

Irina KUSHNIRETSKA^{}, Oksana KUSHNIRETSKA^{**},
Andriy BERKO^{***}*

DESIGNING OF STRUCTURAL ONTOLOGICAL DATA SYSTEMS MODEL FOR MASH-UP INTEGRATION PROCESS

Abstract

This paper describes the construction of structural ontological data systems model in the process of data integration. The usage of ontologies in the context of improving the process of dynamic semantic data integration has been characterized. The algorithm of constructing of structural ontological data integrated systems model has been designed. This algorithm is based on the rules of using and application of ontological modeling. The algorithm as a sequence of five steps describing tasks for each of them has been presented.

1. INTRODUCTION

Today, in the active development of new and improved information technology era increases the need also of improving the technology and work with web-systems. Solution of this problem is the development of integration systems that use of Mash-Up technology. Mash-Up is an approach to application development that allows users to combine data from multiple sources into one integrated tool [1].

Mash-Up system work technology is to dynamically integrate data from different web-systems data sources. In order to integrate the some application to Mash-Up system, it is necessary first of all to know the structure of the application or system that is integrated. And knowing the structure of each input web-system we are faced with the task of creating of the common structural model of systems that are integrated.

^{*} Lviv Polytechnic National University, Ukraine, 79013, Lviv, Bandera str., 28a, presty@i.ua

^{**} Lviv Polytechnic National University, Ukraine, 79013, Lviv, Bandera str., 28a, presstail@i.ua

^{***} Lviv Polytechnic National University, Ukraine, 79013, Lviv, Generala Chupryny str., 130, berkoandriy@yandex.ua

2. THE PROBLEM FORMULATION

Data sources can have different properties, essential for the choice of methods Mash-Up Data Integration - they can support the presentation of data in terms of a data model, can be static or dynamic, etc. Many sources of data integration can be homogeneous or heterogeneous with regard to the characteristics that respective of used integration level [1].

To ensure efficient search, web application must clearly understand the semantics of documents submitted in web. In this regard, one can observe the rapid growth and development of technologies Semantic Web, which is currently available. The W3C developed a concept based on the active use of metadata markup language XML, RDF language (Resource Definition Framework) and ontological approach. All the proposed tools allow data sharing and their reuse.

Existing methodologies and tools for building software systems are focused on well-structured problem with sufficiently formalized of subject areas and permanent local of knowledge source. So, the actual problem today is to develop mechanisms that can take into account the peculiarities of different input information systems, the knowledge specific required for the subject area forming and the distributed nature of their origin. In turn, the domain models must have of the internal mechanisms of domain model dynamic adaptation for the whole system lifetime.

Hence, the aim of this work are:

- to investigate the using of ontologies and ontological modeling principles in the context of the process improving of dynamic data semantic integration;
- to develop an algorithm of constructing of structural ontological data integrated systems model.

3. THE BASIC MATERIAL PRESENTATION

3.1. The usage of ontologies for data integration

One promising avenue of research is the use of ontologies for data integration task solving [3].

In [11] is the definition of information component ontological specification as a set of definitions and concepts and also rules (axioms), relating to definitions and concepts of the domain (application context).

The term “ontology” is now used in two contexts:

- philosophically: ontology is a system of categories used to consider taking into account the specific vision of the world [12];

- In the context of information systems, ontologies is formal description of the conventional understanding of some domain, through which people can communicate with a computer systems [12]. Software components use ontologies to interact with each other as part of an integrated system of heterogeneous resources. It is anticipated that ontology is independent of the domain representation language.

Data integration methods based on ontology has shown in practice to be effective, but building the ontology requires of expert knowledge in studies subject area and can take a significant amount of time [2, 3]. Therefore, an important task is to develop methods and algorithms for automating the process of building the ontology.

Ontological system built on the basis the following principles [3]:

- formalization, that is the description of objective reality elements using a single, strictly defined samples (terms, models, etc.);
- the using of a limited number of basic terms (entities) on which all other concepts construct;
- internal completeness;
- logical consistency.

Recently, more and more prevalent becomes use of ontologies for modeling of automated information systems domains [4, 5]. The most commonly such approach is used for intelligent systems [6], especially systems designed for the operation in the Internet. This is due to the fact that ontological model allows us to develop a metadata model, which greatly improves the system use of a wide range of users in terms of interaction organization, especially if that is dynamic Mashup system.

