Semantic data integration methods based on ontologies in intelligent business analytics systems

Victoria Vysotska[1,†], Andrii Berko[1,*,†], Lyubomyr Chyrun[2,†], Sofia Chyrun[1,†], Valentyna Panasyuk[3,†], Ihor Budz[1,†], Iryna Shakleina[1,†], Olga Garbich-Moshora[4,†] and Ivanna Andrusyak[1,†]

[1] *Lviv Polytechnic National University, Stepan Bandera 12, 79013 Lviv, Ukraine*
[2] *Ivan Franko National University of Lviv, University 1, 79000 Lviv, Ukraine*
[3] *West Ukrainian National University, Lvivska 11, 46004 Ternopil, Ukraine*
[4] *Ivan Franko Drohobych State Pedagogical University, Ivan Franko 24, 82100 Drohobych, Ukraine*

Abstract

The basis of the data integration service of cognitive systems is the linguistic means of describing integrated data (Integrated Data Framework - IDF). IDF is a specialized language for creating metadata that describes the properties of a data set and the order of its application in integration processes. The syntax of IDF is based on the XML syntax, which provides the possibility of processing language constructions by standard means. The peculiarity of the means of integration is that, unlike known technologies, they are implemented not at the instrumental level, but at the protocol level. This solution makes it possible to perform data integration using an intermediate layer, on which actions are performed independently of the platforms and technologies of implementation of other components of the open information system. In the course of the work, an analysis of existing approaches to the specification of formal description languages, in particular UML, OWL, XACML, etc., was carried out. The model description language is based on XML, as a flexible tool that allows you to expand the syntax and semantics of the language and is based on the vast majority of modern languages used in ontological modelling. The scope of application of the model description language is determined by the main variants of its use.

Keywords

intelligent system, semantics, ontology, data integration, content integration, business analytics

CLW-2024: Computational Linguistics Workshop at 8th International Conference on Computational Linguistics and Intelligent Systems (CoLInS-2024), April 12–13, 2024, Lviv, Ukraine

* Corresponding author.
† These authors contributed equally.

victoria.a.vysotska@lpnu.ua (V. Vysotska); andrii.y.berko@lpnu.ua (A. Berko); Lyubomyr.Chyrun@lnu.edu.ua (L. Chyrun); sofiia.chyrun.sa.2022@lpnu.ua (S. Chyrun); v.panasiuk@edu.wunu.ua (V. Panasyuk); ihor.s.budz@lpnu.ua (I. Budz); iryna.o.shakleina@lpnu.ua (I. Shakleina), garbich79@gmail.com (O. Garbich-Moshora); andrusyak.ivanna@gmail.com (I. Andrusyak)
0000-0001-6417-3689 (V. Vysotska); 0000-0001-8653-1520 (A. Berko); 0000-0002-9448-1751 (L. Chyrun); 0000-0002-2829-0164 (S. Chyrun); 0000-0002-5133-6431 (V. Panasyuk); 0000-0002-5400-0984 (I. Budz); 0000-0003-0809-1480 (I. Shakleina);0000-0002-3172-5499 (O. Garbich-Moshora); 0000-0001-6601-4374 (I. Andrusyak)

1. Introduction

Semantics is an inherent property of data that ensures its meaningfulness and the possibility of using data for its intended purpose. In general, semantics is defined by a set of correspondences between formal notations and real subject areas (SA) concepts of an information resource, which allows unambiguous interpretation of data at various stages of working with them. In modern information technologies, several different means of determining the data semantics are used. The simplest means of data interpretation is the interpretation of its schema, for example, the description of table columns, XML tags, document sections, etc. In large systems that use integrated information resources, there are more complex means of determining the semantics of data, in particular, thesauruses (dictionaries of data), metadata and ontologies [1-3]. The main theoretical foundations, principles and provisions of semantic integration (integration of semantics) of data are formulated, in particular, in [3-9]. In [10-15], the concept of semantic integration is substantiated, as well as the methods of contextual analysis and the application of ontologies as a tool for building an integrated content space, the concept of the conditions of semantic integration of data and the concept of semantic distance between data sets are defined.

2. Related works

The basic principles of the formation and application of metadata as a means of determining the semantics of data are outlined in [11-15]. In particular, the Zachman scheme for the metadata organization is defined. This scheme provides for the formation of the metadata composition according to certain principles, which require the definition of such elements as metadata about the data object, data subject, time indicators, the location and application of data, the purpose of data and the method of their use. In [16], a scheme for constructing a description of the information resource "Dublin core" (named after the city of Dublin, Ohio, USA, where the working group to create the corresponding standard worked) is described. Dublin Core is a metadata standard (format), a simple and effective set of values for describing a wide range of information resources of various types. Dublin Core Semantics was created by an international interdisciplinary group of specialists in computer science, the Internet, librarianship, text coding, museum science, and other related fields.

The Dublin Core standard is divided into two levels [16]:

- simple (unqualified), which includes 15 elements;
- competent (qualified), consisting of 18 elements and a group of qualifiers that clarify the semantics of the elements to improve the quality of the search for information resources.

The basic set of metadata elements of the simple Dublin core (Dublin Core Metadata Element Set; DCMES) consists of 15 units: *Title*; *Creator*; *Subject*; *Description*; *Publisher*; *Contributor*; *Date*; *Type*; *Format*; *Identifier*; *Source*; *Language*; *Relation*; *Coverage*; *Rights*.

[16] show the possibilities of combining the principles of metadata construction based on the Dublin core with such means of describing the semantics of information resources as RDF-XML and OWL.

The application of ontologies in the processes of semantic data integration is described in [5-18]. Ontology makes it possible to define a set of concepts and correspondences between them without the subject area specification. In general, ontologies form the infrastructure necessary to define the semantics of data, which can be adequately perceived and processed both by software and by humans. This approach allows you to correctly solve problems related to the data content at the formal level using information technology tools. The difference between an ontology describing an information resource and metadata, such as, for example, the Dublin Core, is, at first glance, subtle but significant. Although both tools are used for the semantic integration of data, the fundamental difference between them lies in the degree of human participation in the integration processes. Metadata, in general, is created, edited and interpreted by people, so subjective factors, in particular, limitations on the complexity of their presentation and understanding, are a decisive factor. In contrast to traditional metadata, ontology is a basic formal model of means of integration of information resources and implementation of various additional functions. As a result, the ontology's use allows us to operate with more complex and formalized concepts that are often beyond human competence. Data ontology is considered a means of versatile and detailed formalization of knowledge about data using a conceptual scheme. As a rule, the composition of such a scheme includes a description of the data structure, which contains definitions of all relevant classes of objects, their relationships and rules (theorems, restrictions) specified in the subject area of the data set [18-21]. Data ontologies are described by various means, and today there are quite a few languages for describing ontologies. However, given that in any ontology terms are defined and logical connections between them are set, the semantics of describing terms and connections in different languages will be the same. Ontological systems are built based on the following principles [21-27]:

- formalization, i.e. description of objective elements of reality using single, clearly defined samples (terms, models, etc.);
- use of a limited number of basic terms (entities), based on which all other concepts are constructed;
- internal completeness and logical consistency.

Unlike an ordinary dictionary, an ontological system is characterized by internal unity, logical interrelationship, and consistency of the concepts used [28-33]. One of the directions in the research of methods and means of processing data based on ontologies, which are actively developing today, is the direction related to the use of Web-ontologies [24-28]. The tools created in this field, such as XML-RDF (Resource Definition Framework) [23-33], and OWL (Web Ontology Language) are to some extent suitable for solving problems of semantic data integration. The main tasks performed in this part of the work are [34-42]:

1. generalization and classification of approaches to the semantic integration of data at different levels of their presentation and perception to determine the most acceptable for use in open systems environments;
2. development of basic provisions and principles of integration of data semantics in open information systems for their further implementation in appropriate means;
3. determination of the order and place of application of semantic integration in the processes of forming the content of the global integrated information resource of open systems;
4. development of formal requirements for the semantic integration of disparate data to avoid contradictions and conflicts in their content and the formation of an agreed result.

3. Models and methods

3.1. Means of semantic data integration

The integration of data semantics involves the formation of a single content space for the perception, interpretation and application of data regardless of their presentation format and structure. One of the main problems in this process is the formation of requirements for the semantic integration of data sets, with the help of which it is possible to assess the possibility or impossibility of combining their content [25] The most famous approaches to semantic integration today are the following [42-45]: integration based on metadata; contextual semantic integration; integration based on ontologies.

Semantic integration based on metadata. This approach involves comparing the composition and content of metadata of two sets to determine the possibility of their semantic integration. Metadata provides the formation and application of a description of the main properties of a set of data (information resource), in particular, those that determine its semantic characteristics. One of the common ways of creating metadata is the dimension scheme of Zachman [26], which provides for the creation of six categories of metadata (dimensions) that describe the following IR properties [46-54]:

- data objects – description of entities that associate with values from the data set;
- data subjects – description of persons who create or use data;
- time indicators – a description of time points or intervals characterizing the creation, maintenance and use of data;
- data placement – a description of the location of the data and the methods and order of access to the resource;
- purpose of data – description of functions and tasks that use the information resource;
- the procedure for using data - rules and restrictions on working with the information resource.

The general structure of information resource metadata, built according to the Zachman scheme, is presented in Fig. 1. Each of the dimensions of metadata is a certain set of values that characterizes one of the aspects of organization, perception and application of a certain set of data, in particular, in the processes of semantic integration with other resources.

Set of metadata – MD_i of some data set D_i is given as:

$$MD_i = < MD^o_i, MD^s_i, MD^p_i, MD^t_i, MD^g_i, MD^m_i >, \tag{1}$$

where MD^o_i is metadata about data objects; MD^s_i is metadata about data subjects; MD^p_i is metadata about data placement; MD^t_i is metadata about time data; MD^g_i is metadata about the purpose of using the data; MD^m_i is metadata about the order of data use.

The coincidence of metadata values of two sets according to defined dimensions forms a condition of semantic integration. The degree of coincidence of metadata values of a certain category depends on specific tasks, but, as a rule, its numerical expression should not be less than 80%. Determining the categories of metadata that form the conditions of semantic integration and the procedure for determining their coincidence is a task that is difficult to formalize and often requires the participation of an expert. This factor significantly limits the possibilities of using such metadata as a means of operational semantic integration in open information systems. An alternative method of organizing metadata to the Zachman scheme is the Dublin core. The main principles of building a metadata system based on the Dublin core are simplicity, clear semantics, internationalization, extensibility, ambiguity, simplified representation of metadata, correctness of values, connection with syntax, and use of standard namespaces.

