Approach to a Subject Area Ontology Visualization System Creating

Taras Basyuk and Andrii Vasyliuk

Lviv Polytechnic National University, Bandera str.12, Lviv, 79013, Ukraine

Abstract

This article analyzes the existing technologies and systems for displaying ontologies, presents the features and shows the need to create a new web-based visualization system of this data structure. The display structures analysis is performed and the ontology representation peculiarities in the form of frame models and application of Simple Knowledge Organization System (SKOS) for the Network router concept are shown. The ontology construction stages of the subject area are outlined and described, which include: formation of the list of concepts included in the ontological structure of the upper level; ordering the list of concepts; creation of a glossary; mapping arcs between created concepts. An example of building an ontology for the subject area «information security» in describing the concept features of «network computer attack». The main functions of the system are highlighted and the stage of its design is performed using the object-oriented approach and UML language tools. The diagram of precedents which visualizes the basic variants of the developed system usage is shown. The construction architecture was chosen in favor of three-tiered client-server architecture and the peculiarities of interaction between levels are described. The development language choice peculiarities are given and the choice of libraries and frameworks is proved, which ensure its effective functioning. The result of the work is a prototype of a software system for visualization of ontologies, which can be used to visualize the specified data structure. The study creates the prerequisites for building a full-featured system of visualization of ontologies, to be part of the decision support system for a particular subject area.

Keywords 1

visualization, ontology, web system, ontograph, subject area

1. Introduction

Informatization and information technologies have covered almost all spheres of life in modern society. It is impossible to name at least one branch of human activity, where their use would not be in demand. Ontologies are one of the best tools for a formal presentation of knowledge about a particular subject area. Most often, the concept of ontology is expressed by defining the basic objects (individuals, attributes, concepts) and the relationship between them, but the analysis of complex concepts may require considerable effort and time [1, 2].

Visualization of ontologies is an important aspect of their practical use. Qualitative visualization provides experts with an understanding of ontology or its fragments (ontology comprehension) [3, 4] in solving various problems of knowledge engineering. The effectiveness of an approach to the visualization of the ontology significantly depends on the problem. One of the problems that requires the presentation of the ontology in the form of some visual image, is its own understanding [5]. As a result of which, the user must understand the overall structure of the ontology, ignoring insignificant details. This task usually occurs when choosing it for reuse. In this case, the process of comprehension

COLINS-2021: 5th International Conference on Computational Linguistics and Intelligent Systems, April 22–23, 2021, Kharkiv, Ukraine
EMAIL: Taras.M.Basyuk@lpnu.ua (T. Basyuk); Andrii.S.Vasyliuk@lpnu.ua (A. Vasyliuk)
ORCID: 0000-0003-0813-0785 (T. Basyuk); 0000-0002-3666-7232 (A. Vasyliuk)

allows the user to decide whether, in general, this ontology or its fragment is suitable for use in solving the problem [6].

In the context of formal ontologies visualization, the problem of logical conclusions results visualization is also an important task. Its essence is to create a visual representation that can illustrate the derivation of logical statements, to explain the results of derivation. Thanks to this idea, the developer can better understand the ontology, as well as quickly find and correct problematic axioms that lead to semantic conflicts.

2. Analysis of recent researches and publications

The study showed that heterogeneous knowledge in intelligent information systems is poorly formalized, which complicates the use of information systems for the complex objects of operational management and processes and their evaluation in terms of staff. At the same time, the use of visual images contributes to a sharp increase in perception and allows you to quickly and effectively convey your thoughts and ideas. Numerous studies [7,8] confirm that:

- 90% of information a person perceives through sight;
- 70% of sensory receptors are in the eyes;
- about 50% of neurons in the human brain are involved in the processing of visual information.

The authors [9-11], based on new scientific, technical and philosophical-methodological approaches, formulate a scientific hypothesis, which states that: information-symbolic systems (whatever perfection they reach) are not sufficient in the context of realizing the possibilities of modern artificial intelligence systems; more heuristically powerful and information-intensive are information-image structures. Given the same, visualization (information and imaging systems) is associated with the future prospects for the development of computer technology, in particular, the future of artificial intelligence. The analysis showed that there are two directions in the development of data visualization systems. The first is the development of universal visualization tools. The second - specialization in all areas, including the creation of special graphics stations with the implementation of graphical algorithmic and software. If we consider the first approach [12-15], the existing methods of visualization are universal for different subject areas (visualization of modeling results in physics, chemistry and other applied areas). Typically, the application software systems created on their basis are quite complex and cumbersome and do not contain tools that can fully satisfy the main aspects of the data structure visualization. As for the second approach, it focuses on the narrow specialization of methods, in which there are certain achievements that are embodied in different software:

- Ontolingua. The system was developed at Stanford University's Knowledge Systems Laboratory (KSL) and became the first ontology engineering tool. It consists of a server and a language for presenting knowledge. The Ontolingua server is organized as a set of ontologies related to Web applications that are built on the Ontolingua knowledge representation system. The ontology editor is the core of the system and is built using HTML forms. In addition to the ontology editor, the Ontolingua server includes the Webster network application (concept definition), the OKBC server (access to OKBC ontologies), and Chimaera (analysis, aggregation, ontology integration). All applications, except the OKBC server, are implemented on the basis of HTML forms, which is implemented in Lisp [16].
- Protégé. A local, freely distributed Java application developed by Stanford University Medical Informatics. The program is designed to build (create, edit and view) ontologies of the application area, with the main purpose - to help software developers to create and maintain explicit models of the subject area and include these models directly in the program code. Protégé includes an ontology editor that allows you to design ontologies by deploying a hierarchical structure of abstract or specific classes and slots. Based on the established ontology, the system can generate forms of knowledge for the introduction of instances of classes and subclasses [17].
- OntoEdit. The editor was developed at the AIFB (Institute of Applied Informatics and Formal Description Methods) of the Karlsruhe University, which checks, reviews, encodes and modifies ontologies. Currently, OntoEdit supports the following presentation languages: FLogic, OIL, RDFS extension. The editor has created its own OXML (OntoEdit's XML-based

Ontology representation Language). Among the advantages of the system are ease of use, development of an ontology using a wizard, development of axioms, extended documentation [18].

