Ontological approach to the integration of knowledge from external sources

Korshunov R.A., Sadouski M.E., Zagorskij A.G.
Belarusian State University of Informatics and Radioelectronics
Minsk, Belarus
korshunov19101998work@gmail.com, sadovski@bsuir.by, alexandr.zagorskiy.work@gmail.com

***Abstract*—The article is dedicated to the problem of automatized provisioning of the knowledge base from external sources. Within the framework of the article, the classification of data sources by the type of stored data, the justification of the need to integrate data into the knowledge base, the analysis of existing approaches to solving the specified problem are given. Our own ontological approach is proposed as well as the implementation of the approach on the example of WikiData is considered.**

***Keywords*—OSTIS, ontological approach, integration of knowledge**

I. Introduction

A **knowledge base** is a finite information structure that is a formal representation of all the knowledge sufficient for the operation of a computer system and that is stored in its memory [1]. The knowledge base is a key component of any intelligent system. The main problem that it solves is the acceleration of the search for the necessary information, which is achieved through the accumulation of any intellectual accomplishments of experts in the subject domain.

Any knowledge base should be loaded to work with it. The process of loading the knowledge base has a number of features. Firstly, it is necessary to involve qualified experts who will be responsible for the correctness of the embedded knowledge. Secondly, it is necessary to spend a certain amount of labor and time resources.

Undoubtedly, any complication of certain processes causes a logical need to find less expensive solutions. Within the framework of this article, an automatized mechanism for integrating knowledge from external sources is proposed.

The implementation of this mechanism will allow solving a number of problems. Firstly, to reduce the time and labor costs for enriching the knowledge base. This may be particularly relevant when it is necessary to add a large amount of data to the knowledge base, which has a common, recurrent format (for example, data from tables of a relational database). Secondly, the proposed approach will allow using various external sources to load the knowledge base. This can be useful when it is necessary to unify several sources in one common resource. At the same time, such sources can radically differ from each other in the format of knowledge representation. Thirdly, the mechanism allows automating the process of integrating knowledge from external sources. As a result, the knowledge base of the intelligent system will be in a current, updated state.

From the point of view of the quality of knowledge in the knowledge base, the implementation of the proposed approach will allow:

- solving the problem of synonymy of source data, thereby reducing storage expenses and increasing the processing speed of the total knowledge base;
- unifying the format of information representation, which will allow producing new knowledge based on logical rules from already formed one;
- monitoring the consistency and quality of integrated knowledge.

II. Classification of data sources

Before getting to the proposed mechanism for integrating knowledge from external sources, it is worth paying attention to what these sources can be. Today, there are various data sources, but they generally can be classified according to the type of data they store. In turn, data can be divided into the following groups. This is [2]:

- structured data;
- unstructured data;
- semi-structured data;
- metadata.

Figure 1. The data types

A. Structured data

Structured data is data that corresponds to a predefined data model and can therefore be easily analyzed [2]. Structured data corresponds to a tabloid format with relations between different rows and columns. Common examples of structured data are Excel files or SQL databases. Each of them has structured rows and columns that can be sorted.

Structured data depends on a data model – a model of how data can be stored, processed and accessed. Because of the data model, each field is discrete and can be accessed separately or jointly with data from other fields. This makes structured data extremely powerful: it is possible to quickly aggregate data from various places in the database.

Structured data is considered as the most "traditional" form of data storage, since the earliest versions of database management systems (DBMS) could store, process and access structured data.

B. Unstructured data

Unstructured data is information that either does not have a predefined data model or is not organized in a predefined way [2]. Unstructured information usually contains a lot of text but can also contain such data as dates, numbers and facts. This leads to inaccuracies and ambiguities that make it difficult to understand the usage of traditional programs compared to data stored in structured databases. Unstructured data includes [3]:

- text sources: books, journals, presentations;
- media: MP3, digital photos, audio and video files;
- data from websites and social networks.

In recent years, the ability to store and process unstructured data has become much in demand, and many new technologies and tools have appeared on the market, which can store specialized types of unstructured data.

The ability to analyze unstructured data is especially relevant in the context of big data, since most of the data in organizations is unstructured.