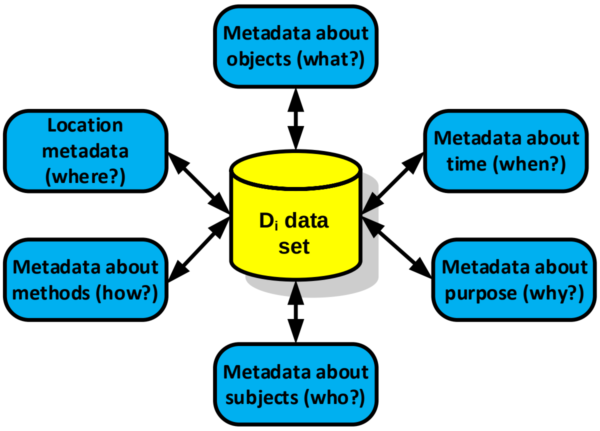

Metadata about objects (what?)
Location metadata (where?)
Metadata about time (when?)
D$_i$ data set
Metadata about methods (how?)
Metadata about purpose (why?)
Metadata about subjects (who?)

Figure 1: Zachman scheme of metadata organization

The use of such a list of information resource properties allows formalizing and unifying the description of the data to be integrated, in particular, their semantic content. Metadata built according to the principles of the Dublin core have the following features (Fig. 2):

- make it possible to create a description of any kind of information resource - from an art book to web pages, electronic documents and DB;
- provide a complete and comprehensive description of all data properties, in particular, those that determine their semantics;

- it is possible to display metadata in XML format, which greatly simplifies the processes of their formal processing.

An example of a set of metadata describing an information resource that contains the text of a scientific article built according to the principles of the Dublin core in the format of an XML document is shown in Fig. 2.

Formally, the set of metadata of some information resource *D*, built according to the principles of the Dublin core, will be presented in the form of a combination of description elements provided by the standard:

$$MD^{DC}(D) =\{MD_1^{DC}(D), MD_2^{DC}(D), \dots , MD_{15}^{DC}(D)\}, \tag{2}$$

where $MD^{DC}(D)$ is a set of metadata; $MD_k^{DC}(D)$, k=1,2,...,15 value of the corresponding element of the base set of the Dublin kernel.

```
<?xml version="1.0"?>
<!--  An example of metadata built according to the principles of the Dublin core -->
<metadata xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:dcterms="http://purl.org/dc/terms/">
<!-- Resource name --><dc:title> Semantic data integration </dc:title>
<!-- Resource author --><dc:creator> Andriy Berko  </dc:creator>
<!-- Surname of the author in his native language -->
<dc:creator xml:lang="UA"> Андрій Берко </dc:creator>
<!-- Subjects according to UDC -->
<dc:subject xsi:type="dcterms:UDC">  UDC 004.652 </dc:subject>
<!-- Topics in Ukrainian -->
<dc:subject xml:lang="UA"> моделі баз даних </dc:subject>
<!-- Опис змісту ресурсу українською мовою -->
<dc:description xml:lang="UA">
У роботі розглянуто моделі семантичної інтеграції баз даних та даних довільного формату на основі метаданих,
контекстного аналізу та онтологій
</dc:description>
<!-- Resource creation date --><dc:date> 01.10.2023  </dc:date>
<!-- Resource type --><dc:type> article </dc:type>
<!-- Presentation format --> <dc:format> MS Word 2019 </dc:format>
<!-- Resource identifier --><dc:identifier> stattia_10.2023.docx </dc:identifier>
<!-- Data source -->
<dc:source>http://Andriy.Berko.Lviv.net/stattia_10.2023 </dc:source>
<!-- Resource presentation language --><dc:language> UA </dc:language>
<!-- End of metadata -->
</metadata>
```

Figure 2: Example of an IR description based on the Dublin Core in XML format

In this case, the condition of semantic integration of data is formulated based on the matching function of the elements of the description of two information resources D_1 and D_2, which are determined to be essential for their combination into a single integrated resource:

$$Map(MD_i^{DC}(D_1), MD_i^{DC}(D_2))= true, \tag{3}$$

where $i \in (1 - 15)$. Depending on the values of $MD_i^{DC}(D_1)$ and $MD_i^{DC}(D_2)$, the value of the coincidence function can be true or false (Fig. 3a).

The list of metadata elements that are essential in a particular case depends on the nature and tasks of their semantic integration. Two sets of data for which the matching function on essential metadata elements takes the value "true" is considered suitable for semantic integration. In general, it can be one or more elements of the Dublin core. An important problem of this approach is that the concept of coincidence of values, in the general case, means their formal equality, therefore, a clear definition of the coincidence function is a task that cannot always be performed without the participation of a human expert. For this, special tables of correspondence of metadata values are used, which make it possible to conclude the correspondence of the content of individual elements of the Dublin core. For example, the topic of the "Internet" data set is similar in content to the "World Wide Web" topic. The need for expert participation in the formation of criteria for the integration of information resources significantly reduces the universality and scope of the use of metadata as a means of semantic integration.

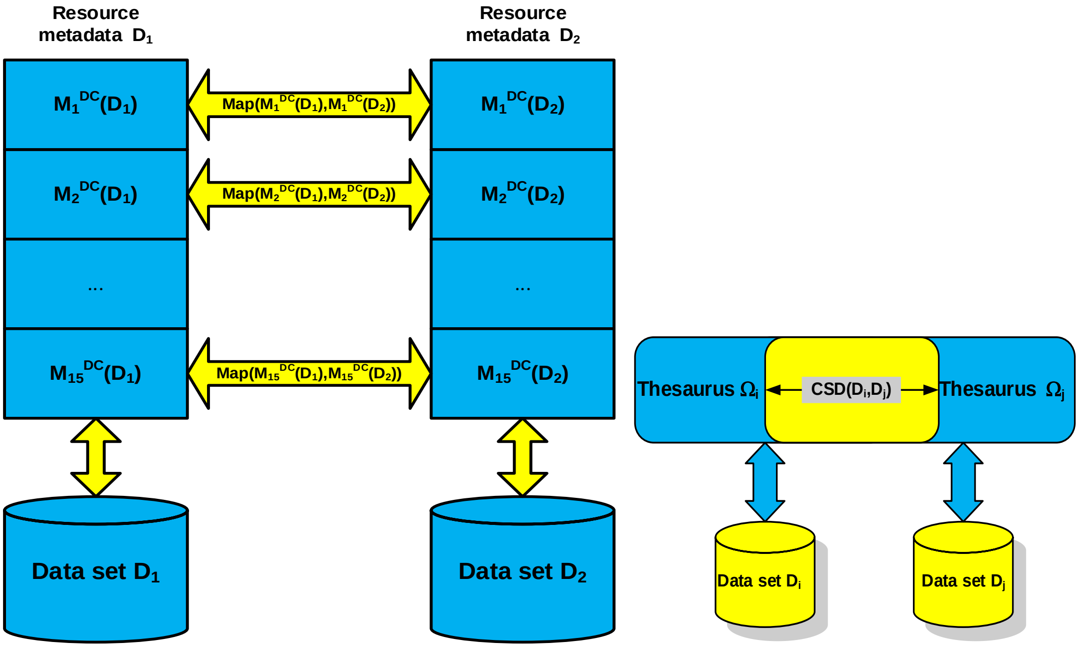


Figure 3: Based on the Dublin core and contextual semantic integration

Semantic integration based on contextual analysis. This method is based on a meaningful comparison of the information content of data sets to be integrated. This approach makes it possible to evaluate the possibilities of integrating both structured (relational) data and loosely structured data presented in arbitrary formats. An essential aspect of determining the semantics of heterogeneous data is their context. Context can take many forms, such as text and hyperlinks on a Web page, the name of the directory in which the data is stored, accompanying annotations and comments on the data, links to physically or logically close data elements, a list of keywords and concepts, etc. In such an application, context helps interpret the content of the data. Loosely structured data is often less accurate than data in traditional databases. The fact that they are obtained from an unstructured resource makes such data semantically heterogeneous or sensitive to the conditions under which they were recorded. Since, in most cases, semantic analysis of the full content of information resources is difficult, and often impossible, in integration processes, it is replaced by contextual analysis of thesaurus terms of data sets. Thesaurus terms are a list of key concepts related to data and are included in a special list - the thesaurus. They are used to describe the semantic correspondence between the lexical units of this data set and specific concepts and meanings from the subject area. The basis for forming a thesaurus is, for example, a list of descriptions of database table columns, XML tags in a document, names of sections and points of a text document, hyperlinks on a Web page, etc.

The condition for the possibility of semantic integration of two data sets in this case is the function of contextual semantic distance between them (CSD function). The value of the CSD function is calculated as follows. Let and Ω_i and Ω_j be the sets of thesaurus terms of the data sets D_i and D_j, respectively, Ω_{ij} be the set of thesaurus terms that are semantically common to the two sets, $|\Omega_i|$, $|\Omega_j|$, $|\Omega_{ij}|$ are powers of the corresponding sets. Then the value of the contextual semantic distance function between data sets is calculated as the share of common values in the smaller of the sets of thesaurus terms of two data sets:

$$\mathrm{CSD}(D_i,D_j) = \frac{|\Omega_{ij}|}{\mathrm{Min}(|\Omega_i|,|\Omega_j|)}. \tag{4}$$

It is considered that the semantic integration of two data sets is possible if the value of the contextual semantic distance function satisfies the condition $CSD(D_i, D_j) \geq 0{,}8$. Fig. 3b presents a general diagram of the process of semantic integration of two sets of disparate data using thesaurus terms and the semantic distance function.

Compared to the order of semantic integration based on metadata, contextual analysis allows to implementation of the verification of the conditions of semantic integration at the formal level and does not require the direct participation of an expert. The most difficult element of this method is the formation of a thesaurus for a data set that contains a certain information resource. Due to the heterogeneity of data formats and structures to be integrated, creating sets of key terms can be quite time-consuming and inefficient. In addition, the formation of the thesaurus largely depends on the subjective human factor, which significantly affects the universality of the method and its independence from specific conditions.

3.2. Application of ontologies for semantic integration of data

3.2.1. Ontologies as a means of describing data semantics

The most promising approach to the integration of data semantics today is the integration based on ontologies. This method of integration involves the use of the main provisions of the two previous approaches - integration using thesauri and metadata, but is much more general than them and takes into account more aspects of data semantics. The use of ontologies as a means of semantic integration is substantiated in [4-5]. In general, an ontology is considered a complete formalized specification of some subject area, which aims to provide the same interpretation of knowledge about this subject area at the human and computer levels. In the case of data integration, the object of the description presented in the form of an ontology is a certain information resource. Therefore, it is appropriate to talk about a specific category of ontologies - data ontologies. In the general case, the formal representation of an ontology is a triple of the form $O = <X, R, F>$, where X is a finite set of concepts (classes, concepts) SA with their properties (attributes), R is a finite set of relations (connections, correspondences) between concepts, F is a finite set of interpretation functions (constraints, axioms). According to the requirements of the IDEF5 standard, concepts are divided into classes and class values. At the same time, classes can form a hierarchy, that is, the value of a class can be another class (subclass), for example, the class "*documents*" can include subclasses "*text documents*", "*XML documents*", "*PDF documents*", etc. as values. Connections between concepts are divided into classification ones - between classes and subclasses, and structural ones, which describe the interaction of classes. An example of structural connections is correspondence between the sections of this work, which in combination form a complete information resource. Fig. 4 shows an example of IR classification from the point of view of integration. The "*Information resource*" class contains the "*Text*", "*Web resource*" and "*Database*" subclasses, which, in turn, are divided into smaller subclasses. The number of levels of hierarchical classification depends on specific requirements and features of data integration processes.