- OilEd. An autonomous graphic ontology editor developed at the University of Manchester as part of the European IST On-To-Knowledge project. The tool is built in the OIL language (now adapted for OWL), which combines the frame structure and expressiveness of Description Logics with reasoning services. This allowed to provide a clear and intuitive style of the user interface and the benefits of reasoning support (detection of logically contradictory classes and hidden subclass relations) [19].
- WebOnto. The editor is designed for the Tadzebao system, an ontology research tool designed to support collaborative viewing, creation and editing of ontologies. The main purpose of the system is the ability to build large ontologies, which are modeled using the language OCML (Operational Conceptual Modeling Language). In WebOnto, the user can create structures, including classes with multiple inheritance and OCML integrity control. The tool has a number of useful features: saving structural diagrams, separate view of relationships, classes, rules, etc. [20].
- OntoSaurus is a Web browser for LOOM knowledge bases, consisting of two main modules: an ontology server and a Web browser for editing and viewing LOOM ontologies using HTML forms. It provides limited editing tools, but you need to understand the LOOM language to build complex ontologies. OntoSaurus implements all the features of the LOOM language, namely: provides automatic compatibility control, deductive reasoning support [21].
- ODE (Ontological Design Environment) - involves interaction with the user at the conceptual level. The reason for creating ODE was that it is easier for people to formulate ontologies at the conceptual level, given that ODE provides users with a set of tables to fill (concepts, attributes, relationships) and automatically generates code for them in LOOM, Ontolingua and FLogic. ODE is part of the methodology of the full life cycle of building an ontology according to Methontology. The tool was further developed as a WebODE system, which integrates all ODE services into one architecture, stores its ontologies in a relational database, provides additional services (output machine, axiom construction, ontology collection, directory generation) [22].
- KADS22 is a tool to support the design of knowledge models according to the CommonKADS methodology. CommonKADS models are defined in CML (Conceptual Modeling Language) and support the following functionality: parsing CML files, printing, hypertext viewing, search, glossary generation and HTML [23].
- The i.com tool is a tool to support the conceptual stage of an integrated information systems project through the use of an Extended Entity-Relationship Model (EER). The i.com tool is fully integrated with the DL-based reasoning server and is designed for intelligent conceptual modeling [24].
- OntoSphere. The system visualizes the ontology by type of node-connection. The OntoSphere3D user interface is fairly simple, focused on mouse control, and supports scene visualization by rotating, panning, and zooming. The elements of the ontology are represented as follows: concepts are displayed in the form of spheres, instances are represented in the form of cubes, literals are displayed in the form of cylinders, and the relationship between entities using directional lines with a conical arrow. A feature of the system is the support of SPARQL queries [25].
- Harmony Information Landscape. It was designed to visualize hypertext and organize the nodes presented as three-dimensional objects directly on the plane. Objects differ in color and size depending on the content. In addition, connections and types of relationships can be visualized between hypertext documents and ontologies [26].

As can be seen from the analysis, this area is extremely important but not fully explored, given that the important and urgent task is to create a system of visualization of ontologies of the subject area, which would allow users to display this structure and adjust playback modes depending on end user requirements.

3. The main objectives of the study and their significance

The aim of the research is to create the subject area ontology visualization system on the example of ontology models. The study will provide tools for dynamic display of different types of ontographs in accordance with the formed criteria of visualization and reproduction of their properties for further use in information systems for operational management of objects and the end user.

To achieve this goal it is necessary to solve the following main tasks: to analyze existing research and systems used in the field of ontology visualization; identify the main tasks that arise, build models of the visualization process and implement the proposed approach using modern software.

The results of the study solve the current scientific and practical problem of creating software for visualization of the ontology of the subject area, which would provide the simplest and most adaptive mechanism for displaying these ontologies of any structure and format.

4. Major research results

4.1. Choice of ontology representation structure

Traditionally, within computer display, ontology is used as a mean of computer processing knowledge. However, with the growth of formalized knowledge accumulated on media, there is a problem of their effective transfer to other systems and user-experts. In this case, the cognitive properties of this structure come to the fore, which directly affect the simplicity and accuracy of their interpretation in order to obtain a concept [27].

Given the analysis, two possible ways of presenting ontologies were identified: in the form of frames and the use of the Simple Knowledge Organization System (SKOS). If we consider the frame representation, it can be represented as a visualized fragment of the ontology, which provides the means to display the structure for further transmission to the stakeholder. In this case, the frame contains two components - internal content (corresponds to the ontological context) and visual representation (determines the features of the display to the end user) [28,29].

The visual representation itself, according to the founder of the theory of frames work, M. Minsky [30] suggests that the processes of human thinking are based on numerous data structures stored in human memory - frames through which a person identifies visual images (frames of visual perception); understands words (semantic frames); judgments, actions (frames-scenarios); messages (story frames). In formal notation, a frame can be represented as a vulture structure, where the main situation denotes an object, and the subordinate vertices are elements of this object that can be observed from certain positions. It is believed that a person's long-term memory stores a large set of different frames that are updated during the presentation of new scenes. In order to recognize scenes in a person's memory, the frame that best corresponds to the hypothesis of the perceived object must be actualized. In this way, the object is identified in the human mind. In that case, if the required frame cannot be found, the best of the found frames are «adapted» to the real scene, and it is memorized for further use. Conceptually, according to Guarino [31], a frame consists of a set of units that are organized around a certain concept (action, process) Fig.1.

