Use of Ontological Knowledge for Multi-Criteria Comparison of Complex Information Objects

Julia Rogushina[1], Anatoly Gladun[2]

[1] *Institute of Software Systems of the National Academy of Sciences of Ukraine, 40, Ave Glushkov, Kyiv, 03181, Ukraine*
[2] *International Research and Training Center for Information Technologies and Systems under NAS and MES of Ukraine, 40, Ave Glushkov, Kyiv, 03680 GSP, Ukraine*

Abstract

We propose ontology-based formal model of complex information object (CIO) as an element of decision making by intelligent information systems. The main stages of CIO comparison with similar structure based on the use of knowledge from domain are considered. We use various evaluations of semantic proximity and semantic similarity to match CIO properties and their values with requirements of user task that are formalized as CIO reference model. The basis for the reference CIO construction is the natural language task description compared with domain ontology that defines the CIO structure.

Domain ontology is used also as a source of comparison criteria that can be constructed from various combinations of characteristics of ontology classes and individuals used in CIO elements, and we propose an algorithm for recursive generation of the CIOs comparison criteria set. Decision that we retrieve is a CIO that is the most similar to this reference model according to these criteria but current significance of them is defined as hierarchy by experts for actual environment.

Keywords [1]

Complex information object, ontology, semantic similarity, semantic proximity

1. Introduction

In this work we consider comparison of *complex information objects* (CIO) as a component of intelligent *decision-making* (DM). For many tasks available decisions can be represented as some complex sets of information objects with different structure linked by various meaningful relations. At the same time, generation of decision as a sequence of certain actions in a dynamic information environment is beyond the scope of this work. We pay attention to some group of specific situations where comparison of CIOs is based on big number of criteria, and their selection and evaluation of their relative importance for current task is a significant part of general DM process.

Such situations are typical for areas where:

- information is obtained from a dynamically changing environment;
- decision can use some fixed set of accessible objects;
- information structure is defined by various external knowledge sources that are not specific for user task and therefore contains a lot of unnecessary elements;
- the set of available resources and their values also changes over time.

Specifics of the proposed approach is based on the relatively small number of compared CIOs defined by task semantics: we have to compare not all are theoretically possible CIOs, but only those ones that can be chosen in the current situation. The comparison problem is not consist in finding an optimal (according to some criteria) solution, but it has to provide selection of an acceptable solution from the limited set of available ones. For example, fulfillment of research project requires to select a group of employees from a certain department, and not from all scientists from this domain.

[1]13th International Scientific and Practical Conference from Programming UkrPROGP'2022, October 11-12, 2022, Kyiv, Ukraine
EMAIL: ladamandraka2010@gmail.com (A. 1); glanat@yahoo.com (A. 2)
ORCID: 0000-0001-7958-2557 (A. 1); 0000-0002-4133-8169 (A. 2)

Therefore for some situations all possible solutions can be unsatisfactory and change the situation only for the worse. For example, project fulfillment by incompetent employees will lead to a loss of time and resources, but the desired result will not be obtained. At the same time, the significance of the comparison criteria can be change over time due to changes in the dynamic information environment, and unsatisfactory CIO becomes acceptable. The most common example of changing priorities is the cost of performing work and the speed of obtaining results: in some extreme conditions time becomes the most important criterion instead of value or potential damage.

2. Complex information objects

From the point of view of ontological analysis, information objects (IOs) are considered as classes or instances of ontology. Ontology classes are characterized by their structure as a set of properties and their characteristics, as well as possible relations with other classes. Instances of ontology classes can also have the values of properties defined by constants or by instances of ontology classes. But many practical tasks need in analyzes of more complex sets of information where IOs are related to each other by certain relations and satisfy some restrictions.

Complex semantic search usually provides many examples of such a task: to find a group of people with certain qualifications that work in the same organization from the defined set; to determine the countries where results of scientific projects on a certain topic are published in a selected set of journals for a certain period of time, etc.

The search results are usually limited by the set of IOs from one or several classes (IO "Person", IO "Organization", etc), while restrictions are used only to select acceptable values of their properties. But many other tasks need in result represented by the set of IO collections of different types linked by relations that corresponds to certain more complex conditions. Examples of them are staff and environment of organization allocates for some project; plan of learning for desired vacancy coordinated with specialties provided by educational institutions; composition of programming committee by topics of scientific conference; the infrastructure of the settlement with means of its support and personnel; set of hierarchically related units that perform common task with use of own technical means.

We define CIO as a set of more than one IO, which are related to each other by ontological relations and meet the requirements regarding the structure and values of CIO properties [1].

We have a long-time experience in the development of information systems that apply the comparison of various CIOs. In this work, we consider problems related to the generation of comparison criteria on base of the domain ontology and determining relative significance of these criteria for the current state of the information environment.