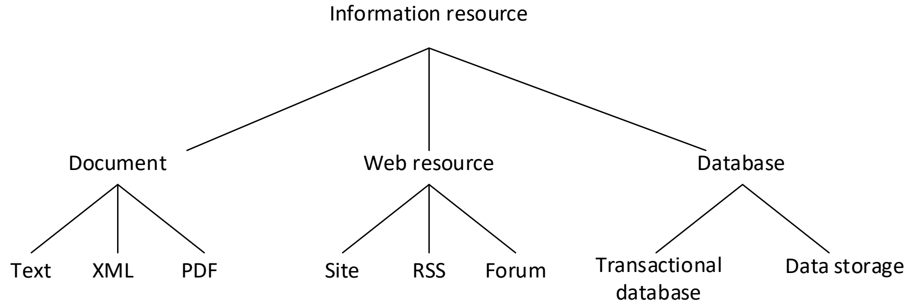


Figure 4: An example of defining classes of the ontology of an IR

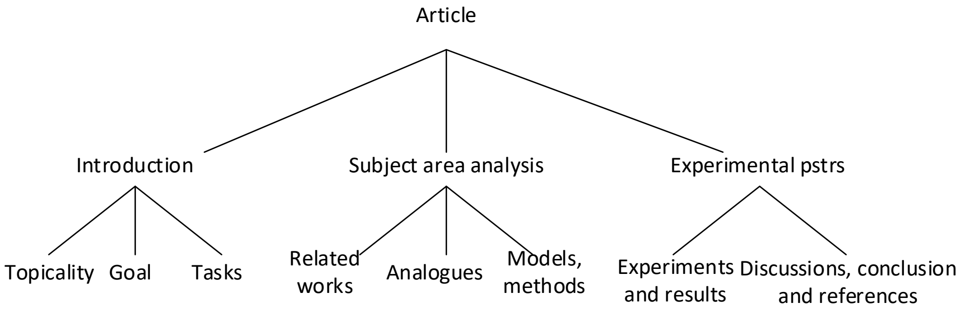


Figure 5: An example of defining an ontology describing the composition of IR

Fig. 5 shows an example of an ontology that describes the composition of a scientific article in the form of concepts that describe its content elements and the connections between them. The processes of semantic data integration involve the construction for each input data set D_i of its own ontology $O(D_i)$, which forms an unambiguous description of the semantics of both the entire information resource and its elements $O(D_i) = <X(D_i), R(D_i), F(D_i)>$, where $X(D_i)$ is a set of concepts that describe data units, their content, properties and belonging to a certain class or category; $R(D_i)$ is a set of connections and relationships between data units that determine the order of their interaction and mutual application; $F(D_i)$ is a set of semantic restrictions and data interpretation functions that connect them with real concepts and objects of the subject area, and also regulate the order of determining such correspondences. Such an ontology describes the semantic relationship of defined and specified data elements with the subject area concepts, forming a coherent "data-content" structure. Since the object of ontology description in the case of semantic integration is data, it can be classified as an applied ontology and presented as a metadata system of a special kind. In this way, the task of semantic integration of data can be reduced to the construction of some global integrated information resource by combining a set of input local ontologies, while identifying correspondences between them, as well as eliminating contradictions and conflicts between ontologies. For this, the process of semantic integration of data is built based on the fulfilment of several predefined conditions, which aim to ensure the correct, from the point of view of content, combination of local input data in the global output resource. The conditions for the possibility of semantic integration of input local data sets, in this case, are formulated as a sequence of requirements for the coordinated joint application of elements of various data ontologies. Input data sets D_i and D_j are considered semantically integrated, suitable for joint use in the formation of a global integrated resource, if the two ontologies $O(D_i)$ and $O(D_j)$ corresponding to these data sets fulfil the rules:

- in sets of concepts $X(D_i)$ and $X(D_j)$

(1) there are no identical concepts described in different ways;
(2) there are no concepts of different content described in the same way;

- in the sets of connections $R(D_i)$ and $R(D_j)$

(1) there are no connections of different content and direction between the same concepts;
(2) there are no connections of the same type that cannot be implemented simultaneously;

- in sets of interpretation functions $F(D_i)$ and $F(D_j)$

(1) there are no functions, the simultaneous implementation of which will lead to ambiguous interpretations;
(2) same-type concepts of different ontologies are not associated with restrictions that cannot be fulfilled simultaneously.

Verification of the specified conditions of semantic integration of data can be implemented both at the formal and the expert level, while the result should be the same.

Fulfilment of the entire set of requirements allows us to conclude the possibility of combining two sets of data at the level of their content with obtaining a semantically correct result. The key property of ontologies to create an unambiguous presentation of the content of data both at the human level and at the level of information technologies constitutes a significant advantage of semantic integration based on ontologies over other approaches, namely:

(1) construction of a meaningfully complete agreed description of the semantics of input local and output integrated information resources;

(2) the possibility of applying unified means and procedures independent of SA data;

(3) possibilities of implementation with the help of appropriate software tools;

(4) semantic integration conditions definition and analysis at the formal level;

(5) obtaining a semantically correct result without the direct participation of a human expert.

An analysis of the features of creating a single content-integrated information resource with the use of metadata, contextual analysis and ontologies allows us to draw the following conclusions.

1. The main methods of solving data semantics integration problems are integration using contextual analysis, metadata and ontologies.
2. Ontologies are the most effective means of semantic integration, as they provide the same interpretation possibility at the human and machine level.
3. The approach to semantic data integration based on ontologies involves the use of intelligent methods and tools to solve the problems of semantic data integration.
4. Presentation of ontologies in the form of metadata corresponds to the main principles of open IS - interoperability, uniformity, mobility, etc.

The semantics integration solutions developed in this section serve as a basis for creating appropriate means of data integration in open information systems. The task of forming and using ontologies as a means of describing the semantics of data in the processes of integration of heterogeneous information resources has several solutions, each of which has its principles, models, and tools.

3.2.2. Building a global ontology of integrated data

The result of the semantic integration of data according to the developed approach is the creation of a global ontology that describes the semantics of the integrated source data set. The problem has several solutions.

Data integration is based on a single ontology. In this case, for the explicit specification of the semantics of different data sets, they form a single global ontology with shared, agreed resources distributed by distributed access. A single ontology of integrated data is built in one of the ways (Fig. 6a).

1. By distribution - at the same time, they form a global description of concepts, relations and interpretation functions with distributed dictionaries, which are used to specify the semantics of each of the data sets to be integrated.

2. By unification – this method provides for the formation and replenishment of the global ontology as a result of the agreed unification of vocabulary resources of local ontologies formed for data sets to be integrated.

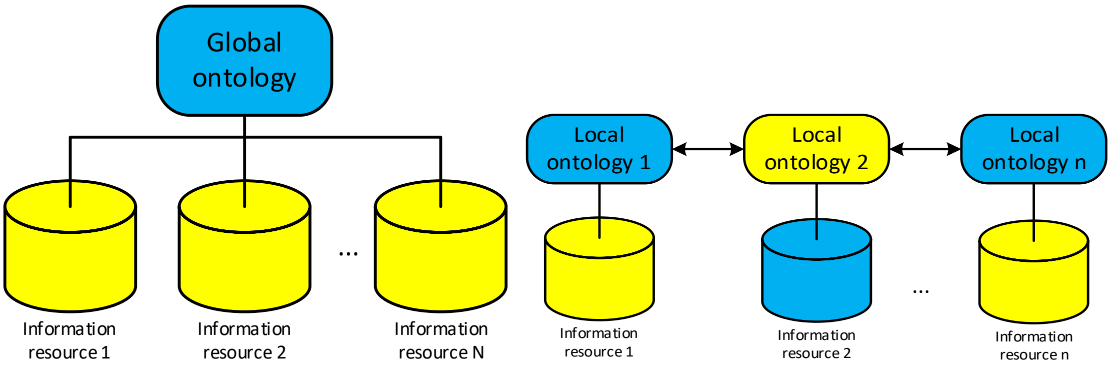


Figure 6: Based on a single ontology and based on a set of ontologies

In the first case, the global ontology of the integrated information resource is presented as a coordinated set of projections of local ontologies of the form O^I=<$\pi_1[O(D_1)]$, $\pi_2[O(D_2)]$, ..., $\pi_N[O(D_N)]$>, in the second as the union of projections of local ontologies:

$$O^I=\pi_1[O(D_1)]\cup \pi_2[O(D_2)] \cup ...\cup \pi_N[O(D_N)], \quad (5)$$

where O^I is the global ontology of an integrated resource; $O(D_i)$ is the ontology of the local resource (i=1, 2,..., N), N is the number of input information resources; $\pi_i[O(D_i)]$ s the projection of the local ontology $O(D_i)$ onto the subject area of the original integrated information resource, which is formed as a subset of the local ontology, which includes only defined concepts, connections and axioms. A feature of semantic data integration based on a single ontology is the possibility of sharing global resources to describe the semantics of each input data set.

Data integration is based on multiple ontologies. In this case, each input data set for semantic integration is described by its ontology, which is not related to others and which operates on its unallocated resources. The process of semantic integration in this case is based on agreement, interaction and resource exchange between independent local ontologies. At the same time, the use of such methods and means of building and processing ontologies is envisaged, which ensures their joint application in the formation of a single semantic space of integrated data. In particular, for the coordinated use of such resources, a check of compliance with the requirements of semantic integration of data, which is described above, is performed. The global ontology of the integrated source data set is formed as a combination of the ontologies of the input local data sets and their matching rules:

$$O^I=< O(D_1), O(D_2), ...,O(D_N), Match(O(D_1), O(D_2), ...,O(D_N))>, \quad (6)$$

where O^I is the global ontology of an integrated resource; $O(D_i)$ is local resource ontology (i=1,2,..., N), N is the number of input SH; $Match(O(D_1), O(D_2), ..., O(D_N))$ is a system of rules for matching ontologies of input local resources, formed according to the conditions of semantic integration.

A hybrid approach to data integration based on ontologies. This method of semantic integration combines the features of the two previously described methods. By analogy with a single ontology, in this case, a common, coordinated distributed resource is created. However, the use of this resource for the specification of the semantics of input data sets occurs through their local ontologies (Fig. 7). The global ontology of the integrated source data set is formed as a combination of the ontologies of the input local data sets and their matching rules

$$O^I=< O^D, O(D_1), O(D_2), ...,O(D_N), Match(O^D, O(D_1), O(D_2), ...,O(D_N))>, \quad (7)$$

where O^I is the global ontology of an integrated resource; O^D is the general distributed ontology of an integrated resource; $O(D_i)$ is the ontology of the local resource (i=1,2,..., N), N is the number of input information resources; $Match(O^D, O(D_1), O(D_2), ..., O(D_N))>$ is a system of rules for mutual agreement of the ontologies of input local resources and the general distributed ontology of the integrated resource, formed according to the conditions of semantic integration.

