https://doi.org/10.31891/csit-2024-3-2
UDC 004.4

Illia LUTSYK, Dmytro FEDASYUK
Lviv Polytechnic National University

ANALYSIS OF APPROACHES TO DESIGN ONTOLOGICAL MODELS OF AN ADAPTIVE SOFTWARE SYSTEM

The paper analyzes approaches to the design of ontological models of adaptive software systems. To evaluate the quality of the projected ontologies, specialized groups of metrics are used, which allow us to determine the shortcomings in the structure and functioning of the model. An analysis of the use of ontological models in the process of designing and developing adaptive software systems has been carried out. It was established that the ontological approach allows for a more effective presentation of the subject area, which improves the software configuration process. The use of ontological models allows flexible responses to changes in user requirements without the need to directly update the source code of the software.

The process of designing ontologies based on classical and abstract approaches is presented. It was determined that the use of a classical approach to the design of ontological models could complicate the adaptation process due to the need to change the structure and reconfiguration of the system. To solve this problem, an abstract approach is proposed that ensures the dynamism of the adaptation process, without the need for a complete reconfiguration of the system.

A set of specialized metrics is defined that allows for the analysis of the ontological model of the software system, the evaluation of its syntactic and semantic quality, compliance with rules and interoperability, and the possibility of expansion. A comparative analysis of abstract and classical approaches to the design of ontological models was carried out based on defined structural metrics, as well as metrics of the scheme and filling of the knowledge base. The obtained values of structural and hierarchical metrics confirm the quality and effectiveness of the proposed approaches. According to the defined metrics, the absence of cycles and entanglement in the ontological model indicates the correctness of the construction of the structure from the point of view of ergonomics and the possibility of further changing the structure and filling of the ontological model. The analyzed values of the schema metrics and the filling of the knowledge base demonstrate the better effectiveness of the abstract approach in the process of determining the software configuration, which provides faster and more correct processing of ontological SWRL rules during adaptation.

Keywords: ontology, adaptive software, adaptive system, ontology metrics, ontology evaluation

Ілля ЛУЦИК, Дмитро ФЕДАСЮК
Національний університет «Львівська політехніка»

АНАЛІЗ ПІДХОДІВ ПРОЄКТУВАННЯ ОНТОЛОГІЧНИХ МОДЕЛЕЙ АДАПТИВНОЇ ПРОГРАМНОЇ СИСТЕМИ

В роботі проаналізовані підходи до проєктування онтологічних моделей адаптивних програмних систем. Для оцінювання якості спроєктованих онтологій використано спеціалізовані групи метрик, що дозволяють визначити недоліки в структурі та функціонування моделі. Проведено аналіз використання онтологічних моделей у процесі проєктування та розробки адаптивних програмних систем. Встановлено, що онтологічний підхід дозволяє більш ефективно представити предметну область, що в свою чергу покращує процес конфігурування програмного забезпечення. Використання онтологічних моделей дозволяє гнучко реагувати на зміни вимог користувачів без необхідності безпосереднього оновлення початкового коду програмного забезпечення.

Представлено процес проєктування онтологій на основі класичного та абстрактного підходів. Визначено, що використання класичного підходу до проєктування онтологічних моделей може ускладнювати процес адаптації через необхідність зміни структури та реконфігурації системи. Для вирішення даної проблеми запропоновано абстрактний підхід який забезпечує динамічність процесу адаптації, без необхідності повної реконфігурації системи.