C. Semi-structured data

Semi-structured data is a form of structured data, which does not correspond to the formal structure of data models connected to relational databases or other forms of data tables but nevertheless contains tags or other markers to separate semantic elements and provide a hierarchy of records and fields within data [2]. Therefore, it is also known as a self-describing structure. As examples of tools for representing semi-structured data, it is possible to distinguish [3]:

1. **XML markup language.** This is the language of semi-structured documents. XML is a set of rules for encoding documents that determine the format that can be read by a human and a computer (although the statement that XML is readable is of little consequence: anyone who tries to read an XML document can better master their time). Its value is that its tag-driven structure is very flexible and encoders can adapt it to generalize the data structure, storage and transport in the network.
2. **JSON** is another semi-structured data exchange format. Its structure consists of name/value pairs (or an object, hash table, etc.) and an ordered list of values (or an array, sequence, list). Since the structure is interchangeable between languages, JSON successfully handles data transfer between web applications and servers.
3. **NoSQL** is a type of database that differs from traditional ones in that they do not separate the organization (schema) from data. This makes NoSQL the best choice for storing information that does not fall within the format of a record and a table, for example, when the question is about the texts of various lengths. This also makes it easier to exchange data between databases. Some new NoSQL databases, such as MongoDB and Couchbase, also include partially structured documents, initially storing them in the JSON format.

The reason why this category exists (between structured and unstructured data) is that semi-structured data can much easier be analyzed than unstructured data. Many big data solutions and tools can "read" and process JSON or XML. This simplifies the analysis of structured data compared to unstructured data.

D. Metadata

Metadata is data about data [2]. From a technical point of view, this is not a separate data structure but one of the most important elements for big data analysis and big data solutions.

For example, in a set of photos, metadata can describe when and where the photos were taken. Then the metadata provides fields for dates and locations, which themselves can be considered as structured data. For this reason, metadata is often used by big data solutions for initial analysis.

From all the above, it can be concluded that as the simplest sources for integration, those will serve, which store structured data. This simplicity is based on the rigor of the model of such data. At the same time, the rigor of the model limits the data, makes it less diverse and makes it impossible to add something to them that does not fall within the framework of the model. In contrast to unstructured data, structured data is more flexible and diverse. However, the absence of any clear structure complicates the processing process and makes it impossible to write simple integration rules.

III. Overview of existing approaches

The problem of enriching the knowledge base is not new. Attempts to solve it have been made before. For example, a solution worthy of attention was proposed by the Bosch Center for AI.

The Bosch Center has developed a semantic data lake for automotive data as a centralized platform for developing and testing applications for autonomous driving. The architecture of the lake can be found in figure 2 [5].

The **"data lake"** is a centralized storage that allows storing all structured and unstructured data at any scale

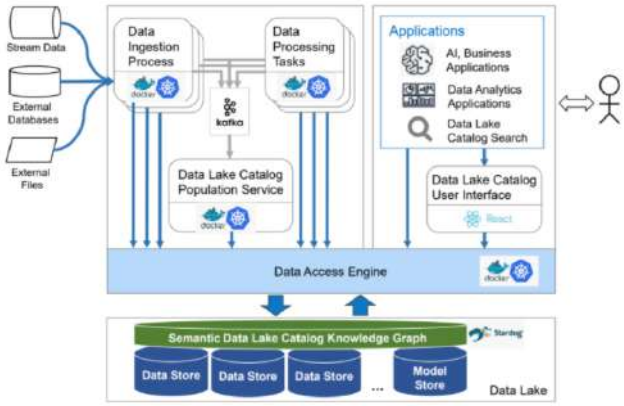

Figure 2. The architecture of the semantic data lake

[4]. A **"semantic data lake"** is a special form of data lake, in which the upper semantic layer enriches and connects data semantically. The semantic level overcomes the fragmentation of data and enables semantic search across all data [5]. The semantic data lake is more flexible than a regular data lake, which allows using the data stored in it more efficiently. This prevents the transformation of the data lake into a "data swamp" – a storage, in which potentially valuable information is simply stored on drives without usage.

A special DCPAC (Data Catalog, Provenance and Access Control) ontology was developed for data processing, which accumulates and coheres other W3C ontologies [5].

