UDC 004.912
DOI: 10.15587/2706-5448.2021.243587
Article type «Reports on Research Projects»

Victoria Kostenko,
Olga Bulgakova,
Barbara Stelyuk

ANALYSIS OF APPROACHES FOR IDENTIFICATION THE ONTOLOGICAL MODEL COMPONENTS OF THE SEARCHING SYSTEM

The object of research is the components of an intelligent system for searching information in electronic repositories of unstructured documents, which based on the ontologies of the subject area. One of the most problematic areas is the processing and analysis of information contained in electronic repositories of unstructured documents. There are considered the some possibilities of increasing the efficiency of information processing. In the course of the study, using the method in which ontologies comprise sets of terms presented in it. In addition, the ontological set also includes information about subject areas, areas of definitions, etc. There are obtained the sequence of defining the conceptual representation of an intelligent search system based on ontological components. There are presented the composition of the ontological system model. There are described the main functional components of the system for intelligent processing of information about electronic documents.

The proposed approaches for identifying the component components of the ontological model of the search system have a lot of features. This is due to the fact that the search system model must have a set of properties: integrity, coherence, organization, integrability, mobility. Ontologies which representing the basic concepts of the domain in a format available for automated processing in the form of a hierarchy of classes and relationships between them allow automated processing. The using of ontologies in the role of an intermediary between the user and the search process, between the search process and the search system that can facilitate the solution of a number of complex and non-standard tasks of information retrieval (for example, the automation of the search process). It is possible to solve the problem of knowledge representation for displaying information relevant to user requests, as well as to solve the problems of filtering and classifying information. Compared to similar well-known search systems, this provides such advantages as creating a common terminology for software agents and users, protecting the information store from total overflow and errors, as well as solving the issue of information aging.

Keywords: *intellectual search system, processing of information, subject area ontologies, semantic system, knowledge bases systems.*

Received date: 31.05.2021
Accepted date: 29.07.2021
Published date: 07.12.2021

How to cite
Kostenko, V., Bulgakova, O., Stelyuk, B. (2021). Analysis of approaches for identification the ontological model components of the searching system. Technology Audit and Production Reserves, 6 (2 (62)), 10–14. doi: http://doi.org/10.15587/2706-5448.2021.243587

1. Introduction

Today, the current definition of the concept of «architecture» of an information system is the definition formulated in the ISO/IEC/IEEE 42010 Systems and Software Engineering – Architecture Description standard. According to this definition, the architecture of a system is the fundamental concepts and properties of a system in its environment, embodied in its elements, relationships, as well as the principles of its design and development [1]. This definition is the most general definition suitable for describing the architectures of almost any system, including information retrieval systems. In most cases, the problem of forming the architecture of such systems, which, on the one hand, would be the best for the representatives of the enterprise customer of the system, and on the other hand, would meet the capabilities of the enterprise developer of the system, should be considered unsolved by now [2].

A promising direction for solving the problem of adapting a software system to changes in the environment is the use of knowledge engineering approaches, in particular ontological modeling. When using model approaches, it is necessary to focus on compiled models or elaboration of rules. In contrast to model approaches, in the ontological modeling approach, a formal model of the subject area (ontology) is built taking into account its features and limitations. Such a model can be reused to build other software systems in the same subject area [3].

Intelligent systems are used in various subject areas to solve complex problems. They are knowledge systems. Knowledge is understood as the subject's ideas about the phenomena and laws of the external, in relation to the subject, the world [4].

Knowledge is information. In contrast to data, it is characterized by internal interpretability, structuredness and coherence. Intelligent systems operating with knowledge include systems that exhibit the ability for purposeful behavior [5].

Complex systems can be represented by more than software systems, although software systems are usually basic and necessary components. For example, when designing large-scale domain applications, it is necessary to take into account several types and levels of human and society interaction with physical-software systems and socio-technical environment, and also include cultural, legal and economic components [6].