Визначено набір спеціалізованих метрик, які дозволяють проаналізувати онтологічну модель програмної системи, оцінюючи її синтаксичну і семантичну якість, відповідність правилам, інтероперабельність та можливість розширення. Проведено порівняльний аналіз абстрактного та класичного підходів до проєктування онтологічних моделей на основі визначених структурних метрик, а також метрик схеми та наповнення бази знань. Отримані значення структурно-ієрархічних метрик підтверджують якість та ефективність запропонованих підходів. Відповідно до визначених метрик, відсутність циклів та заплутаності у онтологічній моделі вказує на коректність побудови структури з точки зору ергономіки та можливість подальшої зміни структури та наповнення онтологічної моделі. Проаналізовані значення метрик схеми та наповнення бази знань демонструють кращу ефективність абстрактного підходу у процесі визначення конфігурації програмного забезпечення, який надає швидше та коректніше опрацювання онтологічних SWRL правил під час адаптації.

Ключові слова: онтологія, адаптивне програмне забезпечення, адаптивна система, метрика онтології, оцінка онтології

Introduction

Modern software systems are characterized by increased requirements for dynamic software configuration in accordance with changes in user requirements. One of the methods to solve this problem is the use of semantic networks and ontological models, which are characterized by universality and flexibility [1, 2]. The use of ontologies in the development of a software system, its interface, and functional components ensures the implementation of software adaptability without adjusting the source code based on the formation of a dynamically changing knowledge base [3].

The use of ontological models in the process of adapting the graphical interface is a common approach to creating a personalized user experience depending on the requirements [4]. Most modern systems offer various configuration options for both individual elements and the overall structure and layout of the graphical interface, which improves interaction with the software. In this case, in the process of contextual adaptation, the use of a formalized and standardized structure of ontological models allows us to ensure that the configuration process is performed regardless of the device [5].

Another important aspect of software adaptability is the ability to personalize functionality to the user's needs. The use of an ontological approach in the design process provides the possibility of a detailed representation of objects and connections of the subject area [6, 7]. In addition, it should be noted that this approach allows you to create smart systems that will be personalized to the needs of the user without the need to pre-install all available functional components. Thus, dynamically identifying recommended software modules using ontologies reduces the load on system resources, which in turn reduces hardware requirements.

While stating the positive aspects of using ontological models in the process of creating software systems, it should be noted that it is necessary to check and control the quality of the created model. Therefore, in the process of designing ontologies, it is important to track their compliance with a certain goal. For this, a certain set of criteria or metrics is used that form an assessment of the quality of the ontology [8]. The ontology quality assessment process is a complex task, which is based on determining the key properties of the ontological graph and the interaction between its vertices.

Existing software quality models focus on the study of the characteristics of the software code, as well as on the processes associated with its development, modification, support, and maintenance. At the same time, the presented quality assessments cannot be applied directly to ontological models, since they fundamentally differ from programs not only structurally or functionally, but also have significant differences in the processes of design, development, and support [9]. In such cases, unified quality methodologies should be used, based on existing criteria or metrics that are defined in ontology engineering in combination with quality criteria according to software engineering theory [10].

Therefore, taking into account the indicated problems and challenges in the creation of adaptive software systems, the purpose of this work is to reveal approaches to the design of ontological models and assess their quality based on a set of specialized metrics.

Description of approaches to designing ontological models of software systems

The process of designing an ontology depends on the implementation of components of a certain subject area. Therefore, an important task is to correctly define a set of concepts and relationships, which will allow the knowledge base to be qualitatively filled with instances and axioms. In addition, when designing an ontology, it is necessary to ensure that the resulting model meets the following requirements [11, 12]:

- the model must be defined in a standardized formal language (e.g. OWL, RDF, etc.);
- the concepts of the ontological model must be expressed through axioms;
- the model must conform to the previously formed requirements specification for the relevant subject area.

The process of designing and developing an ontological model classically involves several stages [12]:

1. Selecting a subject area;
2. Defining the core elements of a subject area and designing based on concepts and classes;
3. Defining properties for each generated class;
4. Creating instances of designed concepts.
5. Designing connections between ontology concepts and creating axioms.

Designing an ontological model of a software system in accordance with the specified sequence allows you to control its quality and characteristics. In addition, the model formed in this way will provide the ability to add semantic rules in the future. The presence of these rules will ensure the execution of the semantic decision-making mechanism to obtain new concepts or ontological connections.