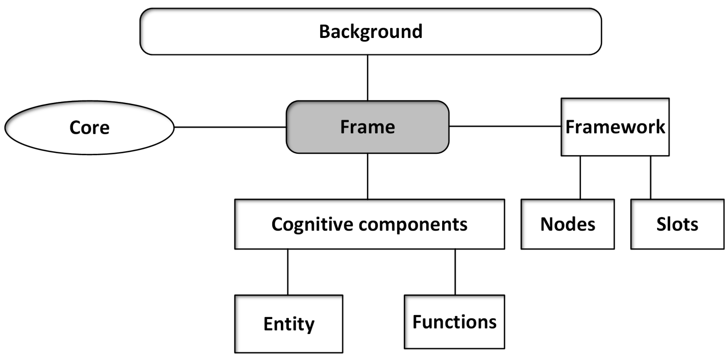

Figure1: Frame structure (by R. Gurino) [8]

The basic components of the frame are cognitive components that form an idea of typical events. In this case, any frame consists of filled slots (terminals), the similarity between which is necessary to establish links between the information contained in them. Since there can be many slots in a single frame, their diversity must be hierarchically organized. As an example in Fig.2. shows the taxonomy of the frame for the Network router concept.

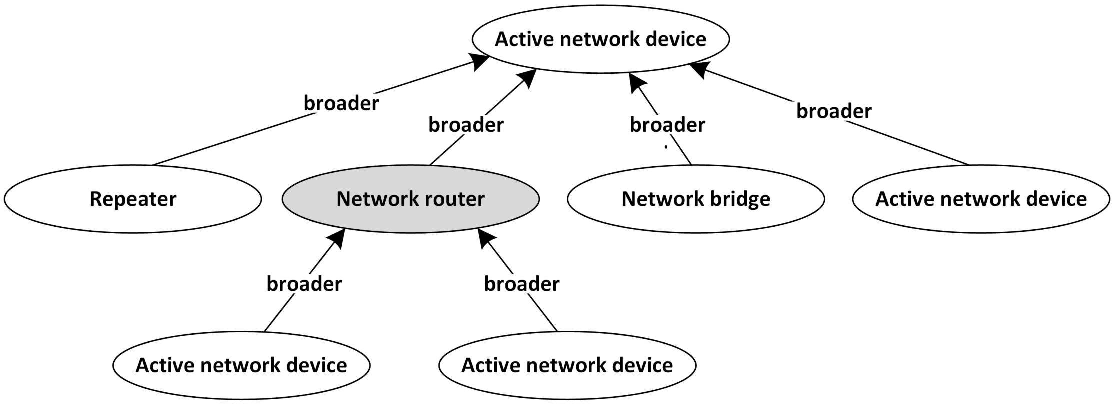

Figure 2: Taxonomy frame for concept "Network router"

The use of this data structure involves a high systematization of knowledge by increasing the number of structural relationships between vertical and horizontal elements.

The SKOS system is a model of knowledge organization developed by a consortium of the World Wide Web, the main purpose of which is to use in the Semantic Web to facilitate the interaction of information resources through standardization of thesauri, classifications, taxonomies [32]. Unlike previous standards, this model is designed to make it as easy as possible to bring other systems of knowledge organization to it. Examples are the subject index of the Library of Congress and the multilingual thesaurus maintained by the European Union Publishing Office – EUROVOC. SKOS identifies several semantic relations that can link thesaurus concepts, such as skos: broader, skos: narrower, and skos: related. In addition, SKOS supports the description of individual relationships, for example, skos: narrower is a more general relationship: skos: narrowerTransitive, which in turn refers to skos: semanticRelation. An example of SKOS-ontology for the concept of «Network router» is shown in Fig.3.

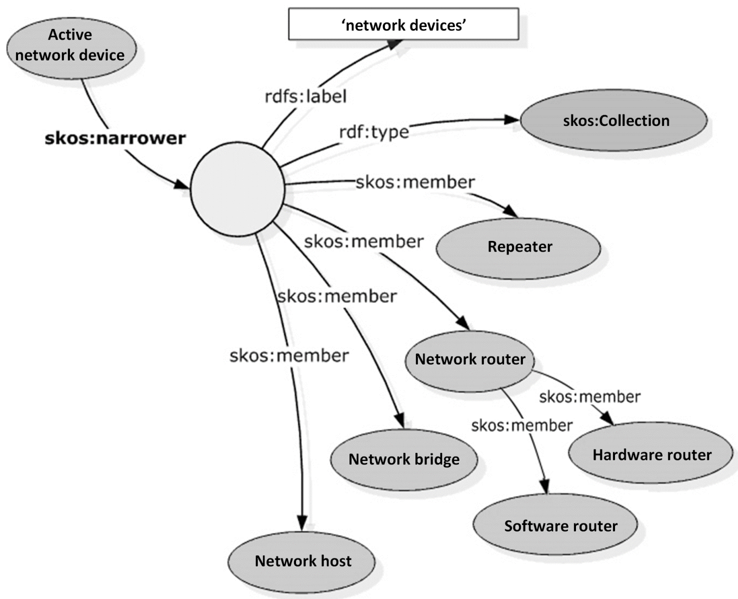

Figure 3: Example of SKOS ontology

To compare dictionaries, SKOS has developed semantic relationships that express different degrees of closeness of concepts: skos: exactMatch, skos: narrowMatch, skos: broadMatch, skos: closeMatch, etc.

4.2. Stages of ontology visualization

As it was shown, the subject area ontology visualization is a difficult stage in the design process, given that its display is proposed to be carried out in fragments. Namely, first to visualize a part of the ontological mapping (fragment) with the subsequent connection and use of the ontology of the categorical level. In this case, the process of creating an ontology of the subject area is proposed to perform as a sequence of the following stages:

- formation of a list of concepts that are part of the ontological structure of the subject area;
- ordering the list of concepts;
- creation of a glossary;
- display of arcs between created concepts.