The main goals of using intelligent systems are to reduce the time for making decisions or improve their quality or both. Problems to be solved include information retrieval, document annotation, report preparation, data extraction from documents, decision support in design and process management, etc.

To solve the problems of obtaining, analyzing, processing and presenting knowledge, a number of types of intelligent systems are used. These are expert systems, information retrieval systems, decision support systems, operational analytical data processing systems, knowledge management systems, etc. Their knowledge is represented in the form of taxonomies, ontologies, semantic networks, etc. Systems for collecting, storing, searching and issuing knowledge are called knowledge bases. That is, a knowledge base (KB) is a systematized repository of unstructured information. Knowledge base systems are being created, otherwise KBS (knowledge-based systems). They include, in addition to data, also knowledge management tools, modeling and assessment of situations, inference and decision support. The models of many KBSs are based on the ontology of subject areas. In particular, ontologies should contain, if not all, then at least a greater number of concepts that characterize possible requests for finding solutions and appear in the metadata of the KBS documents [4].

In general terms, ontology is understood as a system of concepts of a certain subject area, which is represented by a set of entities. Of course, they must be connected with each other by relationships. Ontologies are used for the formal specification of concepts and relationships that characterize a specific area of knowledge. The advantage of ontologies as a way of representing knowledge is their formal structure, which simplifies their computer processing.

It is possible to talk about the implicit application of ontologies as systems of concepts in the natural sciences (biology, medicine, geology, etc.), where they serve as a kind of foundation for the construction of theories. Taxonomy is an integral part of any ontology and it is possible to talk about the presence of ontology elements in special classifications and indexing systems.

Search engines must work in huge amounts of information, providing the user with a result – fast and accurate. It is believed that systems based on the ontological approach are more advanced and meet the needs of users.

Ontology needs to be applied as an intermediary between the user and the search process, between the search process and the search engine. To build an ontology requires a formal declarative presentation of clearly organized structures containing a vocabulary of terms in the thematic area, a description of the definitions of these terms, the existing relationships between them and, in general, theoretically possible and impossible relationships.

The ontology is used during:

1) development and analytics of the request;

2) control of the characteristics of information in electronic databases of documents (aging and compliance).

Explicitly, ontologies are used as data sources for many computer programs. For example, for information retrieval, text analysis, knowledge extraction. This allows more efficient processing of complex and varied information.

The process of designing and developing ontologies is called ontological engineering.

Understanding and analyzing the concepts that must be represented in an ontology are fundamental problems in the development of ontologies. They are two of the main aspects of ontology development that differentiate it from software or even systems development. In the development of an ontology, there are (at least) two phases for representing a concept or object. First, it is an understanding of what a concept or entity is. The second stage is the stage of analysis, in which a concept or object is presented (or modeled) using the selected or available constructs. For example, someone may choose OWL as an ontology language or work in the context of a basic, reference, or subject ontology [6].

However, the development of ontologies has more than advantages. The tasks of ontological engineering are poorly formalized. The problem is the need to identify methods for developing a system for effective data retrieval in electronic document repositories.

The basis of the ontology is made up of many terms presented in it. However, not only. The ontological set also includes information about subject areas, domains of definitions, etc.

Ontology is called such a scheme, which consists of classes related to each other by different relationships and rules. This is a peculiar form of representing a certain area of knowledge in a formal form [7].

In the modern world, ontologies are widely used in programming, teaching, and various research projects.

In ontology modeling, knowledge of what an ontology is needed for and how detailed or unique it can be is relevant. Since ontology is a model of the real world, the concepts in it must also reflect reality.

Owing to ontologies that represent the basic concepts of the subject area in a format available for automated processing in the form of a hierarchy of classes and relations between them, ontologies allow automated processing of the semantics of information units [8]. In particular, the use of ontologies of problem situations allows to save and reuse knowledge about decision making and use intelligent software agents to support decision making. In addition, it is important that to build an ontology of a problem situation, it is advisable to use knowledge from the already described existing ontologies. But this is a very difficult task, since due to the presence of ontologies similar in competence, the principle of multiple ontologies should be taken into account. Today, this principle manifests itself both at the level of top-level ontologies (SUMO, DOLCE, UFO, GFO), which use various principles of construction, and at the level of industry-specific ontologies [9].