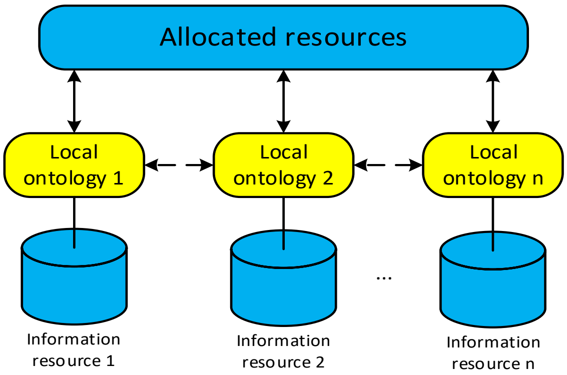


Figure 7: Scheme of hybrid semantic data integration

Similarly, to the previous case, the semantics of the information resource to be integrated is described by a separate ontology. But for the compatibility of local ontologies, global distributed semantic resources are created, in which the basic terms and concepts common to the SA of integrated data are concentrated. Therefore, in this case, matching, according to a defined set of semantic integration requirements, is performed not only between local ontologies but also with distributed semantics description resources. These three approaches, in general, define typical variants of semantic data integration using ontologies.

4. Experiments, results and discussions

4.1. Language tools of the data integration service

The DISP (Data Integration Service Protocol) data integration service protocol is based on a set of integrated data description language tools (Integrated Data Framework - IDF).

IDF is a specialized language for creating metadata that describes the properties of a data set and the order of its application in integration processes. The syntax of IDF is based on the XML syntax, which provides the possibility of processing language constructions by standard means. IDF lexical units are formed in the form of tags of a unified format, the structure and presentation of which are regulated by predefined norms. Each lexical unit has its interpretation, which gives it the meaningful value of the characteristics of one of the properties of the data set that appear in the integration processes. A comprehensive description of the properties of a data set forms a functionally complete, consistent set of its metadata - a meta schema. A data set meta schema is a document built-in XML format, made using IDF lexical constructions, which provides opportunities for processing the properties of a data set in solving data integration problems. Linguistic means of describing integrated data are divided into the following parts (Fig. 8) as tools for describing:

- the syntax of integrated data – *IDF-Syntax*;
- the structure of integrated data – *IDF-Structure*;
- the semantics of integrated data – *IDF-Semantics*;
- additional properties of integrated data – *IDF-Properties*.

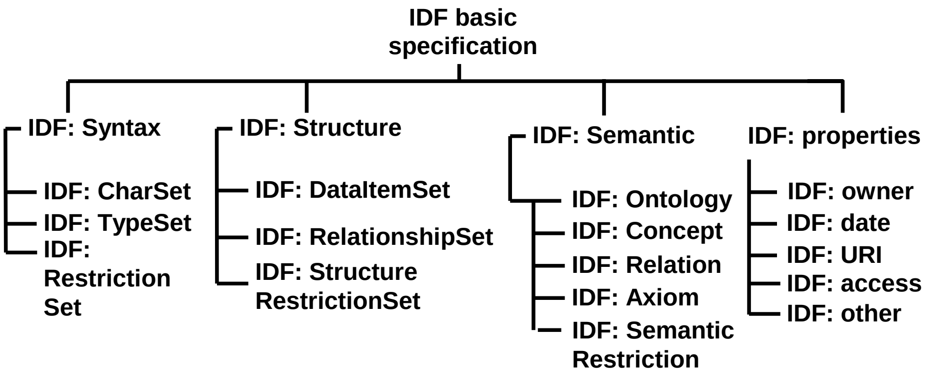


Figure 8: Structure of the Integrated Data Description Language (IDF)

Each subset of the IDF language is a means of constructing a description of the corresponding part of the meta schema.

Linguistic means of describing data syntax. The initial element is the syntax section description instruction. The purpose of the instruction is to build a unified description of how to display data in a set, how to type data and additional requirements for how to present data. The instruction submission format is an XML tag, which can be used as part of a single set of metadata, or in a separate XML document defined to describe the syntax. The syntax description section is part of a general set of metadata that describes the metadata schema of a defined set of data in the processes of their integration. The composition and correspondence of the set of elements used to describe the structure are shown in Fig. 9. Elements of the description in the figure are marked with rectangles, attributes of elements with text inscriptions above arcs, values of elements and attributes with ovals, correspondence between elements, attributes and values with lines.

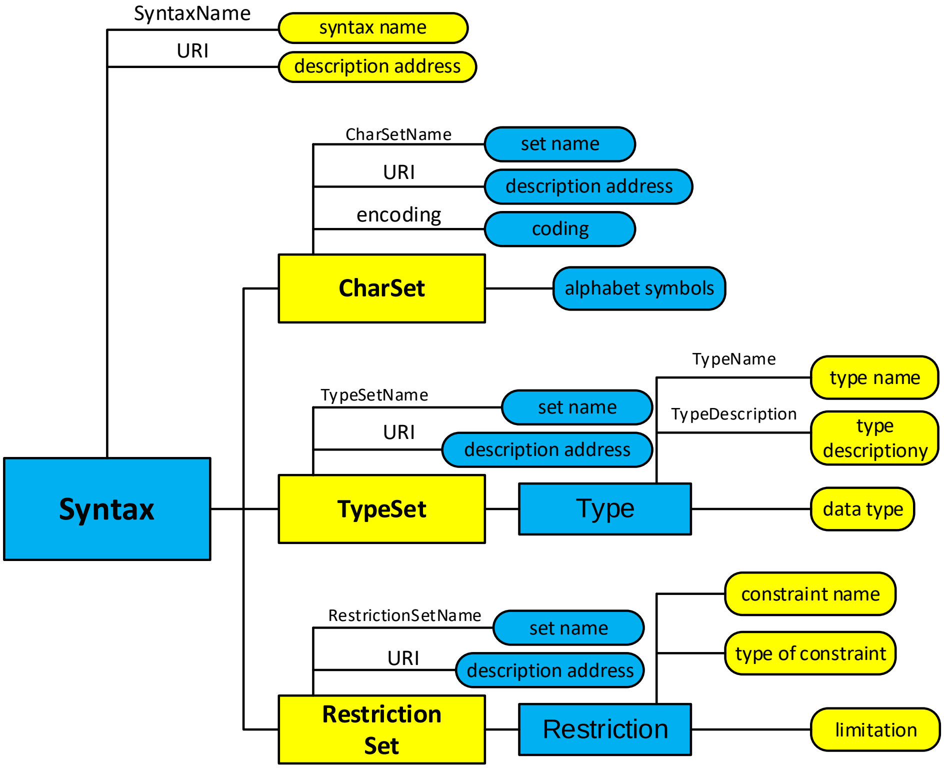


Figure 9: Syntax description structure of integrated data

The general form of the instruction has two forms of presentation - abbreviated and expanded. The abbreviated form of the data syntax definition instruction is used to refer to a previously constructed and saved description. It has the following format:

```
<Syntax SyntaxName="syntax name" URI="address of stored syntax description"/>
```

The expanded form is used to construct an explicit description of the syntax in the general set of metadata, which is a description of the meta schema of the defined set. Its format:

```
<Syntax Syntaxname=" syntax name " > syntax description <Syntax>
```

- syntax name – a symbolic constant, a unique identifier of the syntax in the data integration environment;
- syntax description – specification of the alphabet, specification of types, specification of syntactic constraints;
- the address of the saved syntax description – URI (Unified Resource Locator) specification of the location of the data set in which the previously formed syntax description is stored, executed according to general rules in the XML document format.

The purpose of the alphabet specification instruction is to describe the set of characters that are used to represent the data values of the specified set. It is used as a component of the data syntax description. Alphabet specification instruction format:

```
<CharSet CharSetName=" alphabet name " Encoding=" coding table " URI=" address of stored alphabet " />|
<CharSet CharSetName=" alphabet name " Encoding=" coding table " > list of symbols < CharSet >
```

- the name of the alphabet is a unique identifier of a set of symbols in the environment of the information system;
- encoding table – a reference to the standard table of codes used for transmitting and storing alphabet symbols;
- the address of the saved alphabet – the access path to the XML document file in which the character set is saved;
- list of characters – a character strip without delimiters, which contains all characters allowed for a given set of data.

The data type specification statement defines the set of data types that are used to type the data in the specified set. The instruction describes the general data syntax description section and has the following format:

```
<TypeSet TypeSetName=" the name of a set of types
" URI=" the address of the saved description of the set of types " /> |
<TypeSet TypeSetName=" the name of a set of types " > list of type specifications </TypeSet>
```

- the name of the set of types – the unique identifier of the set of types in the information system environment;
- the address of the saved description of the set of types - the access path to the XML document file in which the previously prepared set of data types is saved;
- type specification – a description that contains the name of the type, the class to which it belongs and the procedure for defining the data type.

A data type specification is part of the specification of a set of data types used to type the values of a defined set. Each type specification describes the content and procedure for defining a single data type. A type specification statement is represented as an XML element of a type^

```
<Type TypeName=" type name " TypeClass=" type class "> type definition </Type>
```

- type name – a unique identifier of the type in the information system environment;
- type class – can take one of the preset values: *Basic* – basic elementary type; *Generic* – generated type, formed based on the basic one according to the standard procedure; *User Defined* – a generated type formed from a base in a way defined by the user.
- definition – a reference to the base type and the method of formation for generated types.

The syntax constraint description statement is part of the syntax description of the data set. With its help, additional requirements regarding the form of presentation of the values of individual data elements in the corresponding set are determined. The syntactic constraint description instruction has the following XML tag format:

```
<RestrictionSet RestrictionSetName=" the name of the constraint set " URI=" the address of the stored description
of the constraint set " /> |
    <RestrictionSet RestrictionSetName=" the name of the constraint set " >
list of constraint specifications </RestrictionSet>
```

- the name of the set of restrictions – the unique identifier of the set of restrictions in the information system environment;
- the address of the saved description of the set of restrictions – the access path to the XML document file in which the pre-prepared set of restrictions is saved.