During the receiving and processing of new input data, the service for enriching the data lake catalog is launched, which automatically creates a semantic description and a layer on top of the data. The semantic level has the form of a knowledge graph. The service for enriching the data lake catalog reads the available metadata about the loaded data assets and creates the corresponding semantic data by reconciling, annotating and enriching the input data with the concepts of the DCPAC ontology. The resulting knowledge graph forms a semantic layer on the assets of the data lake and describes their contents, origin and access rights. This information plays a decisive role in the semantic data lake, since it semantically describes the stored data and, therefore, turns it into valuable information and information assets. This allows tracking aspects of the data related to the origin and determining access rights to data assets [5].

Semantic Web proposes to use its own technologies to integrate data from various sources. The essence of integration is to create mappings from the source format to the RDF [6] data model. For example, a special language R2RML [7] has been developed to display data from relational databases. An extended version of the language called RML also allows creating rules for displaying data from the CSV, TSV, XML and JSON formats [8].

It is also worth highlighting the approach described in the article [9]. It proposes the usage of a specially developed framework for integrating and supporting dynamic ontologies developed by means of Semantic Web.

Integration takes place with the help of several software components-wrappers. At first, the new ontology O_b is mapped to the initial one O_a, forming a third unified ontology O_c. The formed ontology O_c, as well as its two initial ones, are merged into a single integrated ontology O_d. At the same stage, semantic integration problems are being solved. Further, from the difference between the ontologies O_d and O_b, the specialist is offered to merge and create a new version of the knowledge base.

IV. Proposed approach

As mentioned earlier in Section II, data can be divided into 4 basic groups: structured, unstructured, semi-structured data and metadata. Various external languages are used to represent them. For example, natural languages are used to represent unstructured data. Formal languages are used to represent un-, semi-structured data and metadata.

Natural languages are understood as languages that developed in the usual way for communication between humans [10]. These include Russian, English and other existing languages. In turn, formal languages are languages created by a human to solve any specialized problems. These include all kinds of programming languages (C++, Python, Java), markup languages (XML, HTML), etc.

The essence of the proposed approach for the integration of knowledge is as follows. The address of some external source is input to the system. If the system supports the source format, then with the help of a special agent, it "pulls in" the proposed knowledge and converts it according to the format that is accepted within this system. Then, with the help of another agent, the system tries to integrate the acquired knowledge into the hierarchy of existing one.

Within the framework of the approach, it is necessary to solve some problems. **Firstly**, it is necessary to describe the syntax and semantics of the external language. **At the second step**, the problem is to build rules for the transition of knowledge from constructions in an external language to constructions in an internal language used in the system to represent knowledge. At this step, the previously obtained descriptions of syntax and semantics are applied. **The last stage** is the integration of the obtained knowledge with the knowledge base. This stage includes checking the correctness of the received constructions.

There may be difficulties at each of the stages. The first difficulty arises during the description of the syntax. Since each external language can have its own set of unique features that distinguish it from the internal language of

the system and other external languages, it is necessary to describe the concepts, relations and rules for transiting the constructions of the external language into the constructions of the internal language. For these purposes, the subject domain for a specific external language should be described. This subject domain is inherited from the *Subject domain of information constructions and languages* and the *Subject domain of entities*.

In addition to the problems described above, there are difficulties in the final integration of the obtained constructions. It is not enough just to immerse them in memory – they need to be correctly implemented into existing knowledge while eliminating synonymy, contradictions and other problems.

V. Implementation of the proposed approach

As an example, the process of integrating knowledge from WikiData into the ostis-system knowledge base will be considered. **WikiData** is an open and free knowledge base. Its purpose is to represent the actual information from Wikipedia in a compatible, machine-readable format [11].

The OSTIS Technology is a complex (family) of technologies that provide design, production, operation and reengineering of intelligent computer systems (ostis-systems) designed to automate a variety of human activities and that are based on the semantic representation and ontological systematization of knowledge as well as agent-oriented knowledge processing [12].

Among the advantages of ostis-systems, it is possible to distinguish:

- the ability to implement semantic integration of knowledge in its memory at a high level;
- the ability to integrate different types of knowledge;
- the ability to integrate various problem-solving models.

The json-format is used to represent knowledge in WikiData. This is a text format for data exchange. Its basic syntactic units are:

- a “key-value” pair;
- an ordered set of values.

As an example, let us consider a json-fragment with information about the city of Minsk. It consists of three basic blocks:

- “entities”;
- “relations”;
- “triplets”.