Modern information retrieval models implemented in many information systems in most cases do not use the knowledge described in special thesauri and ontologies. Most often they are based on the use of a common set of words. In this case, a technique is used that takes into account only the frequency of occurrence of words in a sentence, text or set of documents.

The search task becomes more complicated in cases of developing large data sets, the so-called Data Lakes. Such data arrays can include structured data from relational databases (these are strings and strings), semi-structured data (CSV, log files, XML, JSON), unstructured data (messages, pdf documents) and even binary data (video, audio, graphic files).

The relevance of research on this topic is determined by the fact that many search tasks can be automated using ontologies. The basis of the ontology is made up of many terms presented in it. However, not only terms. The ontological set also includes information about subject areas, domains of definitions, etc.

Thus, *the object of research* is the component components of an intelligent system for searching information in electronic repositories of unstructured documents based on the ontologies of the subject area. *The aim of research* is to identify the main components of an intelligent electronic document search system based on domain ontologies, it is advisable to use it for information and analytical services of organizations, enterprises or private firms.

2. Methods of research

The process of creating ontologies can be iterative and can continue throughout the entire life cycle of an ontology.

Let's consider the construction algorithm described in [10].

Step 1. Determination of the scope and scale of the ontology:

It is proposed to start the development of an ontology by defining its scope and scale (i. e., we will answer several basic questions: What area will the ontology cover? Why use an ontology? What types of questions should information in the ontology answer? Who will use and maintain the ontology?).

The answers to these questions may change during the design of the ontology, but at any given point in time they help to constrain the scale of the model.

Step 2. Consideration of options for reusing existing ontologies:

It is necessary to check whether it is possible to improve and expand the existing sources for a specific subject area and task. Reuse of existing ontologies may be required if the system needs to interact with other applications that are already included in separate ontologies or controlled vocabularies. Many ontologies are already available in electronic form and can be imported into the used ontology design environment.

Step 3. Listing important terms in the ontology:

It is helpful to make a list of all terms that are relevant to a specific subject area.

Step 4. Defining classes and class hierarchy:

There are several possible approaches to designing a class hierarchy:

1. The process of top-down development begins with the definition of the most general concepts of the subject area, followed by the concretization of the concepts.

2. The bottom-up development process begins by defining the most specific classes, and then grouping these classes into more general concepts.

3. The blended development process is a combination of top-down and bottom-up approaches: first define the more visible concepts, then generalize and restrict them appropriately.

Step 5. Defining the properties of the slot classes:

Classes by themselves do not provide enough information to answer the *Step 1* proficiency test questions. After defining a number of classes, it is necessary to describe the internal structure of the concepts.

In *Step 3*, there are already selected classes from the list of terms. Most of the rest of the terms are likely to be properties of these classes.

Step 6. Defining facet slots:

Slots can have different facets describing the type of value, allowed values, number of values (cardinality), and other properties of values that the slot can take.

Step 7. Instantiation:

The last step is to create separate instances of the classes in the hierarchy. To define a separate instance of a class, it is necessary:

1) choose a class;

2) create a separate instance of this class;

3) enter the value of the slots.

Creation of ontologies is an iterative process and is performed by a person using supporting software tools. The general approach to ontology creation includes several stages:

1) definition of goals and boundaries of the described application;

2) selection of basic concepts with the definition of sets of synonyms (synsets);

3) definition of slots;

4) definition of types and values (facets) of slots;

5) definition of instances of concepts;

6) verification of the created ontology [10].

There are top-down, bottom-up, and mixed development styles. In a top-down style, the choice of concepts begins with more general concepts, while in a bottom-up style, on the contrary, with more specific ones.