In general, formal presentation of data ontology is the following [8]:

$$O = \langle X, R, F \rangle \quad (1)$$

where: X – finite set of domain concepts with their properties (attributes),
 R – finite set of relations (relationships, correlations) between concepts,
 F – finite set of interpretation functions (restrictions, axioms) [8].

In accordance with the requirements of IDEF5 standard [9], the concepts are divided into classes and value classes. Relations between concepts are divided into classification relations (between classes and subclasses) and structural relations (links that describe the interaction of classes).

The authors of [10] define ontology as a cortege:

$$O = \langle C, I, R, T, V, \leq, \perp, \in, = \rangle \quad (2)$$

where: C – set of classes,
 I – set of instances of classes,
 R – set of relations,
 T – set of data types,
 V – set values (set C, I, R, T, V pairwise disjoint),
 \leq – relation to $(C \times C) \cup (R \times R) \cup (T \times T)$, called specialization,
 \perp – relation to $(C \times C) \cup (R \times R) \cup (T \times T)$, called exception,
 \in – relation to $(I \times C) \cup (V \times T)$, called realization (instantiate),
 $=$ – relation to $I \times P \times \cup (I \cup V)$, called assignment.

Semantics of languages ontology are usually presented through models theory. In particular, it defines the interpretation function that maps each element of ontology at some particular set, called the interpretation domain.

The ontology interpretation (2) there are the couple $\langle I, D \rangle$, where D is the region of interpretation, and I – interpretation function such that:

$$\begin{aligned} \forall c \in C, I(c) &\subseteq D \\ \forall r \in R, I(r) &\subseteq D \times (D \cup V) \\ \forall i \in I, I(i) &\subseteq D \\ \forall t \in T, I(t) &\subseteq V \\ \forall v \in V, I(v) &\subseteq V \end{aligned} \quad (3)$$

About contention expressed of the ontological language, saying that it is satisfied of interpretation, if interpretation agrees with this contention.

The interpretation for ontology (2) is model $m = \langle I, D \rangle$, satisfying all contentions ontology o : $\forall \sigma \in o, m \models \sigma$.

3.2. Designing of general structural ontological data systems model, as part of incoming information resources structure and content determining process

In [7] is described how to determine of the structure and content of incoming information resources to solve the problem of dynamic semantic data integration. This process, according to [7], consists of five steps:

1. Determining of the input data presentation form: structured data, semi-structured data or unstructured data.
2. Classification of input data according to the subject area with allocation and preservation of data semantics.
3. Allocation in the input information resource of the attributes set, that reflecting its main characteristics and aspects of that domain.
4. Setting clear boundaries and meet the basic restrictions related to the web-information input stream.
5. Formation of the input information resource model of general structure defined subject area.

Considering the problem which you need to solve in the second step when is the automatic information classification, we propose in first of the structure ontology automatically modeling of each of integrated applications. The system must perform this work after the sources selecting for Mashup. Due to the construction of the complete structural information meta-model that will combine all systems elements with their relationship we can thus carry out the procedure of input data classification, while retaining the semantics of the data. Thus, at this step, you need to solve the following problem:

1. Integrated system structural ontology. Retrieving information about the structure of each of integrated information systems in an ontological format.
2. The total structural ontology of all integrated systems. Combining the obtained structure ontologies in a general structural ontological information model.
3. The global integrated systems meta-model.

But only solving the first two problems can be fully automated, using the appropriate standard tools and technologies. But the implementation of the third task requires the participation of experts in web-systems integrating field and specialists of knowledge presentation in the form of ontologies. Therefore, an important task is to develop algorithms of designing of global meta-model of combined dynamic data set that has a general structure and unique content. According to solve this problem, we propose an algorithm of constructing of structural ontological data integrated systems model

3.3. The algorithm of constructing of structural ontological data integrated systems model

In any web-system all information is stored in databases. There are relational databases. In relational databases information about the structure and relationships between the structural elements stored in data schemes, and these schemes must be obtained in the work of the algorithm. However, the scheme analysis can only to provide structural interoperability. To achieve semantic interoperability when

data scheme extraction should also take into account of the semantic assignment of these elements, so we need to use domain ontology. This ontology for the resulting model will add connection between concepts in the subject area. Thus, each ontological model obtained from a system database will a subset of the domain ontology.

When building any algorithm, priority is to determine the input and output data. The input data for the algorithm constructing a structural ontological data integrated systems model are: structure schemas of integrated systems database and domain ontology.