The block “entities” includes the entity “Minsk” as well as all the entities, with which “Minsk” is related. The block “relations” includes all relations, in which the entity “Minsk” participates. The block “triplets” includes all the triples “entity-relation-entity”.

To add knowledge from WikiData to the ostis-system knowledge base, it is necessary to convert json to the SC-code. This is an internal language used in ostis-systems. For the operation of intelligent computer systems built on the basis of the SC-code, in addition to the method of abstract internal representation of knowledge bases (SC-code), there are several ways of external representation of abstract sc-texts that are convenient for users and used in the design of the source texts of the knowledge bases of these intelligent computer systems and the source texts of fragments of these knowledge bases, as well as used to display various fragments of knowledge bases to users according to user requests. As such methods of external display of sc-texts, the following ones are distinguished:

- SCg-code, the texts of which are graph structures of a common type;
- SCs-code, the texts of which are strings of characters;
- SCn-code, the texts of which are two-dimensional character matrices that are the result of formatting, two-dimensional structuring of the texts of the SCs-code.

To solve the integration problem, the usage of the ostis-system is advisable for many reasons. The technology initially provides tools for describing the syntax and semantics of external languages. This toolkit allows reducing the development time by several times.

To integrate knowledge, a special translator, as well as the *Subject domain of WikiData*, are required.

Subject domain of WikiData

∋ *class of study objects′*:
- *wiki-entity*
- *wiki-relation*

∋ *relation being studied′*:
- *wiki-identifier*
- *wiki-analog*

Let us consider the process of generating sc-constructions using the example of the entity “belarus” from the block “entities”. First, an sc-node with the identifier “belarus” is created. Since “belarus” is located in the block “entities”, an arc of belonging is generated between the class “wiki-entity” and the node “belarus”. Then it is necessary to determine the wiki-identifier of this entity. For these purposes, an sc-link with the content “belarus” is generated, and then a relation “wiki-identifier*” is generated between the sc-link and the sc-node. At the end, it is necessary to generate the main identifier of the entity. To do this, the sc-link “Belarus” is generated, which belongs to the class “English language”. Then the relation “main identifier*” between the sc-node of the entity and the sc-link is generated. The same procedure is repeated for all other entities.

The process of generating relations from the block “relations” is similar to generating entities, just this once, instead of the class “wiki-entity”, an arc of belonging is generated between the class “wiki-relation” and the sc-node of the relation.