To create and maintain ontologies, there is a number of ontology engineering tools that, in addition to general editing and viewing functions, perform:

- support for documenting ontologies, displaying, aligning and combining several ontologies, as well as annotation tools;
- import and export of ontologies of different formats and languages;
- support for graphic editing;
- management of ontology libraries, etc.

Examples of ontology editors are Ontolingua, Protege, OntoEdit, OilEd, WebOnto, ODE, KADS22, FODA, DSL Tool VS.Net. The degree of compliance of a class document (ontology) is calculated as the sum of the weights of all terms of a given ontology found in the document:

$$R_{dC} = \sum_{t \in C} w_{td},$$

where R_{dC} – the compliance degree of document d with cluster C; w_{td} – weight of the term t in document d.

There are different approaches to weighting the terms in the document, it is possible to find out about them in the source [11]. Taking into account the fact that the concepts in ontologies are divided by roles, it is necessary to introduce different weights for the roles, as well as to separately process complex concepts. The following weighing method has proven itself well:

$$w_{td} = \begin{cases} tf, \text{ if the concept is simple and its role is «object»}, \\ 0.1 \cdot tf, \text{ if the concept is simple}, \\ (1+k) \cdot tf, \text{ if the concept is complex}, \end{cases}$$

where tf – the number of occurrences of the concept in the document; k – coefficient taking into account the complexity of the concept.

3. Research results and discussion

The software architecture includes the components of presentation, services, business logic, data access, and end-to-end functionality that must ensure the interaction of users and external systems with data sources.

For a general presentation of the functionality of the system software, let's use the approach described in [12]. According to this approach, an intelligent search system can have a client-server architecture. The presence of independent software modules increases the fault tolerance and reliability of such a system.

Each data storage format is a software container and requires the implementation of separate mechanisms for retrieving this data in the system.

The search and download mechanism connects the system at the input of the source of text documents. Each electronic document is processed – indexed. This is done in order to simplify further access to the source and finding the appropriate slot. Electronic text files in different languages with the extensions *.txt, *.doc, *.docx are indexed.

The processing of electronic texts should be detailed:

1. Initial processing of a text document in the system (identification, reduction to a single coding style, removal of unnecessary elements, dividing the document into components).
2. Conducting linguistic analysis.
3. Certain actions for the implementation of the vector (matrix) representation of texts.

The classification of the components in the document is also made:

1. Additional processing of texts (creation of thesauri, assessment of the possibility of actions to classify a document).
2. Selection of essential fragments (in this case, let's rely on regular expressions, on logical and statistical rules; let's correct the classification results).

The next component is the analysis of the results:

1. Determination of duplicate documents (these actions involve the identification of documents that have the same meaning, but different types).
2. Rubric of collections of documents.
3. Compilation by headings.
4. Formulation of decision rules and assessment of the quality of training.
5. The learning process depends on the ontology. For each ontology, the following measure is formed: the significance and maximum allowable measure of the search weight.
6. Formation of test sets for headings of documents and evaluation of special parameters for building models.
7. Determination of the resulting rules for individual components and classifiers as a whole.
8. Formation of a report on learning outcomes (description of decision rules and headings terminology, creation of recommendations, description of relationships) [12].

Checking the test collection and adjusting the classification rules:

1. Test collection – the collection by which the system parameters are configured.
2. Checking the examples trained separately for each of the defined headings (this is done by analyzing the added and missing documents in the headings).
3. Correction of the classification rules for individual headings.

Indexing process:

1. Among the set of documents (the size varies), only those that meet the condition are selected.
2. Work with indexes is carried out using software libraries that work with the values of the document index. This can also include displaying additional information, using the mechanism for analyzing the content of the document, storing additional information about the positions of certain words in the body of the document [12].
3. Using the specified libraries is a means of combining indexing and local search components.

The process of clustering large collections of documents in different languages:

1. Formation of groups of homogeneity in the text collection.
2. Definition of reference information for user groups.
3. Conducting secondary clustering as needed.