In a formalized form, the ontological model of a software system can be represented as a combination of four sets:

$$O_{system} = \left\langle C_{system}, R_{system}, Prop_{system}, F_{system} \right\rangle, \tag{1}$$

where C_{system} – a set of entities of the subject area of an adaptive software system containing information about users and possible configurations;

R_{system} – set of relations between entities of the subject area;

$Prop_{system}$ – set of properties and attributes of subject area entities;

F_{system} – set of functions of interpretation of concepts of the ontological model.

It should be noted that the ontology design process according to the classical approach is effective only for systems with clearly defined objects of the subject area. In the case when the ontological model is used for adaptive software systems, its structure will change in accordance with the filling of the software with functional and graphical components. This will impair the efficiency of the adaptation process, because after adding new modules, it is necessary to add new concepts and relationships between elements, and for the software system - to do a complete reconfiguration and redeployment.

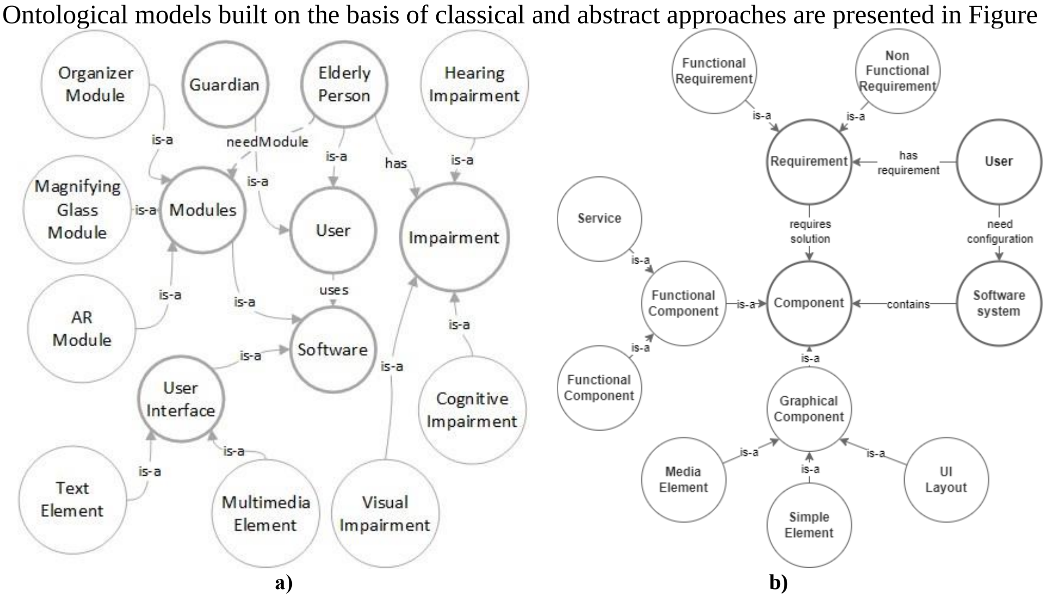
This problem, which arises when using the classical approach to designing an ontological model for an adaptive software system, can be illustrated using the example of software for helping people with cognitive impairments [13]. In accordance with the requirements, such software must respond to changes in the user needs and take into account various types of health disorders, in particular: impaired vision, and hearing, as well as cognitive changes.

So, according to the classical process of designing an ontology, first, the objects of the subject area are identified in the form of concepts of the corresponding model (Fig. 1.a): "Modules", "Impairment", "Software", "User Interface", etc. After the objects are created, appropriate connections are established, in particular: “uses” - denotes the connection between the user and the used software configuration; “needElement” and “needModule” – indicate the software module or graphical element required for installation or modification. The developed model met the basic requirements. However, in the process of further improvement of the structure of the ontological model in accordance with the adaptability requirements (adding new concepts of the ontological model: a software module or a user interface element), such changes led to discrepancies in the knowledge base scheme, which in turn created a problem of timely and correct migration of the ontological model versions. In addition, in the case of software adaptation, a new ontological rule had to be created for each new element, which also caused problems at the software support phase.