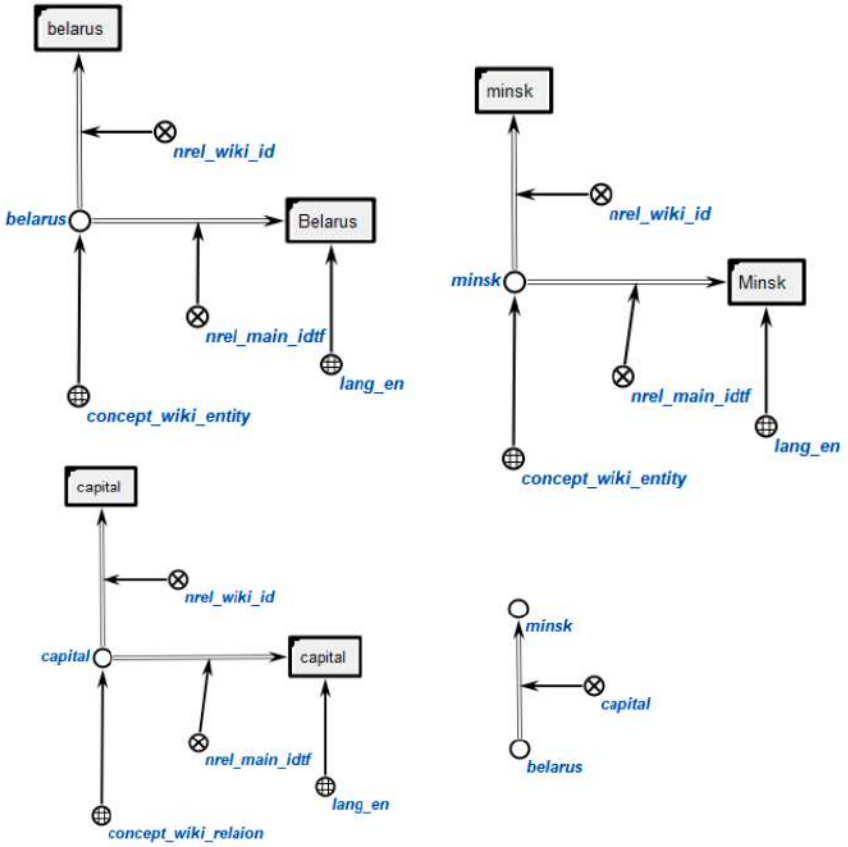

Figure 3. Generated wiki-entities and wiki-relations

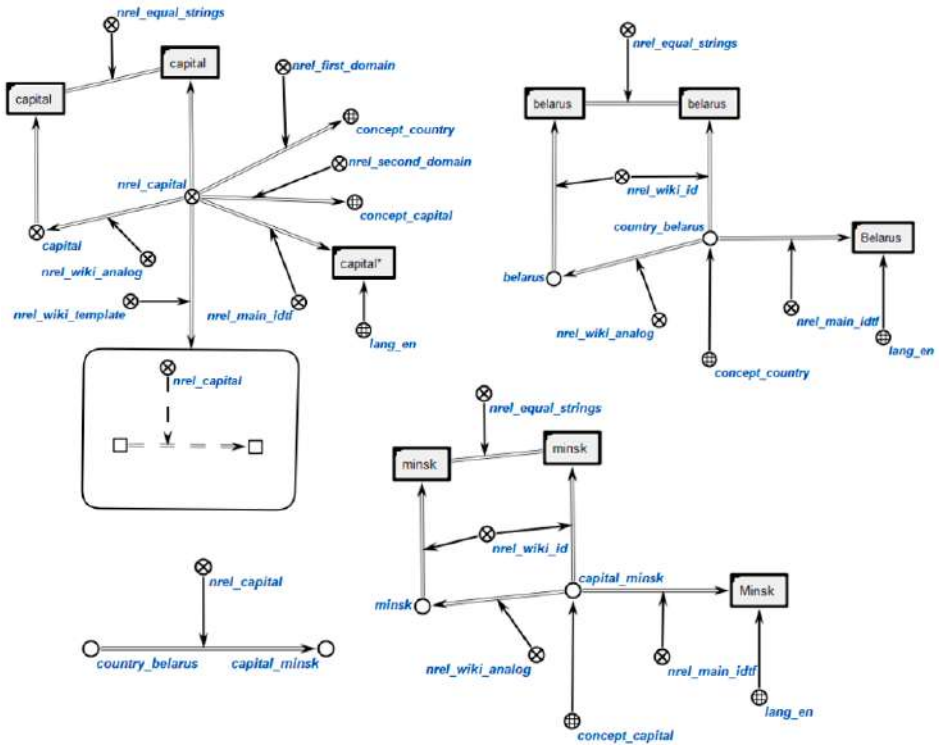

Figure 4. The result of the integration of knowledge

Finally, it is necessary to generate constructions from the block “triplets”. Previously obtained wiki-entities and wiki-relations are used for them. The resulting constructs are shown in fig. 3.

It is important to note that the generated constructions are temporary. Next, they should be integrated with the permanent constructions of the ostis-system knowledge base. In general, there can be two main situations for each generated wiki-relation or wiki-entity: in the knowledge base either there are analogs for them or not. In the previously represented fragment of the *Subject domain of WikiData*, the relations “wiki-identifier*” and “wiki-analog*” were specified. The relation “wiki-identifier*” connects the generated entity (or a relation) with the identifier that was used in WikiData. The relation “wiki-analog*” connects an entity (or a relation) that exists in the knowledge base with the generated wiki-entity (or a wiki-relation). Obviously, there may be a situation when there is an analog in the knowledge base for a wiki-entity (or a wiki-relation), but they are not connected by the relation “wiki-analog*”. Thus, the following scenarios appear:

- there is an analog for a wiki-entity (or a wiki-relation), and they are connected by the relation “wiki-analog*”;
- there is an analog for a wiki-entity (or a wiki-relation), but they are not connected by the relation “wiki-analog*”;
- there is no analog for a wiki-entity (or a wiki-relation).

Next, each of the three situations is considered by examples.

It is necessary to check whether the wiki-relation “capital” is connected with some other relation by the binding “wiki-analog*”. Such a relation was found. This is “nrel_capital”.