The process of accessing information includes:

1. Formation of a request to the search engine.
2. Receiving a response from the system.
3. Changes to the parameters of ranking documents (document elements, specific terms and lexemes).
4. Correction of the sequence algorithm for making a decision to refer a document to a certain ontology.
5. Formation of a statistical report on the work of the system.

Let's define a conceptual representation of an intelligent search system based on ontological components. The main functional components of the system for intelligent processing of information about electronic documents will be considered:

1. Data warehouse (may include ontology classes, index databases, a list of links, thesauri and classifiers).
2. A block for extracting metadata from documents (may include such components as: indexing, working with WEB-dictionaries, working with electronic documents, queries to remote databases and clustering).
3. System administration interface (may include the following components: tasks of the structure of the resource catalog, links and the formation of ontologies).
4. Data administration interface (may include the following components: cataloging, ontology update, wordform generation).
5. User interface for searching documents (thematic collections are displayed, user queries are created and statements are checked).

An intelligent search system designed as a Semantic Network of Design Patterns (SNDP) can have a client-server architecture and consist of independent software modules, which greatly increases the fault tolerance and reliability of such a system. The SNDP is formed thanks to the algorithm, and the formation of metadata of new documents can cause the appearance of new concepts in ontologies.

The search engine model should have the following properties:

- *integrity*. The system will be viewed as a whole. It consists of interacting modules, possibly heterogeneous, but at the same time point-compatible with each other. At the same time, the possibility is not excluded that some blocks exist separately and are connected at the right time. It is not about a violation of the integrity of the system;
- *connectivity*. The presence of significant stable connections between elements and their properties, and from a systemic point of view, not any, but only essential

connections that determine the integrative properties of the system are of importance;

– *organization.* The presence of a certain structural and functional organization, here it is possible to add one of the processes of the system – processing regular expressions;

– *integrability.* The presence of qualities inherent in the system as a whole, but not inherent in any of its elements separately, that is, the properties of the system, although they depend on the properties of the elements, are not completely determined by them;

– *mobility.* The ability to quickly rebuild the model and system for emerging circumstances. A prerequisite is to add the process of «self-learning» of the system.

So, ontologies:

– can and could solve the problem of knowledge representation to display information relevant to user requests;

– would allow to filter and classify information;

– would allow to engage in the creation of a common terminology for software agents and users;

– would help to protect information stores from total overflow and errors;

– are considered one of the ways to address the issue of information aging.

At the time of practical implementation, one should consider:

1) the tasks of ontological engineering, poorly formalized;

2) it is necessary to resolve non-standard situations that arise when performing information retrieval procedures. The self-learning process of the system should be mandatory in the intelligent search system. It is believed that this process will eliminate situations with terms or names that are not recorded correctly in data stores.

It is assumed that in the process of self-learning, rules or functions should be built, differentiated by situations, which the system should use when unfamiliar or non-standard situations arise. A dictionary of terms, rules and conditions will be automatically generated from the generalized rules. Let's continue our research in this direction.

Further research can be related to identifying the characteristics and features of the identified components of the ontology for the implementation of the smart search system. The selection of algorithms and methods for individual components of the system, analysis and selection of the best methods and methods for implementation in the blocks of the search system continues.

4. Conclusions

The use of ontologies for automated processing of the semantics of information units is appropriate and effective due to the fact that they represent the basic concepts of the domain in a format that is available for automated processing in the form of a hierarchy of classes and relationships between them.

The general view of the functionality of an intelligent search system based on the use of ontologies can have a client-server architecture, which assumes the presence of independent software modules, which makes it possible to increase the fault tolerance and reliability of the software. This approach, of course, requires the implementation in the system of separate mechanisms for extracting, searching and loading data with different storage formats. An example of such a mechanism is the conversion of electronic text files in different languages with the extensions *.txt, *.doc, *.docx, when each electronic document is indexed. This certainly makes it easier to further access the source and find the appropriate slot. For the study, a simplified test version of the software module was implemented, which performs the functions of an ontological unit and search in text documents.