The solution to the problem of adapting the software system to the updated requirements based on the ontology is the use of an abstract approach to the design of the ontological model [14].

In accordance with the abstract approach, when creating concepts and connections of the ontological model between elements, abstractions of domain elements are created. Each abstraction can be expanded using entity instances, which in turn provides the ability to dynamically add new elements to the ontological model. In the case of software adaptation, three main abstractions are allocated that directly concern the adaptation process: "Component", "Functional component" and "Graphic component" (Fig. 1.b). In accordance with such a division, the main relationships and adaptation rules are established for the "Component" concept and apply to all subclasses and instances. In this way, the dynamism of the adaptation process is ensured, without the need for a complete reconfiguration of the system.

Ontological models built on the basis of classical and abstract approaches are presented in Figure 1.



a) **b)**

Fig. 1. Designed ontological models: a) classical approach; b) the proposed abstract approach

Comparing the models created according to both approaches, it should be noted, in particular, that the connections defined in the abstract approach provide a higher level of flexibility. In particular, the "needModule" and "needElement" relationships have been replaced by the "need configuration" relationship. The corresponding changes ensure the formation and preservation of different configurations of the software system in the knowledge base, depending on the presence of elements in the given version of the software. It also helps to reduce the number of ontological rules, since concept abstractions and updated relationships allow defining a single rule that will cover all possible configurations (using the has requirement - to identify user requirements and requires solution - to define required components).

To compare the correctness and efficiency of using ontological design for adaptive software systems according to the classical and abstract approaches, we used a quality assessment technique based on the use of specialized metrics.

Metrics for evaluating the quality of the ontological model of the software system

Considering the complexity of designing and developing ontological models of adaptive software systems, it is advisable to use specialized metrics to assess their quality, which, in general, can be divided into two categories: hierarchical and relational. The first group allows you to evaluate ontologies from the point of view of the hierarchy of subject area concepts and their properties. The second category allows you to investigate the relationship between the concepts of the ontological model. Metrics also allow you to analyze specific aspects of the ontology, since, in most cases, the method for constructing an ontology largely depends on the area in which it is developed or will be used [15].

In addition, it is advisable to carry out the process of assessing the quality of the designed ontological model in accordance with three main groups of metrics: structural, functional, and usability profile metrics [16]. These groups of metrics allow you to analyze the ontology according to such attributes as: syntactic quality; compliance with the rules; richness and semantic quality; interpretability; sequence; clarity and pragmatic quality; comprehensiveness, accuracy, and relevance [17].

The analysis of the characteristics of the ontological model should start with the basic structural metrics, which will allow, in combination with the scheme metrics, to determine the qualitative attributes of the created model. The mentioned metrics make it possible to identify ergonomic and structural-hierarchical errors in the ontological model at the initial stages of design.

Since the ontological model is the basis for the creation of a graph and a knowledge base, the initial assessment of the structural and hierarchical properties of the model is carried out using the metrics of depth, width, complexity of the ontology, and the presence of cycles [18]:

Absolute depth (A_{depth}) is defined as the sum of the lengths of all paths, where a path is a sequence of vertices from the root to the leaf vertex of the ontological model graph (1):

$$A_{depth} = \sum_{i}^{P} l_i \quad , \tag{2}$$

where P – is the set of all paths of the ontological graph;

l_i – the length of the i-th path from the total set P .

Average depth (Avg_{depth}) characterizes the ratio of the absolute depth of the ontological graph to the total number of paths:

$$Avg_{depth} = \frac{A_{depth}}{n_p}, \tag{3}$$

where A_{depth} – the absolute depth of the ontological graph;

n_p – the total number of paths from the root element to the leaf vertices.

