conclusions in the ontology created with its help. For example, in OWL Full, classes can act both as a class and as an instance, which can lead

to contradictions in the description of the ontology and the inability to make logical inference based on the existing rules.

- OWL DL (Description Logic) guarantees computational completeness) the logical output is computable) and solvability (calculations are performed in finite time). OWL DL contains all OWL Full constructs, but their use is limited.
- OWL Lite has the least expressive properties, but can be used as an intermediate in the transition from simple taxonomies to ontologies.

Along with the advantages of OWL, which is the de facto standard for describing ontological models, we can note a number of its limitations. One of them is the inability to perform calculations based on fuzzy logic. A complex and ambiguous description in OWL is when the model needs to reflect an exception that clarifies a statement. For example, the fact that "birds fly" requires clarification, as not all species of birds fly.

3 Analysis of Tools for Creating Ontology Repositories

The simplest form of ontology storage is an OWL file. When such a file is read in memory, a model (a set of claims) is created and further work is done on it. The obvious drawback of this approach is the increase in memory costs, as well as a significant increase in the loading time of OWL files as the amount of metadata contained in the ontology increases. The need to use special language tools to extract metadata stored in ontologies necessitates the construction of ontology repositories based on DBMS.

From the point of view of structural features for the storage of ontologies best fit graph database. In this case, the graph vertices can be used to store ontology concepts, and the graph edges can be used to store relationships between concepts. Vertices and edges can contain arbitrary sets of attributes. An edge always has a start node, end node, type, and direction. Graph database management systems support methods for creating (Create), reading (Read), modifying (Update), and deleting (delete) data (CRUD) based on the graph data model. As a language means of data manipulation, SPARQL-like languages are used, the syntax of which is close to the SQL - language of queries to relational databases (for example, the query language *Cypher* in the graph database Neo4j). Graph DBMS began to be actively used with the development of social networks and is now widely used, for example, to solve problems related to the search for fraudulent and suspicious transactions in payment systems. In such tasks, it is important to quickly find the vertices associated with the source (for example, search for friends on social networks or accounts to which funds were transferred from a certain account).

Additional impetus for the use of graph DBMS was obtained by optimizing search queries (fast graph traversal algorithms) and creating indexes. Indexes help optimize the search for specific nodes. Although most queries to graph databases involve fetching the necessary information when traversing relationships between vertices, there are certain situations where you want to select specific nodes directly rather than by detecting them by crawling. For example, to identify nodes that serve as starting points for crawling, you need to find one or more specific nodes based on a specified combination

of attribute values (for example, select all social network members under 20). To increase the efficiency of searching nodes in graph DBMS, there are tools for creating indexes for combinations of labels and properties.

Let us consider the possibilities of relational DBMS as a mechanism for creating and storing ontologies. Traditionally, a disadvantage of relational DBMS when working with graph data models was the lack of tools for implementing hierarchical queries. For example, to find vertices connected to a given vertex through any other vertices, it required numerous JOIN operations, which significantly slowed down the execution of queries. To solve this problem, SQL-1999 provides the ability to create a temporary view that is described by the WITH statement compared to SQL-92. After that, the data is selected from it with a simple SELECT command. A number of DBMS have their own means to implement these requests. For example, in ORACLE hierarchical queries are implemented very efficiently using the CONNECT BY command, in MsSQL-2008 you can use a set of standardized stored functions to work with hierarchy levels (*Hierarchyid*).

Summarizing the above, it can be noted that relational DBMS compared to graph DBMS have a higher efficiency of data search by values superimposed on attributes (the task of selecting graph nodes directly, without taking into account relationships); graph DBMS more effectively implement queries that take into account the relationship between the vertices.

The article [8] describes an approach for semantic data search, taking into account the advantages of both graph and relational DBMS. In the form of a graph in this case, a set of interrelated concepts that reflect the semantics of the subject area is stored. Data is stored in relational tables that contain a significant number of records. Initially, the query selects the vertices of the graph according to the specified conditions imposed on the relationships between concepts. The selected vertices contain the attribute values by which records are searched in relational tables. Search on graph model is organized by means of Neo4j DBMS, search of records in relational tables - by means of MySQL DBMS.