Usually, in the process of the first stage, it is proposed to use information from explanatory dictionaries that reflect aspects of the functioning of the subject area. After that, the formed list is specified by the knowledge engineer, on the basis of professional (expert) data in order to determine the final list of concepts that will be used in the process of the subject area ontology visualization. In this case, firstly it is necessary to determine the technical aspects, which include the features of the reflection of concepts such as part of speech, plurality, etc.

In the second stage, the knowledge engineer, based on his own experience, carries out the process of organizing the list of concepts depending on their «subordination». The definition of «subordination» can be realized by one of the approaches: «top-down», «bottom-up» or combined [33]. The top-down approach is a conventional approach that seems to decompose the system from a high-level to a low-level specification, and an ontological model is built in this direction. In turn, the bottom-up approach works in reverse order, namely, when the ontograph formation is carried out from the lower levels, reaching a certain level of abstraction - the upper. The hybrid approach focuses on the selection of «basic» concepts that are most often used and then on their basis is a further generalization (formation of the upper level) and detailing (formation of the lower level). The analysis of the existing ontographs allowed to form the basic requirements to which the formed ontological images should correspond:

- classes of the same level should always be located on the same «line of visualization»;
- if a certain class contains more than eight subclasses, it is necessary to add intermediate subclasses in order to «reduce the load» of the final ontograph, in the case of a single subclass, an additional study of the subject area should be conducted;
- the location of the classes should be such that the possible intersections of the ontograph arcs are minimized.

Creating a glossary involves the formation of a list of definitions for terms that form the concept of the research subject area. Usually, its formation begins with the definition of basic terms and is clarified and supplemented at this stage. The peculiarity of this stage is the creation of formalized descriptions that contain essential features of certain concepts that affect the accuracy of the final ontology.

Then, on the basis of constructed and reflected classes-concepts, the visualization of the conceptual model of the subject area is carried out in the form of an ontograph which is offered to be visualized in the created software.

As an example, consider the process of forming a fragment of the ontology of the subject area “information security”, which reflects the process of describing the characteristics of a network computer attack. The analysis of the subject area allowed to choose the following concepts: “Illegal Activity”, “Social Engineering”, “Computer Network Attack”, “Local Penetration”, “Pretexting”, “Phishing”, “Protocol”, “Quiproquo”, “Tailgating”, “Travel Apple”, “SMTP”, “FTP”, “HTTP”, “HTTPS”, “HTTP Message”, “HTTP Request”, “HTTP Response”, “Request Line”, “Request Header”, “Status”, “Header”, “Content”, “Method”, “Trace”, “Delete”, “Referrer”, “Put”, “Get”, “Post”, “Version”, “URL”, “Port No”, “Domain Name”, ”Query String”, “Protocol Name”. Let's organize the list of terms, having formed the corresponding tuples:

- Illegal Activity = < Social Engineering, Computer Network Attack, Local Penetration>

- Social Engineering = <Pretexting, Phishing>
- Local Penetration =<Quiproquo, Tailgating, Travel Apple>
- Computer Network Attack=<Protocol>
- Protocol=<SMTP, FTP, HTTP, HTTPS>
- HTTP=<HTTP Message>
- HTTP Message=<HTTP Request, HTTP Response>
- HTTP Response=<Status, Header, Content>
- HTTP Request=<Request Line, Request Header>
- Request Line=<Method, Version, URL>
- Request Header=< Referrer >
- Method = <Trace, Delete, Put, Get, Post>
- URL = <Port No, Domain Name, Query String, Protocol Name>

On the basis of the ordered list of tuples it is necessary to form a matrix of contiguity on exits and on the basis of the developed approach to carry out division of the formed concepts on tiers with use of a tier-parallel form of display [34].

4.3. Research of the software system usage options

Further research was aimed at automating this process, given that a decision was made on the software implementation of the described approach. The analysis of the existing visualization systems allowed to form the requirements to the designed system, which will provide opportunities for the correct display of ontologies in accordance with the proposed criteria. The requirements are described using a mechanism of functions and delimitation of work for users and administrators. After defining the requirements, the system was designed using an object-oriented approach. A set of UML diagrams is constructed, in particular the diagram of use cases is presented in Fig.4.

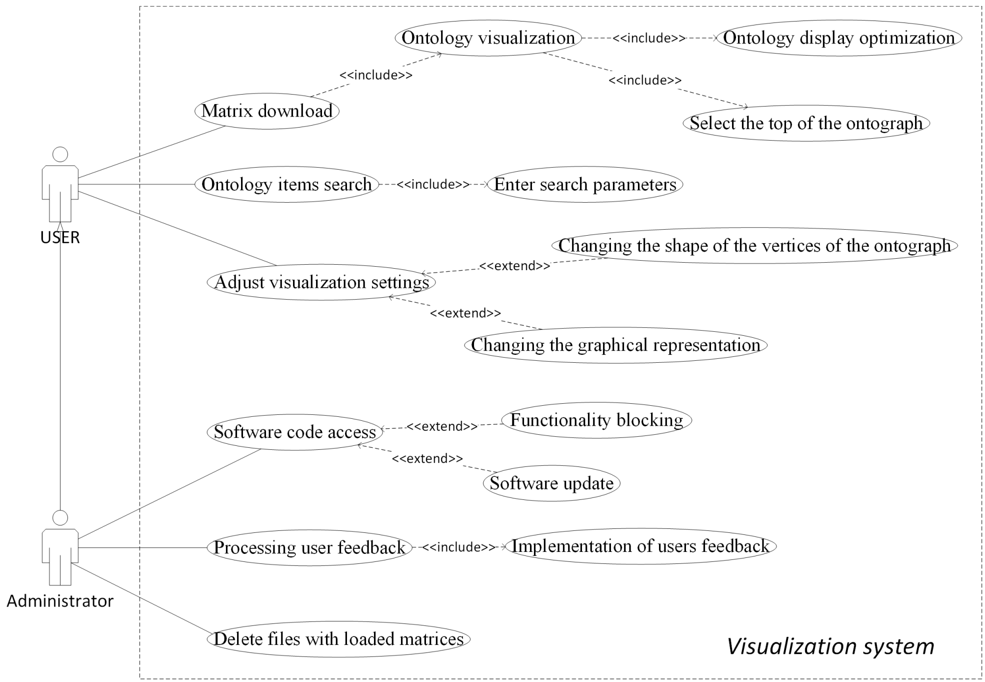

Figure 4: Diagram of options for using the web system for visualization of ontologies.