Absolute breadth ($A_{breadth}$) determines the number of classes at each level of the ontological model hierarchy:

$$A_{breadth} = \sum_{i}^{L} N_i \ , \tag{4}$$

where L – is the set of all levels of the ontological graph;

N_i – the number of vertices at the i-th level of the ontological graph.

Average breadth ($Avg_{breadth}$) characterizes the ratio of the absolute width of the ontological graph to the total number of levels of the ontological model hierarchy:

$$Avg_{breadth} = \frac{A_{breadth}}{n_L}, \tag{5}$$

where $A_{breadth}$ – the absolute breadth of the ontology graph;

n_L – the total number of hierarchy levels in the ontological graph.

Ontology entanglement (Tang) indicates the presence of multiple inheritance: the lower the entanglement value, the better the projected ontology from the point of view of cognitive ergonomics. The entanglement metric is defined as the ratio of the number of classes having more than one parent class to the total number of classes of the ontological model:

$$Tang = \frac{t}{n_g}, \quad (6)$$

where t – number of vertices having more than one parent (multiple input connection *is-a*);
n_g – the total number of vertices in the ontological graph.

Ontology cyclicity (A_{cycle}) is an indicator of the presence of cycles used to determine the quality of the projected ontological model. The degree of cyclicity of an ontological graph is defined as the ratio of the number of cyclic connections to the total number of vertices in the ontological model:

$$A_{cycle} = \frac{N_{cycle}}{n_g}, \quad (7)$$

where N_{cycle} – the number of vertices included in any cycle of the ontological graph;
n_g – the total number of vertices in the ontological graph *G*.

The results of the evaluation of software system ontologies for classical [13] and abstract [3] approaches based on basic structural and hierarchical metrics of ontological models are presented in Table 1.

A comparison of the basic structural and hierarchical metrics for the proposed ontology design approaches shows that the values of the depth and breadth indicators of the ontology model are better for the abstract approach. Improvement of these metrics ensures faster and more correct processing of ontological SWRL rules in the adaptation process.

Metrics of entanglement and cyclicity of the ontological model allow us to determine the structural and functional errors in the process of designing the ontological model. In accordance with their semantic content, the higher the value of these indicators, the worse the structure of the model from the point of view of further modification.

Table 1.

Evaluation results of designed ontologies based on basic structural and hierarchical metrics

Metric	**The value of the metric**	
	Classical approach	Abstract approach
Depth metrics		
Absolute depth (A_{depth})	25	27
Average depth (Avg_{depth})	2,5	2,07
Breadth metrics		
Absolute breadth ($A_{breadth}$)	15	13
Average breadth ($Avg_{breadth}$)	3,5	2,6
Entanglement and cyclicity metrics		
Ontology entanglement (T_{ang})	0,0	0,0
Ontology cyclicity (A_{cycle})	0,0	0,0

In addition, the presence of cyclic dependencies can cause problems and slow down the work during the processing of ontological rules to determine the adaptive configuration. It should be noted that for both approaches the metrics of entanglement and cyclicity are equal to zero, which confirms the syntactic correctness and quality of the designed ontologies from the point of view of cognitive ergonomics and, accordingly, the expediency of using the ontological approach in designing the architecture of adaptive software systems in general.

In order to further assess the quality of the projected ontological models, an analysis of the taxonomy of subject area concepts was carried out. The quality assessment of the generated taxonomy was carried out on the basis of scheme metrics [19, 20]. These metrics allow you to analyze the fullness of instances, connections, and concepts, namely:

- *attribute richness metric (AR)* shows the distribution of functional properties among the projected classes of the subject area. This characteristic is defined as the ratio of the total number of ontology attributes to the total number of ontology concepts/classes:

$$AR = \frac{|ATT|}{|C|}, \quad (8)$$

where ATT – is the total number of ontology attributes;
C – the total number of ontology classes.