A similar check is carried out for the wiki-entity “belarus”. The entity connected to it by the relation “wiki-analog*” was not found, so the wiki-identifier is used for the search. Thanks to the identifier, the entity “country_belarus” was found.

The wiki-entity “minsk” is considered lastly. When searching through the relation “wiki-analog*” or the wiki-identifier, no matches were found. Therefore, the entity “capital_minsk” was generated.

After all, it is necessary to generate arcs between the found analogs. The result is a binding of the non-role relation “nrel_capital”, which connects “country_belarus” and “capital_minsk”.

The result of the integration described above is shown in figure 4.

VI. Conclusion

The mechanism proposed in the article ensures the integration of knowledge from external sources into the knowledge base of an intelligent system built on the basis of the OSTIS Technology.

The essence of the approach is the ontological description of the syntax and semantics of the external language, the construction of rules for the transition of knowledge from a representation in an external language to a representation in an internal language and the subsequent integration of the acquired knowledge with existing knowledge in the knowledge base.

The approach was applied to integrate WikiData with the ostis-system and is considered in the article.

The proposed mechanism allows solving the problem of synonymy of source data, unifying the format of information representation and monitoring the consistency and quality of integrated knowledge.

References

[1] Golenkov, V. V. A Project of an Open Semantic Technology for the Component Design of Intelligent Systems. Part 1: Principles of Creation. Ontologiya proyektirovaniya [Ontology of Design], 2014, № 11, pp. 42–64.

[2] Data Types: Structured vs. Unstructured Data. Available at: https://www.bigdataframework.org/data-types-structured-vs-unstructured-data/(accessed 2021, June)

[3] Datamation. Structured vs. Unstructured Data. Available at: https://www.datamation.com/big-data/structured-vs-unstructured-data/ (accessed 2021, June)

[4] Amazon AWS: What is a data lake?. Available at: https://aws.amazon.com/ru/big-data/datalakes-and-analytics/what-is-a-data-lake/ (accessed 2021, June)

[5] H. Dibowski, S. Schmid. Using Knowledge Graphs to Manage a Data Lake. Lecture Notes in Informatics (LNI), Gesellschaft für Informatik. Bonn, 2021, pp. 41-50.

[6] RDF 1.1 Concepts and Abstract Syntax. Available at: https://www.w3.org/TR/rdf11-concepts/ (accessed 2021, June)

[7] R2RML: RDB to RDF Mapping Language. Available at: https://www.w3.org/TR/r2rml/ (accessed 2021, June)

[8] RDF Mapping Language (RML). Available at: https://rml.io/specs/rml/ (accessed 2021, June)

[9] Vit Novacek, Loredana Laera, Siegfried Handschuh, Brian Davis. Infrastructure for Dynamic Knowledge Integration – Automated Biomedical Ontology Extension Using Textual Resources. Journal of Biomedical Informatics, 41(4), 2008.

[10] Cambridge Dictionary. Available at: https://dictionary.cambridge.org/ (accessed 2021, June)

[11] D.Hernández, A.Hogan, M.KrötzSCh. Reifying RDF: What Works Well With Wikidata? Proceedings of the 11th International Workshop on SCalable Semantic Web KB Systems, 2015, pp. 32–47.

[12] IMS.OSTIS Metasystem. Available at: http://ims.ostis.net/. (accessed 2021, June)

Онтологический подход к интеграции знаний из внешних источников

Коршунов Р. А., Садовский М. Е., Загорский А. Г.

В рамках данной статьи предлагается автоматизированный механизм интеграции знаний из внешних источников.

Суть подхода заключается в онтологическом описании синтаксиса и семантики внешнего языка, построении правил перехода знаний с представления на внешнем языке на представление на внутреннем языке и последующей интеграции полученных знаний с уже существующими знаниями в базе знаний.

Реализация подхода рассмотрена на примере интеграции WikiData с ostis-системой.

Внедрение данного механизма позволит решить ряд задач: сокращение временных и трудовых затрат на пополнение базы знаний, использование различных внешних источников, автоматизация процессов интеграции знаний.

С точки зрения качества знаний реализация предлагаемого подхода позволит решить проблему синонимии исходных данных, унифицировать формат представления информации, следить за целостностью и качеством интегрируемых знаний.

Received 01.06.2021