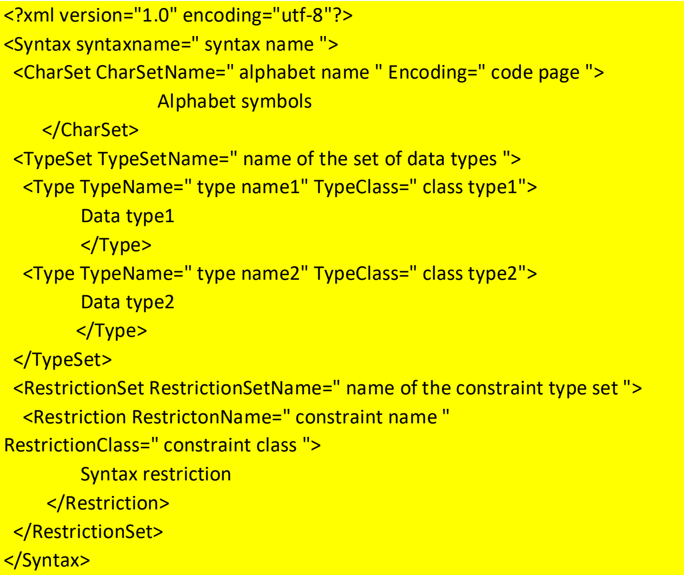
The syntactic constraint specification is a formalized description of the form and content of data value presentation requirements. Each syntax constraint can be applied multiple times for different values. A syntactic constraint specification is provided as an XML element of the type:

```
<Restriction RestrictionName=" the name of the constraint " RestrictionClass=" constraint class "> limit definition
</Restriction>
```

- restriction name – a unique identifier of the restriction in the information system environment;
- restriction class – one of the values: *Format* – basic standardized data presentation format; *Mask* – mask for entering data values; *Template* – a template for converting data values.
- constraint definition – a character string describing a pattern or mask or a reference to a standardized basic format. An example of a description of the syntax of integrated data in the full format is an XML document of the form shown in Fig. 10.

Linguistic means of describing data structure. The purpose of this part of the IDF language is to form a description of the order of determination of data units in a set, their combination, arrangement and formation of complex units based on simple ones. The section describing the structure is part of the general set of metadata describing the meta-schema of the defined set of data in the processes of their integration. The composition and correspondence of the set of elements used to describe the structure are shown in Fig. 11. Elements of the description in the figure are marked with rectangles, attributes of elements with text inscriptions above arcs, values of elements and attributes with ovals, correspondence between elements, attributes and values with lines. The initial element of this subset of the language is an instruction to define a structure description section. It is presented in XML-sentence format:

```
<Structure StructureName=" structure name " StructureType=" structure type " URI=" the address of the saved
structure description "/>
```



```
<?xml version="1.0" encoding="utf-8"?>
<Syntax syntaxname=" syntax name ">
 <CharSet CharSetName=" alphabet name " Encoding=" code page ">
                Alphabet symbols
     </CharSet>
 <TypeSet TypeSetName=" name of the set of data types ">
  <Type TypeName=" type name1" TypeClass=" class type1">
          Data type1
          </Type>
  <Type TypeName=" type name2" TypeClass=" class type2">
          Data type2
         </Type>
 </TypeSet>
 <RestrictionSet RestrictionSetName=" name of the constraint type set ">
  <Restriction RestrictonName=" constraint name "
RestrictionClass=" constraint class ">
          Syntax restriction
      </Restriction>
 </RestrictionSet>
</Syntax>
```

Figure 10: A listing of the XML document with a description of the syntax of the data set

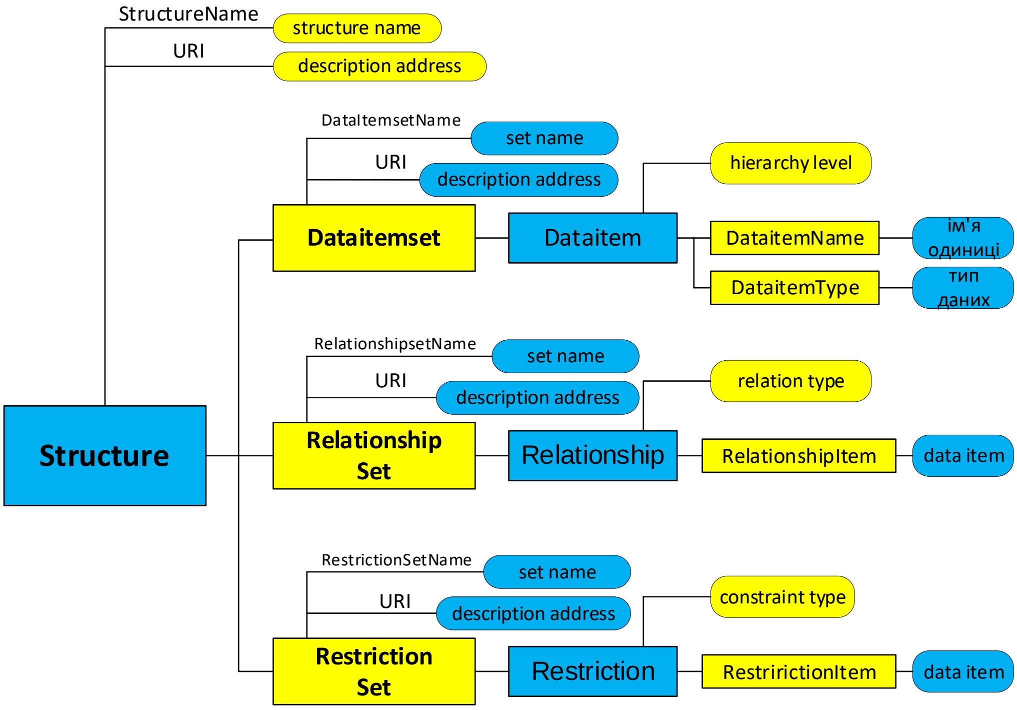


Figure 11: Description of the structure of integrated data

to create an abbreviated description of the structure of the data set, in the case of using a previously built saved set of structure definitions, or in the form:

<Structure StructureName=" structure name " StructureType=" structure type " > description of the structure </Structure>

- structure name – symbolic constant, unique identifier of the data set structure in the integration environment;
- type of structure – a constant that has an auxiliary purpose and informs the data consumer about the way they are structured; it is possible to use such values, the list of which can be expanded by references to standardized formats:
    a. *relational* – for structured relational data;
    b. *XML* – for data provided in XML format;
    c. *text* – for text data;
    d. *HTML* – for data presented in the format of web pages;
    e. *OOXML* – for Microsoft Office standard data;
    f. *semistructured* – for other loosely structured data;
    g. *binary* – for data provided in binary files;
    h. no value indicates that the structuring method is not set;
- the address of the saved description of the structure – URI (Unified Resource Locator) specification of the location of the data set in which the previously formed description of the structure is stored, executed according to general rules in the format of an XML document;
- description of the structure – specification of data units, specification of relationships between data units, specification of structural constraints.

Each of the components of the structure description is specified in the format of an XML element of a specified format. To define a set of data units in an abbreviated format, XML sentences of the form are used:

```
<DataItemSet DataItemSetName=" the name of the dataset " URI=" the address of the saved description of the set of data units "/>,
```

or in full XML-sentence format:

```
<DataItemSet DataItemSetName=" the name of the dataset " >
list of data unit specifications </DataItemSet>,
```

The specification of a data unit is defined by an XML format element:

```
<DataItem DataItemLevel=" hierarchy level "> data unit description </DataItem>,
```

where the *hierarchy level* is a numerical value that indicates the order in which some units are part of others in the specifications of complex data units, the description of the data unit is specified by the following XML elements:

```
<DataItemName> the name of the data unit </DataItemName>
<DataItemType> type </DataItemType>
```

data unit name is the unique identifier of the data unit in the set, *type* is one of the predefined types used to type the data in the set. If the data element is complex, the <DataItem> XML elements are nested according to the hierarchy of occurrence. XML-sentences of the form are used to define a set of relationships between data units in a shortened format:

```
<RelationshipSet RelationshipSetName=" the name of the connection set "
URI=" the address of the stored connection set description "/>,
```

in full format:

```
<RelationshipSet RelationshipSetName=" the name of the connection set "> list of connection specifications
</RelationshipSet>.
```

The communication specification is specified by an XML element of the following form:

<Relationship RelationshipType="тип зв’язку"> list of data items </Relationship>,

connection type – set by the appropriate symbolic constant, the following types of connections are possible: *binary*-1:1 – binary connection of the "one-to-one" type; *binary*-1:* – one-to-many binary communication; *binary*-*:1 – binary connection of the "many-to-one" type; *binary*-*:* – binary connection of the "many-to-many" type; *N-ary* is a relation of arbitrary order.

The list of elements specifies a list of data units between which a relationship is defined. Each element is defined by an XML view tag:

<RelationshipItem> the name of the data unit </RelationshipItem>.

The number of list items is determined by the type of connection. The direction in binary links is determined by the order of the elements in the list.

XML sentences are used to define a set of additional requirements for the data structure (constraints) in a shortened format:

<RestrictionSet RestrictionSetName=" the name of the connection set "
URI=" the address of the stored constraint set description "/>,

in full format - a sentence of the form:

<RestrictionSet RestrictionSetName=" the name of the connection set ">
a list of structural constraint specifications </StructureRestrictions>.

A structural constraint specification defines one of the types of requirements for the combination of data elements in structural units and is specified by an XML tag of the form:

<Restriction RestrictionType=" constraint type "> list of data units </Restriction>,

restriction type – a symbolic constant that defines one of the variants of data structure requirements: *inclusive* – the listed data units are part of one another only in combination; *exclusive* – the listed data units cannot be part of another at the same time.

The data units in the constraint are specified by the XML view tag:

<RestrictionItem> the name of the data unit <RestrictionItem>,

the name of the data unit is set according to the definitions given in the section on the description of structural data units. The part of the structure description defining the structural constraints is optional. Attribute values in XML tags that specify names and addresses have the same usage as names and addresses in tags defined above. A sample description of the data set structure in XML document format looks like this (Fig. 12).