Let O_G – general ontology schemas of all data of integrated systems:

$$O_G = \{O_i, O_D\} \quad (4)$$

where: O_i – ontology of structure of system,
 O_D – domain ontology.

Domain ontology usually developed previously, with experts from the domain and specialist of knowledge presentation in the ontological format. The process of creating a model takes a long time, but it is necessary only at the initial stage of integration. With further addition of new systems operating in this area, the very ontology does not require additional changes.

For describing the work of algorithm of constructing of structural ontological data integrated systems model we introduce some notation concepts.

Let we have any system database schema: S .

$$S = \{T_1, \dots, T_m\} \quad (5)$$

where: T_1, \dots, T_m – the system S database schema tables.

$$T_i = \{A_1, \dots, A_k\} \quad i = \overline{1, n} \quad (6)$$

where: A_1, \dots, A_k – database schema tables attributes.

$$R = \{R_1, \dots, R_z\} \quad i = \overline{1, n} \quad (7)$$

where: R_1, \dots, R_z – relationships between ontology concepts.

Algorithm of constructing of structural ontological data integrated systems model contains 6 steps:

1. Database structure presentation in RDF format (the consistent mapping S schema in RDF format).

$$T_i \rightarrow T(RDF)_i, A_j \rightarrow A(RDF)_j, \quad i = \overline{1, n}, j = \overline{1, k}, \quad (8)$$

where: $T(RDF)_i$ – ontology concepts described by RDF,

$A(RDF)_j$ – ontology concepts properties.

2. Adding of semantic properties and ontology creating. This step is realized through the using of the procedure of identify the common features of database elements and adding of the links between them.
3. Adding the upper level ontology and domain ontology. We realize this step through OWL language, using the command owl: import. Due to the transitivity rule in RDF, additional ontologies expand the domains and add the new concepts and properties.
4. Checking of the created ontology. This step is realized by checking implementation and analysis as far as extracted ontology is “connectedness”. That is, we check whether lack of nowhere semantic relations. If so, go to the fifth step - if not, go to the sixth step.
5. Editing of the extracted ontology using ontology editor (Protégé) and adding links between concepts. Then return to step 4.
6. Storing of the resulted common ontology structure in a file or metadata repository in RDF format.

The flowchart of the algorithm we are showing in Figure 1.

Let us consider the steps of the presented algorithm (fig. 1). In the first step is mapping of data scheme structural elements in RDF format. The main elements of the relational database to be displayed are tables and their attributes. Attributes (fields tables), also have of the important structural information, such as name and type of attribute. This information is obtained using SQL and foreign key mechanism.

When there is data schema analyzing, tables names are automatically the names of new classes, and tables fields names are properties associated with their class. Also, you can separately record of the information about a database tables class RDF matching in a separate XML document or other key-value storage type. Since, according to the proposed method, each extracted proto-entity must have an identifier, the table being analyzed must have a primary key. In its absence, it is necessary to create during table processing. So, the unique proto-entity ID will compose with the table and the primary key.

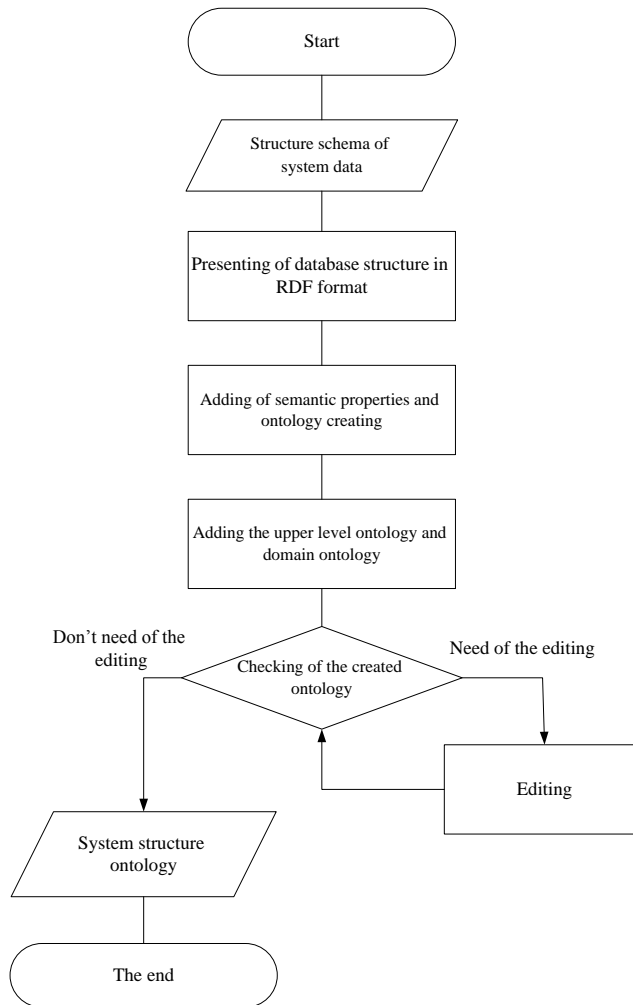


Fig. 1. The flowchart of the algorithm of constructing of structural ontological data integrated systems model [source: own study]