- *relationship richness metric (RR)* shows the distribution of properties that establish relationships between the concepts of the ontological model among the projected classes of the subject area.

Relationship richness is defined as the ratio of the number of non-inheritance relationships between ontology objects to the total number of ontology relationships:

$$RR = \frac{|P|}{|P| + |H|}, \tag{9}$$

where P – the number of relationships that do not include relationships is-a;
H – number of inheritance relationships (relationships is-a).

− *class-attribute ratio (ACR).* This metric analyzes the relationship between classes containing properties and all classes in the projected ontological model:

$$ACR = \frac{|N_{cl+att}|}{|C|} \tag{10}$$

where $|N_{cl+att}|$ - number of classes containing properties;
C – the total number of ontology classes.

The specified basic metrics and metrics of the scheme allow to evaluate the structure and relationships of the ontological model in order to determine the correctness of the construction of the model of the subject area. However, it should be noted that these metrics do not allow us to fully assess the informativeness and filling of entities with specific objects.

Since the main goal of building ontological models is to represent knowledge about the subject area, metrics that allow you to evaluate the quality of the ontology in the process of further expanding the knowledge base are also important characteristics. In this case, the placement of data in the ontology is analyzed with the help of appropriate metrics of the population of the ontology knowledge base: the average population of classes and the metric of class richness.

Average class population metric (AP) allows you to evaluate the distribution of entity instances among all classes of the ontology. A higher value of this metric indicates a better population of the knowledge base. Average class population is defined as the total number of instances in relation to the total number of classes of the ontological model:

$$AP = \frac{|I|}{|C|} \tag{11}$$

where *I* – the total number of entity instances;
C – the total number of ontology classes;

Class richness (CR) indicates the degree of connectivity of concepts and instances of entities of the ontological model and is defined as the ratio of the number of non-empty classes to the total number of classes of the ontological model:

$$CR = \frac{|c_{non-empty}|}{|C|} \tag{12}$$

where $c_{non-empty}$ – the number of classes that have at least one instance
C – the total number of ontology classes

The determined metrics of the scheme and filling of the knowledge base for the projected ontological models for the classical and abstract approaches are presented in Table 2.

Table 2.

The results of the evaluation of the designed ontologies based on the metrics of the scheme and filling of the knowledge base

Metric	**The value of the metric**	
	Classical approach	Classical approach
Schema metrics		
Attribute richness (AR)	1,33	1,38
Relationship richness (RR)	0,294	0,285
Class-attribute ratio (ACR)	0,53	0,46
Knowledge base population metrics		
Average class population (AP)	110	127
Class richness (CR)	0,67	0,69

According to the evaluation results, the values of the attribute richness metric and the relationship richness metric are approximately the same in both cases. Consequently, for both methods, an equivalent distribution of functional properties among the designed classes and properties establishing relationships between the concepts of the ontological model is achieved. However, it should be noted that the value of the scheme metrics for the classical approach is not stable, since new classes will be created in the process of software adaptation, which will affect the average richness of classes with attributes.

Analysis of approaches to ontology design based on knowledge base filling metrics proves that for the abstract approach richness indicators and average class population increase. Such values indicate greater informativeness of the ontological model designed according to the abstract approach, its semantic quality and interoperability, as well as the possibility of its further expansion.

Thus, the determined values of the specialized metrics for both approaches allow us to assert the compliance of the designed ontologies with the requirements in terms of the model structure and dependencies between components, as well as the equal distribution of the functional properties of the classes. From the point of view of the further development of the ontology and the filling of the knowledge base, the speed and correctness of processing the ontological SWRL rules in the process of adaptation, the results of the evaluation of specialized metrics prove the advantages of using an abstract approach in the design of ontological models for adaptive software systems.

Conclusions

An analysis of research aimed at designing adaptive software systems using ontological models has been carried out. It was established that the use of ontologies in the process of software configuration allows to improve the adaptation mechanism due to the use of a formalized and standardized description of the subject area. Such a solution allows to reduce the load on system resources, which, in turn, reduces the requirements for hardware.