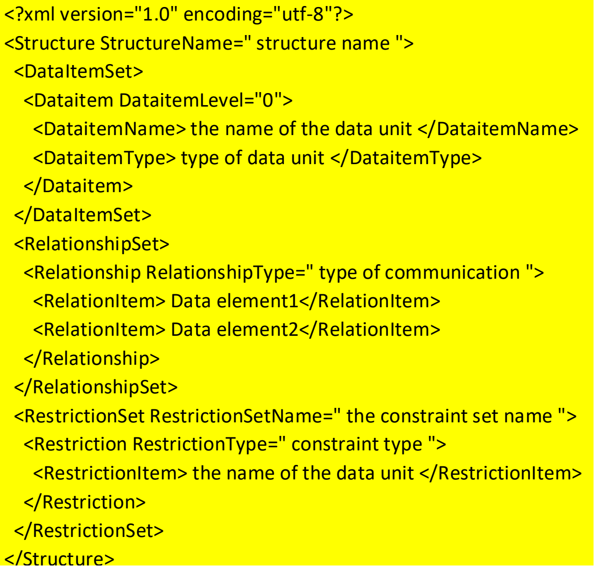
Linguistic means of describing the semantics of integrated data. The purpose of this part of the IDF language is to form a description of the order of interpretation of the data in the set. The semantics description section is part of a general set of metadata describing the meta-schema of a defined set of data in the processes of their integration. The semantics description section is defined by an XML sentence of the following form:

<Semantics SemanticsName=" the name of the semantics description " URI=" the address of the stored semantics description "/>

to create a description of the semantics of the data set in abbreviated format, in the case of using a pre-built saved set of structure definitions, or in the form:

<Semantics SemanticsName=" name of the semantic description "> description of semantics</Semantics
</Semantics>;

to create a description of data semantics in an expanded format.



```
<?xml version="1.0" encoding="utf-8"?>
<Structure StructureName=" structure name ">
 <DataItemSet>
  <Dataitem DataitemLevel="0">
   <DataitemName> the name of the data unit </DataitemName>
   <DataitemType> type of data unit </DataitemType>
  </Dataitem>
 </DataItemSet>
 <RelationshipSet>
  <Relationship RelationshipType=" type of communication ">
   <RelationItem> Data element1</RelationItem>
   <RelationItem> Data element2</RelationItem>
  </Relationship>
 </RelationshipSet>
 <RestrictionSet RestrictionSetName=" the constraint set name ">
  <Restriction RestrictionType=" constraint type ">
   <RestrictionItem> the name of the data unit </RestrictionItem>
  </Restriction>
 </RestrictionSet>
</Structure>
```

Figure 12: XML document Listing with a description of the structure of the integrated data

The description of data semantics consists of one or more ontologies. The composition and correspondence of the set of elements used to describe the ontology are shown in Fig. 17.5. The elements of the description are marked in the figure with rectangles, the attributes of the elements are marked with text above the arcs, the values of the elements and attributes are marked with ovals, and the correspondences between elements, attributes and values are with lines. The ontology description section starts with an XML tag like this:

```
<Ontology OntologyName=" ontology name " URI=" the address of the stored ontology description "> ontology description </Ontology>,
```

the name of the ontology is a unique identifier of the ontology, and *the address of the stored ontology description* is a pointer to the XML document in which the preformed description is stored, this attribute can be omitted in the case of building an explicit description.

The description of the ontology consists of sections describing concepts, describing relations, and describing axioms of data interpretation. The concept description section is specified by an XML tag:

```
<ConceptSet ConceptSetName=" the name of the set of concepts " URI=" the address of the saved concept description "> list of concept descriptions </ConceptSet>,
```

the name of the set of concepts is a unique identifier of the set of concepts, and *the address of the saved description of the set of concepts* is a pointer to the XML document in which the performed description is stored, this attribute can be omitted in the case of creating an explicit description (Fig. 13).

The description of the concept is specified by an XML tag:

```
<Concept ConceptLevel= hierarchy level > list of attributes </Concept>,
```

for simple concepts, or

```
<Concept ConceptLevel= hierarchy level > list of concepts </Concept>,
```

for complex concepts, at the same time, the value hierarchy level is used - an integer that indicates the nesting level of concepts, in the case of defining them according to a hierarchical scheme, the highest level of the hierarchy, as well as the level of the hierarchy for simple concepts, is marked with the value 0.

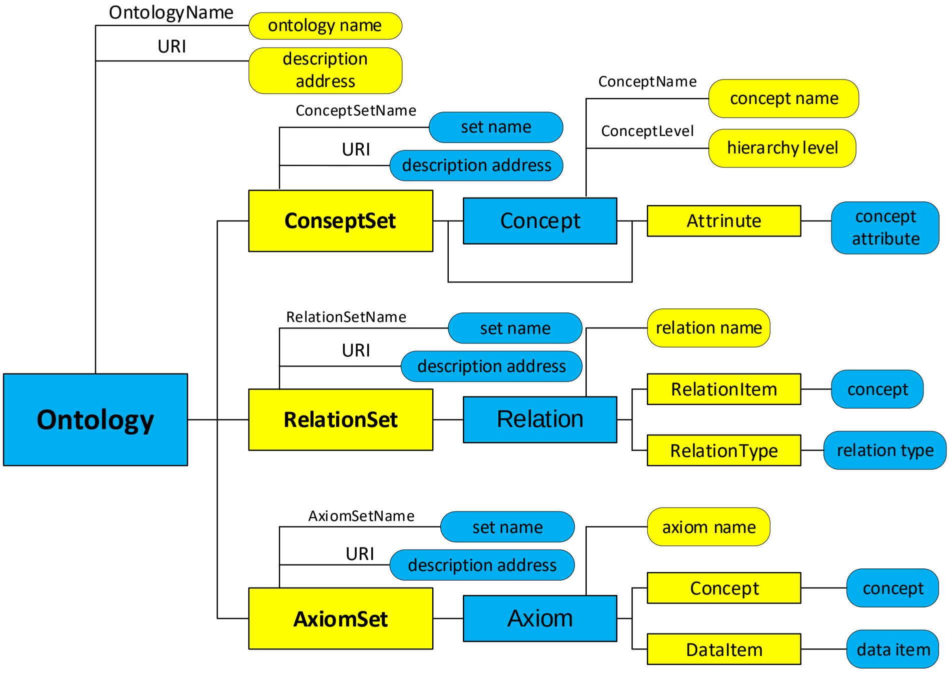


Figure 13: The composition of means of description of the ontology of integrated data

The concept attribute is defined by an XML tag:

<Attribute> attribute name </Attribute>,

the name of the attribute is a character string that is interpreted as a unique designation of the attribute that describes one property of the concept.

Section of relations, intended for the formation of a description of interaction and correspondence between concepts. The section is defined by an XML sentence:

<RelationSet RelationSetName=" the name of the relation set " URI=" адреса збереженого опису відношень "> relationship descriptions list </RelationSet>,

the name of the set of relations is the unique identifier of the set of relations in the data integration environment, and *the address of the saved description of the set of relations* is a pointer to the XML document in which the performed description is saved, this attribute can be omitted in the case of creating an explicit description.

The relationship description is specified by the XML tag of the view:

<Relation RelationName=" relation name " > relation specification </Relation>,

relation name – the unique identifier of the relation in the data integration environment. The relation specification specification expression has the form:

<RelationItem> concept </RelationItem><RelationType> relationship type</RelationType> <RelationItem> concept </RelationItem>

for relations between two concepts, or

<RelationItem> concept </RelationItem>...<RelationItem> concept </RelationItem>

<RelationType> relationship typ</RelationType> <RelationItem> concept </RelationItem>...<RelationItem> concept </RelationItem>

for the description of objects between sets of concepts, *the relation type* is a constant that characterizes the content of the relation and the nature of the interaction of the concepts between which it is defined according to the terminology of the IDEF5 standard: *sub-kind-of*, *instance-of* – classification relation; *part-of* – aggregation relation; *produce* – production relationship; *associate* – association relation. The absence of the <RelationType> tag is considered equivalent to defining an association relation. Axioms that determine the interpretation of data units due to their connection with concepts are specified by an XML tag of the following form:

<Axiom AxiomName=" the name of the axiom " > axiom specification</Axiom>,

axiom name is the unique identifier of the axiom in the data integration environment. The axiom specification has the format of an XML sentence of the following composition:

<DataItem> unit of data </DataItem>…<DataItem> unit of data </DataItem><Concept> concept </Concept>,

the first part of which is a list of data units, the second is the concept that they represent in the information resource. A sample XML document that contains a description of semantics looks like this (Fig. 14).

4.2. Language means of integrated data properties describing

The category of general data set information in integration processes includes data set identifier; data creation/updation date; the address of the location in the integration environment; information about the owner; information about the rights and restrictions of access to the data set; and additional information.

```
<?xml version="1.0" encoding="utf-8"?>
<Semantics SemanticsName=" name of the semantic description " URI=" the address of the saved description ">
 <Ontology OntologyName="ontology name" URI="stored description address">
  <ConceptSet ConceptSetName=" the name of the set of concepts "
URI=" the address of the saved description ">
   <Concept ConceptName=" concept name " ConceptLevel=" concept hierarchy level ">
    <Attribute> concept attribute </Attribute>
   </Concept>
  </ConceptSet>
  <RelationSet RelationSetName=" the name of the relation set "
URI=" the address of the saved description">
   <Relation RelationName=" relation name ">
    <RelationItem> concept </RelationItem>
    <RelationType> relation type </RelationType>
   </Relation>
  </RelationSet>
  <AxiomSet AxiomSetName=" the name of the axiom set " URI=" the address of the saved description ">
   <Axiom AxiomName=" the name of the axiom ">
    <Concept> concept </Concept>
    <DataItem> data item </DataItem>
   </Axiom>
  </AxiomSet>
 </Ontology>
</Semantics>
```

Figure 14: Sample description of dataset semantics in XML format

The description section of the general information about the data set is created in an XML sentence of the form:

<Properties PropertiesSetName=" property description name "> description of properties </Properties>,

where *property description* is a set of XML elements that describe the values of the properties defined above. In the abbreviated format, the description section is specified by an XML sentence of the form:

<Properties PropertiesSetName="property description name" URI="description address">,

where *the address of the saved description* is the address of the XML file in which the pre-prepared description of general information about the data set is saved. The description of each of the properties is specified by the corresponding XML element in the general document, which contains the description of the properties. Dataset identifier –

<DataSetName> dataset name </DataSetName>,

data set name – a unique identifier of the data set, formed according to the rules of the integration environment.

The data set presentation format determines the methods of its formation and the means of maintaining and accessing the data. Definitions of the format are given by a tag of the following form:

<DataSetFormat>data set presentation format<DataSetFormat>,

the data set presentation format is a constant that indicates the means or technology by which the data set is formed and access to it is supported. For this, generally accepted notations and names are used, in particular the following: XML – for data submitted in the general XML document format, HTML x.x – data in the format of a web page, where x.x indicates the language version, OOXML – Microsoft Office document format, DBF – database file format of type xBase, Oracle x.x - database table of Oracle DBMS version x.x, etc. In general, a set of ways to indicate the presentation format is defined within a specific implementation by creating appropriate glossaries. Data set creation/update date –

<DataSetDate> date created/updated </DataSetDate>,

date created/updated – a date/time value that indicates when the data in the set was last updated. The description of the data set owner is

<DatasetOwner>owner</DatasetOwner>,

the owner is the identifier of the user who owns the data set in the information system environment. The description of user access rights to the data set is specified by a composite XML view element:

```
<DataSetAccess> <User>
<UserName> User ID </UserName>
<UserRight> type of action </UserRight>...<UserRight> type of action </UserRight>
</User>
</DataSetAccess>,
```

each <User> type element describes the rights granted to the user with the identifier specified by the <UserName> type element to perform one of several types of actions on the data set specified by the <UserRight> type element. The value of the type of action is specified by a symbolic literal, which can take the following values: *access* - the right to receive data values from the set, *update* - the right to update the data values in the set, *reconstruct* - the right to change the description and properties of the data set, *all rights* -

full rights to perform any actions on the data set, including destruction. Additional information about the data set is specified by the XML view element:

```
<Other> a list of additional data set properties </Other>,
```

a list of additional properties of the data set - symbolic literal or XML elements of arbitrary content that contain comments and explanations about the properties of the data set that are not included in the list defined above.