The main functions of users are:

- downloading matrix descriptions and their visualization;
- search by visualized ontology;
- adjustment of visualization parameters, in particular display: distances between visualized classes, sizes of elements used for display;

- obtaining information about ontology metadata (title, author, etc.) and statistical data (number of nodes, visualization arcs, etc.).

Users-administrators will be able to:

- access the software code, which will allow you to quickly eliminate errors in the work or temporarily block certain functionality;
- delete files with visualized ontologies;
- implement feedback from users.

4.4. Choice of architecture and means of implementation

A studied subject area feature is the presence of different types of calculations and operations with data that need to be visualized, which requires simultaneous operation with the multiple users system. Given that, it was decided to implement it in the form of a multifunctional website, which on the one hand will provide unimpeded access to the resource, and on the other - will facilitate its use on different platforms [35,36].

The design of the software architecture was carried out using an architectural style, which is based on defining the basic elements of the system structure - components of subsystems, objects and relationships between them using three-tiered client-server architecture. This type of software architecture peculiarity is that it consists of three main parts:

- the client component of the program, which usually implements the graphical user interface and transmits requests to the application server using the types of data that it can process;
- a program execution server that implements business logic and sends requests to a "third-tier" server, such as an external API or a remote database server, to retrieve additional data from it;
- database server - manages the database and is responsible for the integrity and storage of data, as well as provides input-output operations when the client accesses information.

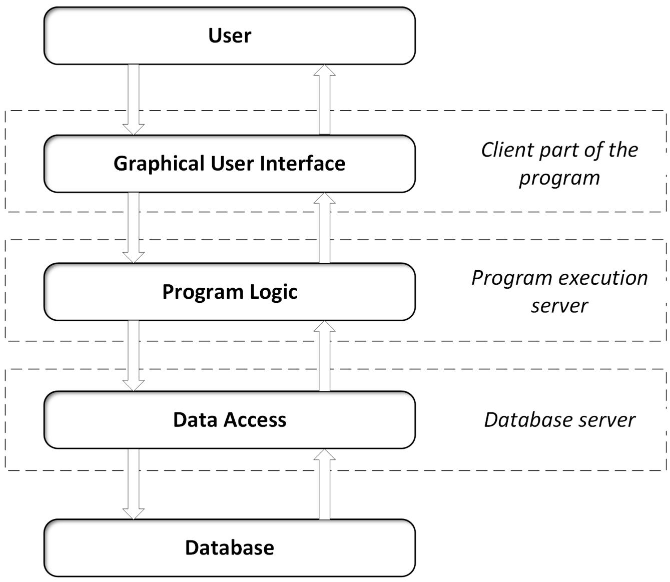

Figure 5: The structure of the three-tier client-server architecture

When developing a visualization system, the use of a three-tier client-server will allow you to use a number of its advantages, in particular [37]:

- reducing the load on the hosting server network, as the links will not exchange significant amounts of information;
- the possibility of independent implementation of change architecture at each level;
- ensuring the scalability of the system and the possibility of easy updating.

Comparing the structure of modern visualization systems [38-40], in particular the main components and purpose of their subsystems, with a three-tier client-server architecture, we can see that they have much in common (purpose, relationships, etc.), which means that its use will be optimal for both program developers and users.

The next stage of the study was the choice of implementing the software system means. The analysis of both the required functionality and the interactivity of the application contributed to the choice as a programming language - JavaScript, which uses the user's web browser as the execution environment. The main technologies that were used:

1. JQuery - a library, the use of which allowed to create a dynamic web page that supports animation, event handling and the use of AJAX-applications;
2. HTML5 (HyperText Markup Language, version 5) - this language has improved (compared to previous versions) multimedia support, provides adaptability of the Internet resource while maintaining backward compatibility and ease of analysis for software spiders;
3. CSS 3 (Cascading Style Sheets, version 3) - the use of the third version of cascading tables provided the means to quickly describe the appearance of a web page created using HTML;
4. Bootstrap 3 - the use of styles in this library has created a product that is adaptable to the screen size and is supported by all modern web browsers.

The Webpack library was used to build individual JavaScript modules, which combines all the software modules of the system and creates static resources that represent them. In Fig. 6 presents a diagram of the Webpack.