The process of designing ontological models based on classical and abstract approaches is presented. It is established that when using the classical approach to designing ontologies, problems of migration and updating of software modules arise during the software adaptation process. The solution to this problem is an abstract approach to designing an ontological model, according to which abstractions of objects of the subject area are distinguished and created. Such a solution provides the possibility of further expansion of the model by an instance of ontological concepts without the need to reconfigure the software system.

A set of structural and hierarchical metrics for evaluating the syntactic and semantic quality of ontological models is defined. A comparative analysis of approaches to the design of ontological models based on the defined metrics of the scheme and the population of the ontological knowledge graph was carried out. The obtained values of the structural and hierarchical metrics confirm the quality of the designed models and their compliance with the requirements for software adaptation. The metrics of entanglement and cyclicity defined for both approaches are zero, which indicates the syntactic correctness of ontology design from the point of view of cognitive ergonomics. This provides the possibility of further modification of the models and confirms the expediency of using the ontological approach in designing the architecture of adaptive software systems in general.

The improved values of the specialized metrics of the scheme and the population of the knowledge base of the ontology developed on the basis of the abstract approach confirm its higher efficiency in the design of ontological models, as it ensures faster and more correct processing of ontological SWRL rules in the adaptation process. The defined metrics indicate greater informativeness of the ontological model designed according to the abstract approach, its semantic quality and interoperability, as well as the possibility of its further expansion.

References

1. Bybychkov Y., Sokol V., Shevchenko A. A model of self-adaptive software based on knowledge from corporate memory. *Weapon systems and military equipment*. 2017. Vol. 4. P. 137–144.

2. Vyshnevskyy O., Zhuravchak L. Semantic Models for Buildings Energy Management. *2023 IEEE 18th International Conference on Computer Science and Information Technologies (CSIT)*, Lviv, Ukraine, 19–21 October 2023. Lviv, 2023. URL: https://doi.org/10.1109/csit61576.2023.10324108.

3. Fedasyuk D., Lutsyk I. The Use of Ontology in the Process of Designing Adaptive Software Systems. *2022 IEEE 17th International Conference on Computer Sciences and Information Technologies (CSIT)*, Lviv, Ukraine, 10–12 November 2022. Lviv, 2022. URL: https://doi.org/10.1109/csit56902.2022.10000528.

4. User interface design patterns and ontology models for adaptive mobile applications / A. Braham et al. *Personal and Ubiquitous Computing*. 2021. URL: https://doi.org/10.1007/s00779-020-01481-5.

5. Mobile Devices Interface Adaptivity Using Ontologies / M. Waseem Iqbal et al. *Computers, Materials & Continua*. 2022. Vol. 71, No. 3. P. 4767–4784. URL: https://doi.org/10.32604/cmc.2022.023239.

6. An intelligent system for energy management in smart cities based on big data and ontology / Z. Sayah et al. *Smart and Sustainable Built Environment*. 2020. Vol. 10, No. 2, P. 169-192. URL: https://doi.org/10.1108/sasbe-07-2019-0087.

7. Ontology Driven Smart Health Service Integration / S. K. Shahzad et al. *Computer Methods and Programs in Biomedicine*. 2021. Vol. 207. P. 106146. URL: https://doi.org/10.1016/j.cmpb.2021.106146.

8. Analysis of Ontology Quality Dimensions, Criteria and Metrics / R. S. I. Wilson et al. *Computational Science and Its Applications – ICCSA 2021*. Cham, 2021. P. 320–337. URL: https://doi.org/10.1007/978-3-030-86970-0_23.

9. Quality model and metrics of ontology for semantic descriptions of web services / H. Zhu et al. *Tsinghua Science and Technology*. 2017. Vol. 22, No. 3. P. 254–272. URL: https://doi.org/10.23919/tst.2017.7914198.