A sample description of general information about a data set in the format of an XML document is shown in Fig. 15.

A complete description of the data set as a separate unit of the integration process combines the description of all predefined components - syntax, structure, semantics and general properties and forms its meta schema (Fig. 16). The meta schema of each separate set is a part of the general metadata that ensures the maintenance and IRP of the system.

```
<?xml version="1.0" encoding="utf-8"?>
<Properties PropertiesSetName=" property description name " URI=" the address of the saved description ">
 <DataSetName> dataset name </DataSetName>
 <DataSetDate> date created/updated </DataSetDate>
 <DatasetOwner> owner </DatasetOwner>
<DataSetAccess>
  <User>
   <UserName> User ID </UserName>
   <UserRight> the right to access the set </UserRight>
  </User>
 </DataSetAccess>
 <Other> a list of additional data set properties </Other>
</Properties>
```

Figure 15: A sample XML document describing the general characteristics of a data set

The complete description of the data set meta schema in the integration process is specified as an XML file of the following general format:

```
<?xml version="1.0" encoding="utf-8"?>
<MetaScheme MetaSchemeName=" the name of the meta schema " URI=" the address of the saved description of
the meta-schema ">
          <Properties> description of general properties </Properties>
          <Syntax> syntax description </Syntax>
          <Structure> description of the structure </Structure>
          <Semantscs> description of semantic </Semantscs>
</MetaScheme>,
```

which consists of four functional sections containing, respectively, a description of the general properties of the data set, syntax, structure and semantics of the data. The described linguistic tools use provides the possibility of manipulating and transforming the syntax, structure and semantics of data in the processes of their integration. The use of XML as the basis for defining the language means of the data integration service protocol makes it interoperable, and independent of data integration platforms and technologies and enables the elaboration of the data set schema using standardized and unified means. The order of functioning of language means of the integration service is as follows:

- user requests through the web interface are directed to the web server, which supports a set of server Java pages, a functional module is executed that solves one of the tasks of the heterogeneous data integration service;

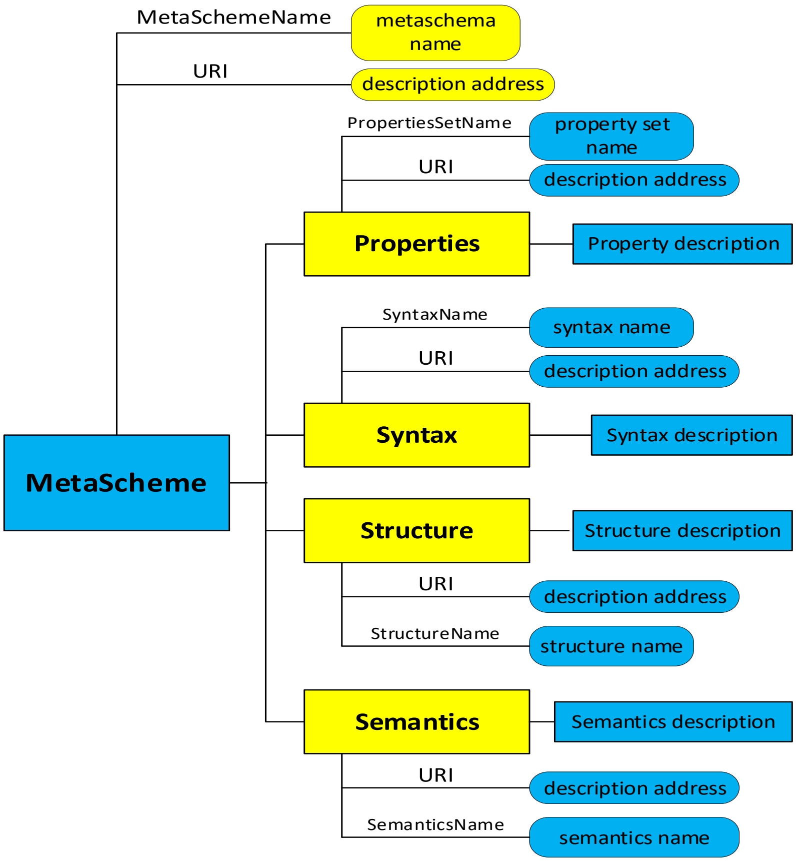


Figure 16: General structure of data set description in integration processes

- the user's request is analyzed and interpreted by server means, as a result of which a list of data to be generated is created;
- based on the list of data units, a request to the metadata repository in XQuery format is built, the result of which is the finding and selection of values that describe the required data units;
- based on this description, a meta-schema of integrated data is created, which forms the result of the user's request;
- the received metaschema is interpreted into a set of data selection requests from repositories of various formats, which are implemented through standardized access interfaces;
- the results of query execution are transferred to the appropriate integration module, which combines them with the requirements of the meta schema and forms the resulting data set;

- the resulting data set is formatted and presented to the user according to the interface requirements.

This project solution made it possible to create a single complex of IT management of the enterprise "112 Ukraine" based on existing IR and systems and thereby avoid the costs of developing new projects "from scratch".

5. Conclusions

DISP is an application-level protocol that defines the data integration service implementing methods, order and means at the open information system user request. The main purpose of using the protocol is the organization of the service for the user of the open information system, which forms the data set he needs by integrating the data located in a set of disparate local information resources. The protocol does not depend on protocols of lower (presentation and session) levels and provides for user interaction with open information system services using standardized interfaces. The goals achieved by the protocol are as follows

- creation of a platform for building a data integration service in environments of open information systems with heterogeneous distributed information resources;
- maintenance of interaction with standard application tools of the user of the open information system;
- perception and interpretation of requests of the user of the open information system to receive the appropriate information product or service;
- ensuring user access to disparate local information resources set through integration;
- formation of a global data set based on local information resources through integration;
- transfer of the results of the request to the user of the open information system.

The goals of the protocol are not:

- determination of ways, methods and means and formation of local information resources;
- determination of the composition and characteristics of technological means of data processing in local repositories;
- administration of repositories in which local information resources are concentrated;
- implementation of data protection and security measures in the IS environment.

Based on the performed analysis and generalizations, the main methods of building information resources of business analytics systems are determined - homogenization, distribution, and integration. The approach based on data integration is the most appropriate to the features of business intelligence systems. Together with the other resources integration of business intelligence systems, data integration forms a single coordinated set of actions for the design, construction and support of business intelligence systems.

Based on the conducted analysis, it is possible to conclude that there are several relevant today's interrelated problems in the e-commerce field.

- The problem of forming a high-quality information resource of business analytics systems that is relevant to the system goals, adequate to its tasks and appropriate to the needs and requirements of users.
- The problem of heterogeneous form, content and properties integration of input information resources into a single agreed common resource of business analytics systems.
- The problem of developing and substantiating a single generalized theoretical concept of data integration for the unified effective methods, tools and modern technologies creation to solve the problem of forming information resources of business analytics systems in various industries and fields of application.

The development of methods of multi-level data integration in business analytics systems allows us to draw the following conclusions:

- It is advisable to base the method on a multi-level data model that takes into account various aspects of this data (syntax, semantics, meaning);
- The key point of the integration process is its implementation based on data meta-schemas, which made it possible to reduce access to the actual data and reduce the time required for integration;
- The general process of integration is carried out as a set of sub-processes of integration of values, syntax and semantics of data;
- The use of ontologies in the processes of semantic data integration made it possible to carry out integration at the level of data content, to achieve the same interpretation of data by both people and machines.

In the process of developing knowledge presentation methods for intelligent business analytics systems, the algebraic theory of types was laid as its basis. An algebraic system has been built in which there is an unambiguous mapping between ontology entities and abstract data types. The development of presentation methods and the architecture of intelligent networks of business processes allows us to draw the following conclusions:

- IS is based on the ontology of business processes.
- Given the changing nature of SA and the incomplete knowledge of the ontology developer, it is necessary to use ontologies that can adapt to changes.
- Intelligent networks of business processes should be presented as a set of interacting executable ontological models.

As a result of the work, a specification of language tools was developed for creating metadata that describes the syntax, semantics, and structure of data in integration processes, as an interoperable service of an intermediate level. The advantages of linguistic means of data integration over instrumental ones are due to the following factors.

- Unification - the use of protocol-level language tools makes it possible to reduce the processes of data integration to the manipulation of typical concepts, objects, properties and procedures for their processing.
- Interoperability – provides the possibility of joint use of means of maintaining the application protocol of data integration with any other means of data processing.
- Mobility – the implementation of protocol means of data integration at the application level makes them independent of the specifics of the platforms and the implementation environment, which ensures the possibility of their free movement.
- Standardization of processing formats and procedures – the use of XML as the basis of the language means of describing objects and processing metadata makes it possible to process the necessary resources by standard means and according to standardized procedures.
- Compliance with the principles of SOA – provides easy access and use of data integration protocol tools for a wide range of information system users.
- Ease of implementation – such means use of integration does not require restructuring of BP, platforms or other means of maintaining the open information system environment, as it involves the formation of an additional, relatively autonomous intermediate layer of data abstraction.
- Insignificant cost – the need absence to rebuild information systems, and develop, purchase and implement complex and expensive data integration tools and platforms makes projects based on protocol tools relatively inexpensive.

References

[1] Y. Burov, V. Vysotska, P. Kravets, Ontological approach to plot analysis and modeling, CEUR Workshop Proceedings 2362 (2019) 22-31.

[2] A.Y. Berko, Models of data integration in open information systems, Actual Problems of Economics 10 (2010) 147-152.

[3] N. Garanina, E. Sidorova, I. Kononenko, S. Gorlatch, Using multiple semantic measures for coreference resolution in ontology population, International Journal of Computing 16(3) (2017) 166-176.

[4] P. Kravets, Y. Burov, V. Lytvyn, V. Vysotska, Gaming method of ontology clusterization, Webology 16(1) (2019) 55-76.

[5] N. Schahovs'ka, Y. Syerov, Web-community ontological representation using intelligent dataspace analyzing agent, in: Proceedings of Experience of Designing and Application of CAD Systems in Microelectronics, CADSM, 2009, pp. 479-480.

[6] V. Lytvyn, The similarity metric of scientific papers summaries on the basis of adaptive ontologies, in: Proceedings of 7th International Conference on Perspective Technologies and Methods in MEMS Design, MEMSTECH, 2011, p. 162.

[7] V. Lytvyn, D. Dosyn, V. Vysotska, A. Hryhorovych, Method of ontology use in OODA, in: Proceedings of the IEEE 3rd International Conference on Data Stream Mining and Processing, DSMP, 2020, pp. 409-413. doi: 10.1109/DSMP47368.2020.9204107.

[8] A.Y. Berko, Methods and models of data integration in E-business systems, Actual Problems of Economics 10 (2008) 17-24.

[9] S. Orekhov, Advanced Method of Synthesis of Semantic Kernel of E-content, CEUR Workshop Proceedings 3312 (2022) 87-97.