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

Separate consideration in this case requires ORACLE database, which today can be attributed to the class of object-relational. ORACLE 11g implements mechanisms United by the term Semantic Technologies (semantic technologies). This version provides the ability to export and import OWL structures and supports the OWLPrime ontology description language, which is a subset of the above OWL DL and includes the following features:

- Ontology structure creation (class, subclass, property, subproperty, domain, range, type);
- Specify characteristics of properties (transitive, symmetric, functional, inverse functional, inverse);

- Comparison of the classes (equivalence and disjointness);
- Comparison of the properties of (equivalence);
- Entity comparisons (same, different);
- Setting property restrictions (hasValue, someValuesFrom, allValuesFrom).

To support OWLPrime, more than 50 rules are implemented that are used in the logical inference process. A rule consists of a condition ("if"), a filter (condition), and an output ("then"). In ORACLE 11g added the ability to configure custom rules using a language OWLIF (structures IF-THEN). You can set specific restrictions on the rules that a user can create. For example, you can specify that the user can create only logical output within the subClassOf hierarchy on the system, and you can limit the number of output steps.

Requests to extract information from ontologies in ORACLE 11g are made using SPARQL. The SEM_RULEBASES construct is used to connect output rules created by the user in SPARQL queries.

4 Summary

The paper shows the possibility of using graph and relational DBMS to create ontology repositories

Based on the above, it can be concluded that currently it is most appropriate to use DBMS ORACLE version 11g to create ontology repositories used to ensure interoperability of IS. However, it should be noted that to work with ontologies of this type, prebuilt inference rules (as mentioned above) will not be used as effectively. This is due to the fact that they are more focused on the work with ontologies, the tops of which are connected as a class (class) and subclass (subclass). In the ontology that can be used to provide interoperability, as shown in [7], there are peaks, e.g., the types of "Association" and "Action". In this case, the processing of such vertices requires writing custom output rules, which were mentioned earlier.

As a positive trend, it should be noted that at present both relational and graph DBMS have mechanisms for working with ontological structures. To date, the greatest development, as indicated, such mechanisms were in the database ORACLE 11g, which was given the name Semantic Technologies. It can be predicted that these mechanisms will continue to develop and at some point it may be more convenient to create ontology repositories by means of other DBMS (for example, Neo4j) or to use the technology of sharing graph and relational DBMS [8].

References

1. Schwab K. The Fourth Industrial Revolution: what it means, how to respond, World Economic Forum, (WEF-2016) https://www.weforum.org/agenda/2016/01/the-fourth-industrial-revolution-what-it-means-and-how-to-respond/, last accessed 2019/03/30.

2. Boychenko A., Korneev D., Kozlova O. Approach to solving the problem of ensuring interoperability of services. CONFERENCE Enterprise Engineering and Knowledge Management (EEKM-2013), pp. 198-206 (2013).
3. Boychenko A., Korneev D. Description of interoperability technology using the OSE/RM model. CONFERENCE Enterprise Engineering and Knowledge Management (EEKM-2014), pp. 212-218 (2014).
4. ISO/IEC 24765-Systems and Software Engineering-Vocabulary, http://www.cse.msu.edu/~cse435/Handouts/Standards/IEEE24765.pdf, last accessed 2019/03/30.
5. EIF - European Interoperability Framework, http://ec.europa.eu/idabc/en /document/ 2319/5938.html, last accessed 2019/03/30.
6. ISO/IEC 42010 Systems and software engineering - Recommended practice for architectural description of software-intensive system, https://www.iso.org/standard/ 45991.html, last accessed 2019/03/30.
7. Boychenko A., Korneev D., Kazakov V. Development of ontology structure for semantic interoperability of information systems. CONFERENCE Enterprise Engineering and Knowledge Management (EEKM-2018), pp. 163-172 (2018).
8. Boychenko A., Korneev D., Kazakov V. Data search based on semantic model. CONFERERENCE Enterprise Engineering and Knowledge Management (EEKM-2015), pp. 123-132 (2015).