Suppose we have the following table:

Tab. 1. Table Product

Field name	Data type
ID	int
Name	varchar
Price	float
Quantity	int
Stock	int

Presented above table 1 matches to the following:

```
TABLE Product (  
ID INT UNSIGNED NOT NULL AUTO_INCREMENT,  
Name VARCHAR(20) NOT NULL,  
Price FLOAT,  
Quantity INT,  
Stock INT,  
PRIMARY KEY (ID)  
);
```

For absolute identification when the table is displayed in the class, when the table fields names are converting in the class properties are transformed in the following way:

$$\text{PropertyOfClass} = \text{TableName} + _ + \text{FieldName} \quad (9)$$

Hence, a class that is a reflection of Product table is (according to the object-oriented programming notation):

```
class Product {  
private int product_id;  
private String product_name;  
private double product_price;  
private int product_quantity;  
private int product_stock;  
public String getProduct_name () {  
return product_name;  
}  
}
```

In this class is defined of the `String.getProduct_name()` method, which returns the name of the product. Also, we see that are transformed not only of the fields names in the table, but also their types. This is a very important task in the database schema analysis. Different systems may use completely different relational database, which in turn can use various data types for the numbering of stored resources. Upon database structure extracting, you must also extract of the information about of the stored data types in its elements and describe them in RDF. Due to the fact that the RDF structure used for modeling is based first on the markup language XML, ontological properties described thus may have different XSD data types. XSD language is a standard language for describing XML documents. When is used XSD you can create a set of rules to be met XML document. This language has of a several primitive data types you can use to describe the item as in XML, and in RDF document.

The result after the first step of the algorithm is an RDF-document contains statements that describe structural information from the database system schema. We show an example of RDF-document which have one class and meet our table Product, described above:

```
<sys:product>
a owl:Class;
rdfs:label "product"^^xsd:string.
<sys:product_id> a owl:DatatypeProperty;
rdfs:domain <sys:product>;
rdfs:label "ID"^^xsd:int.
<sys:product_name> a owl:DatatypeProperty;
rdfs:domain <sys:product>;
rdfs:label "Name"^^xsd:string.
<sys:product_price> a owl:DatatypeProperty;
rdfs:domain <sys:product>;
rdfs:label "Price"^^xsd:double.
<sys:product_quantity> a owl:DatatypeProperty;
rdfs:domain <sys:product>;
rdfs:label "Quantity"^^xsd:int.
<sys:product_stock> a owl:DatatypeProperty;
rdfs:domain <sys:product>;
rdfs:label "ProductStock"^^xsd:string.
```

The example shows how the fields of the table Product are transformed into owl: DatatypeProperty properties with XSD data types. Each property is associated with the sys: product class through rdfs: domain property.

In the second step of the algorithm is used of the procedure of the automatically determine of the common elements in the structure of integrated schemas and identify relationships between them. The main goal in the second step is to increase the number of semantic relations between the different systems ontologies within the overall global ontology. In other words, automatically analyzing of all the structural elements of each system can identify these items and add to them of the semantic property that will help tie them together. Thus, in the second step we need two problems solving: identification of common elements in the integrated schemas structure and adding of links between similar elements.

In [13] are discussed some annotative properties that represent of the classes and properties in general ontology and connect them with other objects in the model.

In the third step is the replenishment of the resulting ontology by the additional upper-level ontologies for the integrated ontological model getting. This replenishment carry, using the owl: import command, which is responsible for importing concepts and relations from external ontologies. Currently, there are much different domains ontology that contain of the different set of concepts and

relationships between them. For example, there are ontologies describing of the relationships between people, ontologies describing of the bibliographic documents, etc. Importing such properties in the resulting structural ontological model, you can get information about the system not only within their domain, but also abroad. Thus, you can access even more options for automated logical decision making, due to the large amount of information objects within the ontological model.