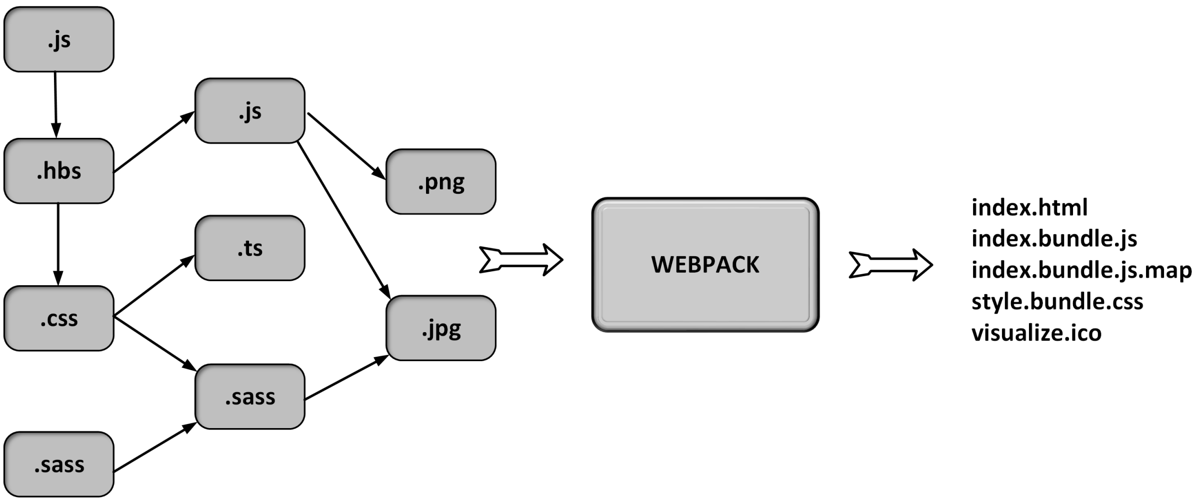

Figure 6: Scheme of the webpack library

In addition to Webpack, we used the Grunt task manager, written in JavaScript and designed to automate a wide range of tasks such as source code compilation, binary code collection, and system deployment. The combination of «grunt-webpack» and «webpack-dev-server» allowed implementing live-server technology, which provides automatic start of the web server after each code update. The main module of the developed system is a visualization unit, based on the D3.js library. The name D3 stands for Data-Driven Documents, which focuses on the use of JavaScript, SVG and CSS, which provide flexible tools for creating ontological structures with animation elements and interoperability. Compared to pixel graphics, SVG has many advantages, namely XML-based, which makes the generated code easy to read and helps reduce the size of the generated images. These features have led to the fact that today, this framework is the most popular for graphical data processing and creating various kinds of graphical interpretations. With the help of this framework, a frame model is built in the created prototype, which adapts to the user's requirements.

4.5. Development of an ontology visualization system prototype

The described architecture formed the software implementation basis of the subject area ontology visualization system prototype, which was implemented in JavaScript. Fig.7 shows the main page of the website which presents a visualized ontograph of the example that was shown in paragraph 4.2.

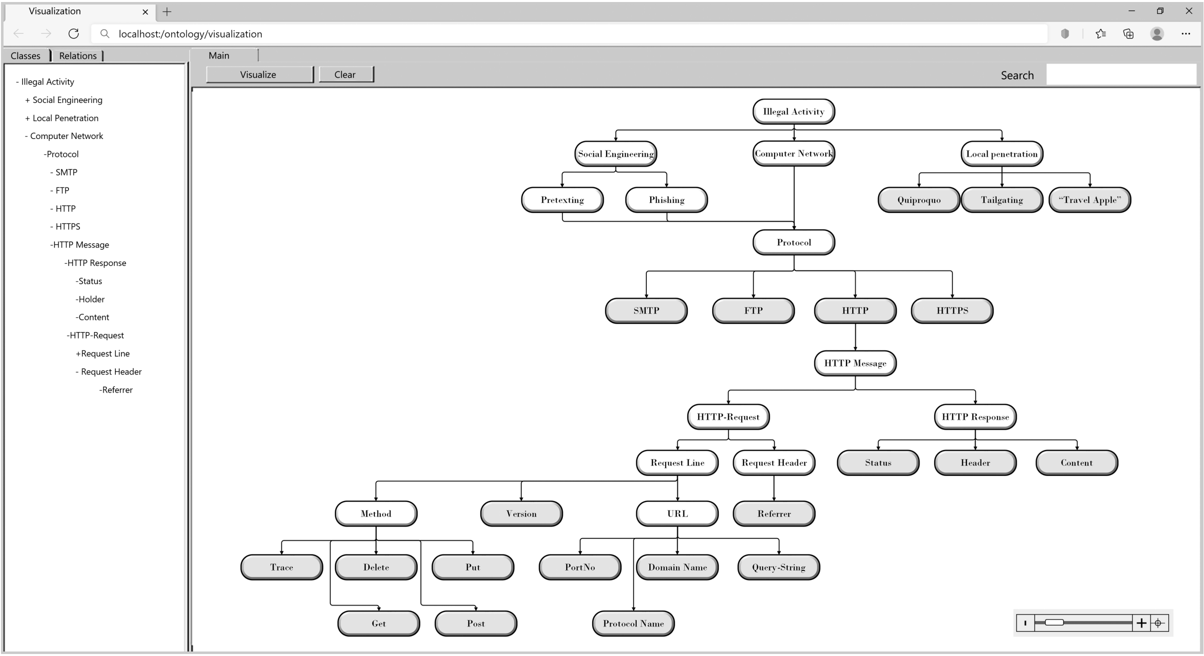

Figure 7: Home page of a web system for visualizing ontologies

As can be seen from the figure, the main window of the system consists of two parts: the left - contains a hierarchical tree that displays the classes of the ontograph and the right - the main window in which the visualized ontograph is displayed. The formed ontograph is presented in a tier-parallel form in which there are no relationships within the tier. This form of presentation fully meets the established criteria of reproduction and is easy to evaluate and further study. A separate function of the system is to search for the ontograph node by key term, and the search result is displayed in an enlarged form with «toning». An example of a search for «HTTP-Request» is shown in Fig.8:

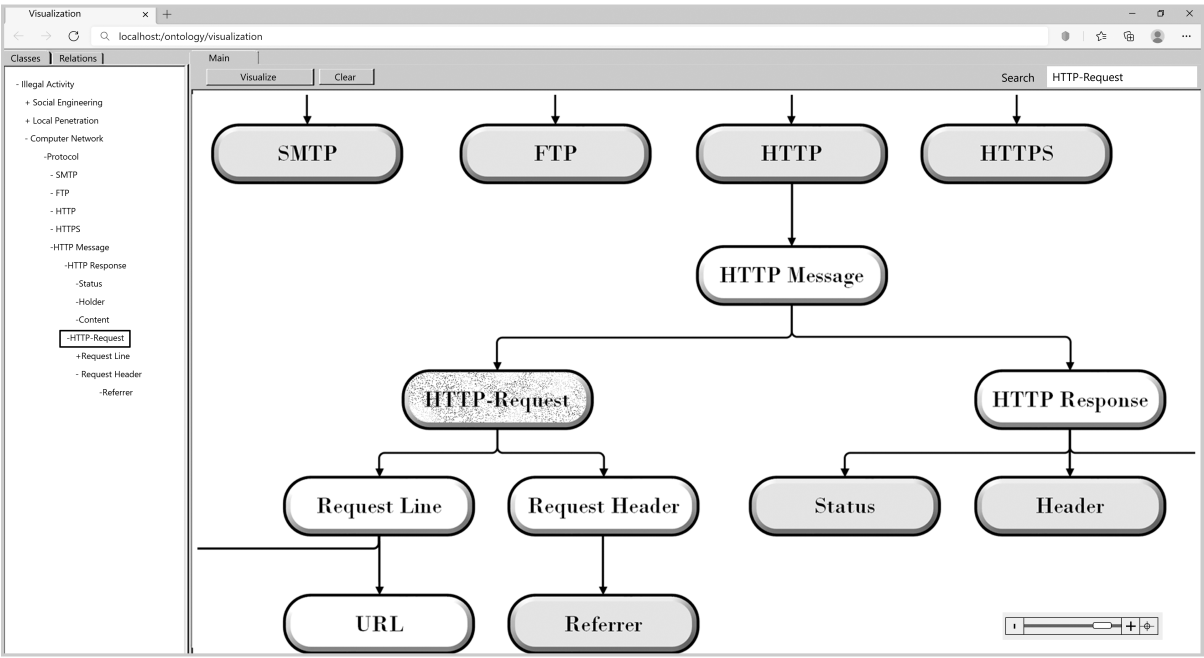

Figure 8: The result of finding a single node in the ontology

Currently, the system operates in the form of a demonstration prototype with limited functionality, which will be expanded in accordance with the described functionality.

5. Conclusion

As a result of the conducted research the existing technologies and systems of ontologies display on the given features which motivate the creation of a new web-oriented system of visualization are analyzed. The analysis of possible means of ontology representation in the form of frame models and use of SKOS is carried out. The main stages of building an ontology of the subject area are described and an example of visualization of the concept of «network computer attack» is given. Further research was based on the creation of a project using an object-oriented approach and the use of UML language tools. The practical implementation of the system is based on the use of three-tiered client-server architecture using the Javascript language. The result of the work is a prototype of an ontology visualization system that can be used to display a given structure and describes a specific subject area.

Further research will be aimed at creating related software modules, their verification and coordination of operation.

6. References

[1] Y. Li, C. Stroe, I. Cruz, Interactive visualization of large ontology matching results, in: International Workshop on Visualizations and User Interfaces for Ontologies and Linked Data Colocated with ISWC 2015, Bethlehem, Pennsylvania, USA, 2015, pp. 37-48.

[2] T. Basyuk, V. Lytvyn, D. Dosyn, Ontological engineering, Lviv Polytechnic Publishing House, Lviv, 2017.

[3] Motta E. A Novel Approach to Visualizing and Navigating Ontologies. In: Aroyo L. et al. (eds) The Semantic Web – ISWC 2011. ISWC 2011. Lecture Notes in Computer Science, vol 7031. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-25073-6_30.

[4] A. Sears, J. Jaco, Human-Computer Interaction. 2nd.ed. Tailor&Francis Group, 2009.

[5] N. Guarino, D. Oberle, S. Staab, What is an ontology? In: Handbook on Ontologies, 2009.

[6] M. Dudáš, S. Lohmann, V. Svátek, D. Pavlov, Ontology visualization methods and tools: A survey of the state of the art, The Knowledge Engineering Review, 33, Published online by Cambridge University Press, 2018. doi:10.1017/S0269888918000073.

[7] A. Dix, Human-Computer Interaction, Springer US, New York, USA, 2009.

[8] B. Bederson, The Craft of Information Visualization: Readings and Reflections, Maryland, USA, 2003.

[9] H. Sharp, Interaction Design: Beyond Human-Computer Interaction, 5th ed. Wiley, 2019.

[10] E. Tufte, The Visual Display of Quantitative Information, 2nd. ed. Graphics Press, Connecticut, 2001.

[11] W. Colin, Information Visualization, 3rd. ed. Perception for Design (Interactive Technologies), New rider, New York, USA, 2012.

[12] A. Katifori, C. Halatsis, G. Lepouras, C. Vassilakis, E. Giannopoulou, Ontology visualization methods - a survey, ACM Computing Surveys, vol. 39, No. 4, New York, United States, 2007. doi:10.1145/1287620.1287621.

[13] S. Falconer, C. Callendar, M.-A. Storey, FLEXVIZ: visualizing biomedical ontologies on the web. In: International Conference on Biomedical Ontology, Software Demonstration, Buffalo, 2009, p. 46-27.

[14] A. Vasyliuk, T. Basyuk, V. Lytvyn, Specialized interactive methods forusing data on radar application models, Proceedings of the 2nd International workshop on modern machinelearning technologies and data science (MoMLeT+DS 2020). Volume 1: Mainconference, Lviv-Shatsk, Ukraine, June 2-3, 2020, Vol. 2631, pp. 1–11.

[15] T. Basyuk, V. Lytvyn, Languages for describing ontologies, Lviv Polytechnic Publishing House, Lviv, 2020.

[16] A. Farquhar, R. Fikes, J. Rice, The Ontolingua server: A tool for collaborative ontology construction, International Journal of Human-Computer Studies, Volume 46, Issue 6, 1997, pp. 707-727. doi: 10.1006/ijhc.1996.0121.