[10] Y. Burov, V. Lytvyn, V. Vysotska, I. Shakleina, The Basic Ontology Development Process Automation Based on Text Resources Analysis, in: Proceedings of the IEEE 15th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT, 2020, 1, pp. 280-284. doi: 10.1109/CSIT49958.2020.9321910.

[11] A. Berko, I. Pelekh, L. Chyrun, M. Bublyk, I. Bobyk, Y. Matseliukh, L. Chyrun, Application of ontologies and meta-models for dynamic integration of weakly structured data, in: Proceedings of the 2020 IEEE 3rd International Conference on Data Stream Mining and Processing, DSMP, 2020, pp. 432–437. doi: 10.1109/DSMP47368.2020.9204321.

[12] O. Tyshchenko, V. Tyshchenko, Metadata of the Linguistic Sources in Lexicographic Electronic Tool, CEUR workshop proceedings 2604 (2020) 306-316.

[13] V. Tkachenko, O. Cherednichenko, M. Godlevskyi, The Concept of Device Meta-Model for Real-Time Communication in the Transboundary Environment Monitoring System, in: Proceedings of the International Scientific-Practical Conference, PIC S&T, 2018, pp. 64-70.

[14] M. Schölzel, E. Eren, K. Detken, L. Schwenke, Monitoring android devices by using events and metadata, International Journal of Computing 15(4) (2016) 248-258.

[15] M. Emmerich, A., Giotis, M. Özdemir, T. Bäck, K. Giannakoglou, Metamodel–assisted evolution strategies, Lecture Notes in Computer Science 2439 (2002) 361–370.

[16] The Dublin Core Metadata Element Set. Draft Standard: ANSI/NISO Z39.85-2007. National Information Standard Organization, 2007.

[17] J. Rogushina, Ontological Approach in the Smart Data Paradigm as a Basis for Open Data Semantic Markup, CEUR Workshop Proceedings 3403 (2023) 12-27.

[18] Y. Burov, V. Vysotska, V. Lytvyn, L. Chyrun, Software based on ontological tasks models, Lecture Notes on Data Engineering and Communications Technologies 149 (2023) 608–638.

[19] V. Lytvyn, V. Vysotska, Y. Burov, V. Hryhorovych, Knowledge novelty assessment during the automatic development of ontologies, in: Proceedings of the IEEE 3rd International Conference on Data Stream Mining and Processing, DSMP, 2020, pp. 372-377. doi: 10.1109/DSMP47368.2020.9204124.

[20] V. Lytvyn, V. Vysotska, Y. Burov, O. Brodyak, Approach to Automatic Construction of Interpretation Functions during Ontology Learning, in: Proceedings of the IEEE 15th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT, 2020, 1, pp. 267-271. doi: 10.1109/CSIT49958.2020.9321920.

[21] K. Rukkas, G. Zholtkevych, Probabilistic model for estimation of cap-guarantees for distributed datastore. Advanced Information Systems 4(2) (2020) 47–50. doi: 10.20998/2522-9052.2020.2.09.

[22] V. Shynkarenko, L. Zhuchyi, Semantic Checking of Different Type Information Sources About Permitted Speeds in Railway Transport, CEUR Workshop Proceedings 3171 (2022) 711-723.

[23] N. Sharonova, I. Kyrychenko, I. Gruzdo, G. Tereshchenko, Generalized Semantic Analysis Algorithm of Natural Language Texts for Various Functional Style Types, CEUR Workshop Proceedings 3171 (2022) 16-26.

[24] L. Savytska, M. Turgut Sübay, N. Vnukova, I. Bezugla, V. Pyvovarov, Word2Vec Model Analysis for Semantic and Morphologic Similarities in Turkish Words, CEUR Workshop Proceedings 3171 (2022) 161-176.

[25] O. Duda, V. Pasichnyk, H. Lypak, N. Veretennikova, N. Kunanets, O. Matsiuk, V. Mudrokha, Formation of Integrated Repositories of Social and Communication Data by Consolidating the Resources of Museums, Libraries and Archives in Smart Cities Projects, CEUR Workshop Proceedings 2870 (2021) 1420-1430.

[26] N.B. Shakhovska, Y.J. Bolubash, O.M. Veres, Big data federated repository model, in: Proceedings of 13th International Conference: The Experience of Designing and Application of CAD Systems in Microelectronics, CADSM, 2015, pp. 382-384.

[27] P. Kravets, V. Lytvyn, V. Vysotska, Y. Burov, I. Andrusyak, Game Task of Ontological Project Coverage, CEUR Workshop Proceedings 2851 (2021) 344-355.

[28] O. Oborska, M. Teliatynskyi, D. Dosyn, V. Lytvyn, S. Kostenko, An Intelligent System Based on Ontologies for Determining the Similarity of User Preferences, CEUR Workshop Proceedings 3403 (2023) 283-292.

[29] V. Hryhorovych, Calculation of the Semantic Distance between Ontology Concepts: Taking into Account Critical Nodes, CEUR Workshop Proceedings 3396 (2023) 206-216.

[30] V. Hryhorovych, Analysis of Scientific Texts by Semantic Inverse-Additive Metrics for Ontology Concepts, CEUR Workshop Proceedings 3171 (2022) 801-816.

[31] V. Hryhorovych, Construction of Semantic Metric for Measuring the Distance between Ontology Concepts, CEUR Workshop Proceedings 2870 (2021) 498-510.

[32] T. Basyuk, A. Vasyliuk, Approach to a Subject Area Ontology Visualization System Creating, CEUR Workshop Proceedings 2870 (2021) 528-540.

[33] T. Batiuk, L. Chyrun, O. Oborska, Ontology Model and Ontological Graph for Development of Decision Support System of Personal Socialization by Common Relevant Interests, CEUR Workshop Proceedings 3171 (2022) 877-903.

[34] D. Dosyn, Y. Ibrahim Daradkeh, V. Kovalevych, M. Luchkevych, Y. Kis, Domain Ontology Learning using Link Grammar Parser and WordNet, CEUR Workshop Proceedings 3312 (2022) 14-36.

[35] J. Chen, D. Dosyn, V. Lytvyn, A. Sachenko, Smart data integration by goal driven ontology learning, in: Advances in Intelligent Systems and Computing 529 (2017) 283-292.

[36] N. Khairova, A. Kolesnyk, O. Mamyrbayev, G. Ybytayeva, Y. Lytvynenko, Automatic Multilingual Ontology Generation Based on Texts Focused on Criminal Topic, CEUR Workshop Proceedings 2870 (2021) 108-117.

[37] V. Shynkarenko, L. Zhuchyi, Ontological Harmonization of Railway Transport Information Systems, CEUR Workshop Proceedings 2870 (2021) 541-554.

[38] S. Sachenko, S. Rippa, Ya. Krupka, Pre-Conditions of Ontological Approaches Application for Knowledge Management in Accounting, International Workshop on Antelligent Data Acquisition and Advanced Computing Systems: Technology and Applications 2009, 605-608.

[39] M. Davydov, O. Lozynska, Mathematical method of translation into ukrainian sign language based on ontologies, Advances in Intelligent Systems and Computing 871 (2018) 89-100.

[40] O.H. Lypak, V. Lytvyn, O. Lozynska, R. Vovnyanka, Y. Bolyubash, A. Rzheuskyi, D. Dosyn, Formation of Efficient Pipeline Operation Procedures Based on Ontological Approach, Advances in Intelligent Systems and Computing 871 (2019) 571-581.

[41] S. Lupenko, V. Pasichnyk, N. Kunanets, Organization of the Content of Academic Discipline in the Field of Information Technologies Using Ontological Approach, Advances in Intelligent Systems and Computing 871 (2019) 312-327. doi: 10.1007/978-3-030-01069-0_23.

[42] L. Serhii, P. Volodymyr, K. Nataliia, Axiomatic-deductive strategy of the organization of the content of academic discipline in the field of information technologies using the ontological approach, in: Proceedings of the IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT, 2018 - Proceedings, Vol. 1, Art No. pp. 387-390. doi: 10.1109/STC-CSIT.2018.8526687.

[43] I. Pelekh, A. Berko, V. Andrunyk, L. Chyrun, I. Dyyak, Design of a system for dynamic integration of weakly structured data based on mash-up technology, in: Proceedings of the IEEE 3rd International Conference on Data Stream Mining and Processing, DSMP, 2020, pp. 420–425. doi: 10.1109/DSMP47368.2020.9204160.

[44] A. Berko, et al., Information resources analysis system of dynamic integration semi-structured data in a web environment, in: Proceedings of the IEEE 3rd International Conference on Data Stream Mining and Processing, DSMP , 2020, pp. 414–419. doi: 10.1109/DSMP47368.2020.9204101.

[45] I. Galushka, S. Shcherbak, Devising a mathematical model for pattern-based enterprise data integration, Eastern-European Journal of Enterprise Technologies 2(9) (2015) 59-64.

[46] P. Kryndach, V. Vysotska, S. Chyrun, L. Chyrun, S. Goloshchuk, R. Holoshchuk, Analysis of Semantic Relationships in Ukrainian Text Content Based on Word2Vec and Machine Learning, in: Proceedings of the International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT, 2023.

[47] S. Albota, Creating a Model of War and Pandemic Apprehension: Textual Semantic Analysis, CEUR Workshop Proceedings 3396 (2023) 228-243.

[48] S. Albota, Modelling the Impact of the Pandemic on Online Communication: Textual Semantic Analysis, CEUR Workshop Proceedings 3171 (2022) 471-486.

[49] S. Albota, Linguistically Manipulative, Disputable, Semantic Nature of the Community Reddit Feed Post, CEUR Workshop Proceedings 2870 (2021) 769-783.

[50] S. Olizarenko, V. Radchenko, Method for determining the semantic similarity of arbitrary length texts using the transformers models, Advanced Information Systems 5(2) (2021) 126–130. doi:10.20998/2522-9052.2021.2.18.

[51] T. Basyuk, A. Vasilyuk, V. Lytvyn, Mathematical model of semantic search and search optimization, CEUR Workshop Proceedings 2362 (2019) 96–105.

[52] D. Hordiiuk, I. Oliinyk, V. Hnatushenko, K. Maksymov, Semantic Segmentation for Ships Detection from Satellite Imagery, in: Proceedings of the IEEE 39th International Conference on Electronics and Nanotechnology (ELNANO), 2019. doi:10.1109/elnano.2019.8783822.

[53] L. Savytska, N. Vnukova, I. Bezugla, V. Pyvovarov, M. Turgut Sübay, Using Word2vec Technique to Determine Semantic and Morphologic Similarity in Embedded Words of the Ukrainian Language, CEUR Workshop Proceedings 2870 (2021) 235-248.

[54] I. Gruzdo, I. Kyrychenko, G. Tereshchenko, O. Cherednichenko, Applıcatıon of Paragraphs Vectors Model for Semantıc Text Analysıs, CEUR workshop proceedings 2604 (2020) 283-293.