10. Ontology Quality Evaluation Methodology / R. S. I. Wilson et al. *Computational Science and Its Applications – ICCSA 2022*. Cham, 2022. P. 509–528. URL: https://doi.org/10.1007/978-3-031-10522-7_35.

11. Bravo Contreras M. C., Hoyos Reyes L. F., Reyes Ortiz J. A. Methodology for ontology design and construction. *Contaduría y Administración*. 2019. Vol. 64, No. 4. P. 134. URL: https://doi.org/10.22201/fca.24488410e.2020.2368.

12. Chaudhary N., Kumar S., Gupta S. A Novel Ontology Design and Comparative Analysis of Various Retrieval Schemes on Education Domain in Protégé. *ICT Analysis and Applications*. Singapore, 2020. P. 487–495. URL: https://doi.org/10.1007/978-981-15-8354-4_48.

13. Fedasyuk D., Lutsyk I. Tools for adaptation of a mobile application to the needs of users with cognitive impairments. *2021 IEEE 16th International Conference on Computer Sciences and Information Technologies (CSIT)*, Lviv, Ukraine, 22–25 September 2021. Lviv, 2021. URL: https://doi.org/10.1109/csit52700.2021.9648702.

14. Fedasyuk D., Lutsyk I. Approach to Implementation of Configuration Process for Adaptive Software Systems based on Ontologies. *International Journal of Computing*. 2023. P. 381–388. URL: https://doi.org/10.47839/ijc.22.3.3234.

15. Tartir S., Arpinar I. B. Ontology Evaluation and Ranking using OntoQA. *International Conference on Semantic Computing (ICSC 2007)*, Irvine, CA, USA, 17–19 September 2007. Irvine, 2007. URL: https://doi.org/10.1109/icosc.2007.4338348.

16. Evaluating the Good Ontology Design Guideline (GoodOD) with the Ontology Quality Requirements and Evaluation Method and Metrics (OQuaRE) / A. Duque-Ramos et al. *PLoS ONE*. 2014. Vol. 9, No. 8. P. e104463. URL: https://doi.org/10.1371/journal.pone.0104463.

17. Lourdusamy R., John A. A review on metrics for ontology evaluation. *2018 2nd International Conference on Inventive Systems and Control (ICISC)*, Coimbatore, 19–20 January 2018. 2018. URL: https://doi.org/10.1109/icisc.2018.839904.

18. Lantow B. OntoMetrics: Putting Metrics into Use for Ontology Evaluation. *8th International Conference on Knowledge Engineering and Ontology Development*, Porto, Portugal, 9–11 November 2016. Porto, 2016. URL: https://doi.org/10.5220/0006084601860191.

19. OQuaRE: A SQuaRE-based Approach for Evaluating the Quality of Ontologies / A. Duque-Ramos et al. *Journal of Research and Practice in Information Technology*. 2011. Vol. 43. P. 159–176.

20. McDaniel M., Storey V. C. Evaluating Domain Ontologies. *ACM Computing Surveys*. 2019. Vol. 52, No. 4. P. 1–44. URL: https://doi.org/10.1145/3329124

Illia Lutsyk **Ілля Луцик**	PhD student, Department of Software, Lviv Polytechnic National University, Lviv, Ukraine, e-mail: illia.i.lutsyk@lpnu.ua. https://orcid.org/0000-0001-7117-5332	аспірант кафедри програмного забезпечення, Національний університет «Львівська політехніка», Львів, Україна.
Dmytro Fedasyuk **Дмитро Федасюк**	DrS on Engineering, Professor, Head of the Department of Software, Lviv Polytechnic National University, Lviv, Ukraine, e-mail: dmytro.v.fedasyuk@lpnu.ua. https://orcid.org/0000-0003-3552-7454	доктор техн. наук, проф., зав. кафедри програмного забезпечення, Національний університет «Львівська політехніка», Львів, Україна.