[17] M. Musen, Domain Ontologies in Software Engineering: Use of Protege with the EON Architecture, Methods of Information in Medicine, 1998, pp.540-550. doi: 10.1055/s-0038-1634543.
[18] Y. Sure, M. Erdmann, J. Angele, S. Staab, R. Studer, D. Wenke, OntoEdit: Collaborative ontology development for the Semantic Web. in: Proceedings of the International Semantic Web Conference (ISWC 2002), Sardinia, Italia, 2002, pp.134-149.
[19] S. Bechhofer, I. Horrocks, C. Goble, R. Stevens, OilEd: A Reason-able Ontology Editor for the Semantic Web. In: Baader F., Brewka G., Eiter T. (eds) KI 2001: Advances in Artificial Intelligence. KI 2001. Lecture Notes in Computer Science, vol 2174. Springer, Berlin, Heidelberg. doi:10.1007/3-540-45422-5_28
[20] J.Domingue, WebOnto: Discussing, Browsing, and Editing Ontologies on the Web, Proceedings of the Eleventh Workshop on Knowledge Acquisition, Modeling and Management, KAW'08, Banff, Canada, 2008, pp.126-134.
[21] R. MacGregor, Inside the LOOM classifier, ACM SIGART BulletinVol. 2, No. 3, 2001. pp.70-76. doi: 10.1145/122296.122309
[22] J. Arpirez, O. Corcho, M. Fernandez-Lopez, A. Gomez-Perez, WebODE: a scalable workbench for ontological engineering. in. Proceedings of the First International Conference Knowledge Capture (KCAP2011). Victoria, Canada, 2011. pp. 96-106.
[23] G. Schreiber, H. Akkermans, A. Anjewierden, R. Hoog, N. Shadbolt, W. Van de Velde, B. Wielinga, The CommonKADS Methodology, Knowledge engineering and management, MIT press, Massachusetts, 2009. pp.45-88.
[24] E. Franconi, G. Ng, The: i.com tool for intelligent conceptual modeling, Proceedings of the 7th Intl. Workshop on Knowledge Representation meets Databases, KRDB'00, Berlin, Germany, 2000, pp.167-174.
[25] A. Bosca, D. Bonino, P. Pellegrino, OntoSphere: more than a 3D ontology visualization tool, Proceedings of the 2nd Italian Semantic Web Workshop University of Trento, Trento, Italy, 14-16 December 2005, pp.1-16.
[26] N. Gershon, J. R. Brown, Computer Graphics and Visualization in the Global Information Infrastructure, in: IEEE Computer Graphics and Applications, vol. 16, no. 2, 1996, pp. 60-82. doi: 10.1109/MCG.1996.486681.
[27] Th. Gruber, What is an Ontology, 1992. URL: http://www-ksl.stanford.edu/kst/what-is-an-ontology.html.
[28] V. Lytvyn, V. Vysotska, V. Mykhailyshyn, A. Rzheuskyi, S. Semianchuk, System Development for Video Stream Data Analyzing. In: Lytvynenko V., Babichev S., Wójcik W., Vynokurova O., Vyshemyrskaya S., Radetskaya S. (eds) Lecture Notes in Computational Intelligence and Decision Making. ISDMCI 2019. Advances in Intelligent Systems and Computing, vol 1020. Springer, Cham, 2020.
[29] M. Lanzenberger, J. Sampson, M. Rester, Ontology Visualization: Tools and Tecniques for Visual representation of Semi-Structrured Meta-Data. Journal Of Universal Computer Science 16(7), 2010. pp.1036-1054. doi: 10.3217/jucs-016-07-1036.
[30] M. Minsky, A Framework for Representing Knowledge, MIT-AI Laboratory, Memo 306, 1974.
[31] N. Guarino, Understanding, Building, and Using Ontologies, 2002. URL. http://ksi.cpsc.ucalgary.ca/KAW/KAW06/guarino/guarino.html.
[32] SKOS Core Guide, 2005. URL. https://www.w3.org/TR/2005/WD-swbp-skos-core-guide-20051102/.
[33] M. Burch, S. Lohmann: Visualizing the evolution of ontologies: a dynamic graph perspective, in: International Workshop on Visualizations and User Interfaces for Ontologies and Linked Data Colocated with ISWC 2015, Bethlehem, Pennsylvania, USA, 2015, pp. 69-76.
[34] R.Dunets, T.Basyuk The structure of the program for converting graphs into a tiered-parallel form, Computer printing technology, Ukrainian Academy of Printing, Lviv, Ukraine, Vol.7, 2002. pp.97 – 102.
[35] T. Basyuk, Popularization Of website and without anchor promotion, in: Proceeding of the XIth International Scientific and Technical Conference Computer Sciences and Information Technologies (CSIT), Lviv, 2016, pp. 193-195, doi: 10.1109/STC-CSIT.2016.7589904.

[36] T. Basyuk, The Popularization Problem of Websites and Analysis of Competitors". in: Shakhovska N., Stepashko V. (eds) – Advances in Intelligent Systems and Computing II. CSIT 2017. Advances in Intelligent Systems and Computing, vol 689. Springer, Cham pp. 54-65, doi:10.1007/978-3-319-70581-1_4.
[37] T. Basyuk, A. Vasyliuk, V. Lytvyn, Mathematical Model of Semantic Search and Search Optimization, Proceedings of the 3rd International Conference on Computational Linguistics and Intelligent Systems (COLINS-2019). Volume I: Main Conference, Kharkiv, Ukraine, April 18-19, 2019, Vol-2362: pp.96-105.
[38] S. Narayanan, S. Mcllraith, Analysis and simulation of Web services, Computer Networks: The International Journal of Computer and Telecommunications Networking. Volume 42 , Issue 15 (August 2013). Special issue: The Semantic Web: an evolution for a revolution. - New York, NY, USA Publisher Elsevier North-Holland, Inc., 2013. - pp. 675-693.
[39] Z. Cui, M. Cox, D.Jones, An Environment for Managing Enterprise Domain Ontology, 2000. URL. http :/www.btexact.com/-proj ect/-ibsr/publications.html.
[40] N. Casellas, Ontology Engineering: Methodologies, Modelling Trends, and the Ontology of Professional Judicial Knowledge, Springer Science & Business Media, 2011.