The fourth step involves the checking of the created ontology to lack of relationships between ontology objects. Because in the third step execution during of the created ontology automated replenishment cannot always be found all relationships between the system elements and some links may be missed and not installed. Therefore, we must to do of the checking and if all is well, then go immediately to step six, if not – to step five.

The fifth step of the algorithm provides of the additional manually links establishing between objects ontology. Links can be established between ontologies integrated systems objects and between imported upper level ontologies and domain ontology. To implement this step, you must use the ontology editing software such as Protégé. In this step, we add of the necessary links manually to create a complete semantic model of the integrated systems.

And finally, in the sixth step is the getting of general ontology structure in the RDF form. The result can be written to a file or to a specialized data repository RDF. The resulting ontological model, in fact, is a conceptual part of the ontology. It contains concepts and links between them in integrated systems within a subject area in which they work, and the set of terms from other areas.

3. CONCLUSIONS

The usage of ontologies and ontological modeling principles in the context of improving the process of dynamic semantic data integration has been presented. The process of designing of the algorithm of constructing of structural ontological data integrated systems model has been described. The flowchart of the algorithm of constructing of structural ontological data integrated systems model and in detail each step of the algorithm has been considered.

On the basis the common global structure model getting as a result of the algorithm work we can get metadata of systems information resources. Using this approach can automatically to describe of the resources semantics at the very early stage of their receipt.

REFERENCES

- [1] KUSHNIRETSKA I., BERKO A.: *Application of MashUp Technology for Dynamic Integration of Semi-Structured Data*. Proceedings of the Sixth International Conference of Young Scientists CSE-2013, Lviv Polytechnic National University Publisher, Lviv, 2013, pp. 220–221.
- [2] WACHE H., VOGELE T., VISSER U., STUCKENSCHMIDT K.: *Ontology-Based Integration of Information - A Survey of Existing Approaches* IJCAI-01 proceedings of the Seventeenth International Joint Conference on Artificial Intelligence, Seattle, Washington, 2001, pp. 64–67.
- [3] GUARINO N.: *The Role of Ontologies in Information Systems Design*. Proceedings of the First International Conference of Formal Ontologies (FOIS'98), Italia, 1998, pp. 56–58.
- [4] USCHOLD M., GRUNINGER M.: *Ontologies: Principles, Methods and Applications*. Knowledge Engineering Review, 11 (2), 1996, pp. 77–79.
- [5] GAVRILOVA T.: *Ontologicheskyye Engineering*. Proceedings of the Eighth Scientific and Practical Conference “Reengineering of business processes based on modern information technology, Knowledge Management System” (RBP-KMS-2005), Moscow, 2005, pp. 79–82.
- [6] LAPSHIN V.: *Ontology in computer systems*. “The scientific world” publisher, Moscow, 2010, p. 222.
- [7] GRUBER T.: *The role of common ontology in achieving sharable, reusable knowledge bases*. Proceedings of the Second International Conference “Principles of Knowledge Representation and Reasoning”, 1991, pp. 601–602.
- [8] KONSTANTINOVA N., MITROFANOVA O.: *Ontology as knowledge of the storage system* [electronic resource]. – Available on: [http://www.sci-innov.ru/icatalog_new/index.php?action=send_att&entryid=68352&fname=68352e2-st08_\(Митрофанова О.А.\).pdf](http://www.sci-innov.ru/icatalog_new/index.php?action=send_att&entryid=68352&fname=68352e2-st08_(Митрофанова О.А.).pdf)
- [9] SMIRNOV A., PASHKIN M., SHILOV N., LEVASHOVA T.: *Approach to building of the intelligent decision support distributed systems in an open information environment*. St. Petersburg Institute of Informatics and Automation, St. Petersburg, 2007, pp. 145–153.
- [10] BAADER F., CALVANESE D., MCGUINNESS D., NARDI D., PATEL-SCHNEIDER P.: *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2003, pp. 132–136.
- [11] EUZENAT J., SHVAIKO P.: *Ontology matching*. Springer-Verlag Heidelberg, Berlin, 2007, p. 332.
- [12] KUSHNIRETSKA I., KUSHNIRETSKA O., BERKO A.: *Determination of the structure and content of input information resources for Mashup system*. Technology Audit and Production Reserves, Vol.20, No. 6/3, 2014, pp. 4–9.
- [13] *OWL Web Ontology Language Test Cases* [electronic resource]. – Available on: <http://www.w3.org/TR/owl-test/>.