In the course of the study, the possibilities of improving the efficiency of information retrieval in arrays of unstructured documents were identified.

The analysis of approaches to defining the conceptual representation of an intelligent search system based on ontological components makes it possible to determine the composition of the system model and describe the main functional components of the system for intelligent information processing in electronic document storages.

It is the possibility of automatic formation of dictionaries of terms, rules and search conditions within the framework of certain complex subject areas and the need to explicitly set search criteria in arrays of unstructured or semi-structured documents that are of greatest interest from the point of view of the practical application of research results.

References

1. *ISO/IEC/IEEE 42010*. Available at: http://www.iso-architecture.org/ieee-1471/index.html
2. Evlanov, M. V. (2013). Ontological model of information system architecture, based on service approach. *Radioelektronika, informatika, upravlinnia, 2,* 130–135.
3. Burov, Ye. V., Pasichnyk, V. V. (2015). Prohramni systemy na bazi ontolohichnykh modelei zadach. *Informatsiini systemy ta merezhi, 829 (2),* 36–57.
4. Norenkov, I. P. (2010). Intellectual Technologies on the Base of Ontologies. *Informatsionnye tekhnologii, 1,* 17–23.
5. Bashmakov, A. I., Bashmakov, I. A. (2005). *Intellektualnye informatsionnye tekhnologii*. Moscow: MGTU im. N. E. Baumana, 304.
6. Schneider, T., Hashemi, A., Bennett, M., Brady, M., Casanave, C., Graves, H. et. al. (2012). Ontology for Big Systems: The Ontology Summit 2012 Communiqué. *Applied Ontology, 7 (3),* 357–371. doi: http://doi.org/10.3233/ao-2012-0111
7. Belousova, I. D., Kurzaeva, L. V., Agdavletova, A. M. (2015). K voprosu o soglasovanii trebovanii k soderzhaniiu professionalnoi podgotovki na osnove ontologicheskoi modeli. *Sovremennye naukoemkie tekhnologii, 11,* 67–70.
8. Pikuliak, M. V. (2014). Ontological approach to construction of subject sphere on basis of quantum frame model. *Medychna informatyka ta inzheneriia, 1,* 50–54.
9. Karpov, I., Burov, Y. (2020). Use of ontological networks in decision support systems under ambiguity. *Journal of Lviv Polytechnic National University «Information Systems and Networks», 7,* 8–15. doi: http://doi.org/10.23939/sisn2020.07.008
10. Noy, N. F., McGuinness, D. L. (2001). Ontology Development 101: A Guide to Creating Your First Ontology. *Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880.* Available at: http://protege.stanford.edu/publications/ontology_development/ontology101.html
11. Antonov, I. V. (2011). Model ontologii predmetnoi oblasti dlia sistem semanticheski-orientirovannogo dostupa. *Elektrotekhnika, 12,* 339–343.
12. Paliukh, B. V., Sotnykov, A. N., Yvanov, V. K. (2013). Architecture of intelligent information support system for innovations in science and education. *Software & Systems, 4,* 203–208.

✉ ***Victoria Kostenko,*** *Senior Lecturer, Department of Computer Science and Software Engineering, University of Customs and Finance, Dnipro, Ukraine, e-mail: viktko@ukr.net, ORCID: https://orcid.org/0000-0003-3847-2110*

Olga Bulgakova, *Senior Lecturer, Department of Computer Science and Software Engineering, University of Customs and Finance, Dnipro, Ukraine, ORCID: https://orcid.org/0000-0001-9834-2970*

Barbara Stelyuk, *PhD, Associate Professor, Department of Computer Science and Software Engineering, University of Customs and Finance, Dnipro, Ukraine, ORCID: https://orcid.org/0000-0002-2692-088X*

✉ *Corresponding author*