3. CIO ontological model

The formal model of CIO is based on the formal model O_{domain} of domain ontology O that is selected according to user task:

$$O_{domain} = < T, R, F > \quad (1),$$

where

- T is a finite set of domain concepts, which is divided into a set of classes T_{cl} and a set of instances of classes T_{ind};
- R is a finite set of domain relations between concepts from T;
- F is a finite set of interpretation functions for concepts and relations of ontology O.
- CIO model contains a subset of elements O_{domain}: IOs from T that belong to CIO and link by semantic relations from R.

Formal model C of CIO has the following structure:

$$C = < T_C, N_C, R_C, \{(t_j, t_k, r_m), t_j \in T_{ind}, t_k \in T_{ind}, r_m \in R\} > \quad (2).$$

where

- $T_C = \{t_i, i = \overline{1,p}\} \subseteq T$;
- $N_C = \{n_i, i = \overline{1,p}\}$;
- $R_C \subseteq R$.

Formal model of domain ontology defines all classes of domain with their properties, characteristics and possible links between their individuals. CIO model uses only some subset of them, and formal description of this subset structure is a definition of CIO. It is important that CIO model, in contrast to ontology model, differs positions of class individuals to indicate by unique names (see Fig.1). Each CIO element is associated with a set of characteristics and restrictions (for example, the element has to be present, has to have a single value or can have several values, etc.).

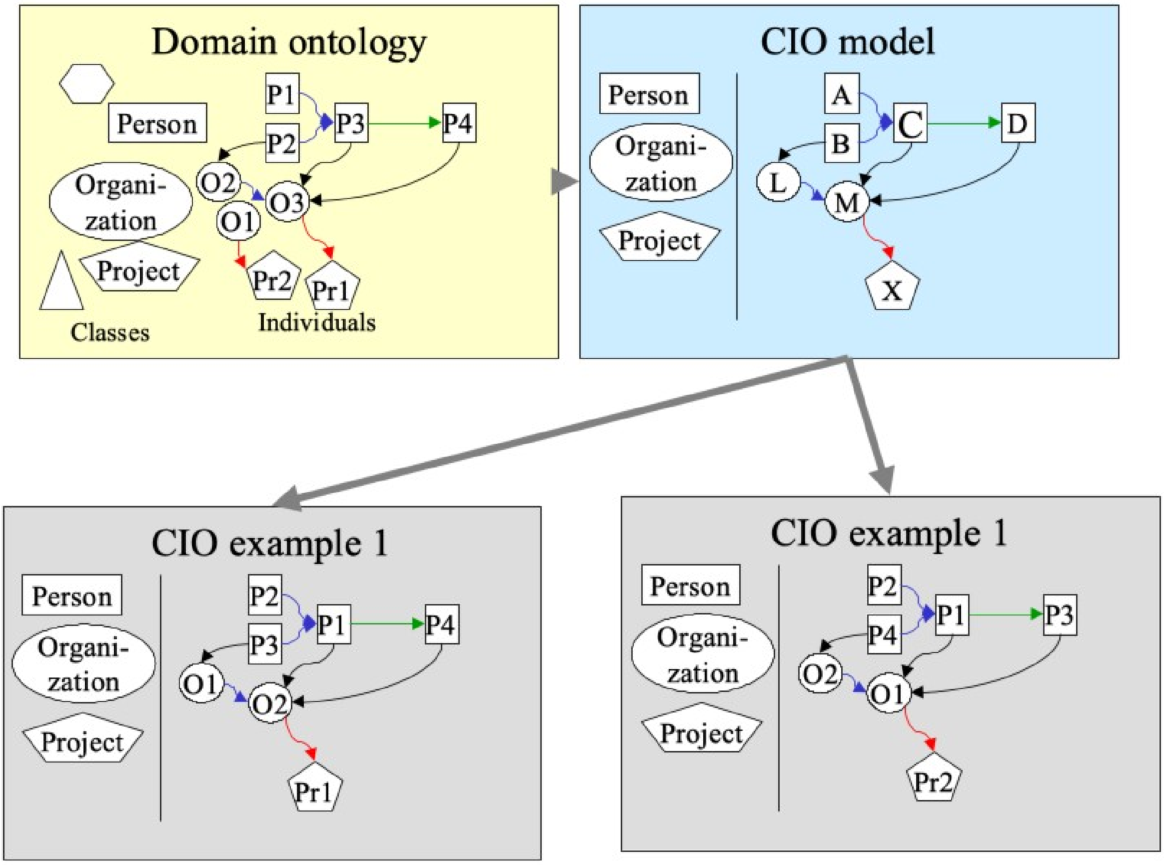

Figure 1. Domain ontology and CIO structure.

If such separation is not important for domain, several formally different CIOs are considered as one. For example, CIO structure contains two IOs of the class "Person" designated by the names "Project Manager" and "Project Executor" respectively. IO "Project Manager" is indicated as mandatory and single-valued, and IO "Project executor" is indicated as mandatory and multi-valued. Then the order of "Project Executor" individuals is not significant; that is, there is no difference between the CIOs, which differ only in the order of information about them. But if two CIOs contain the same "Person" individuals indicated by different IOs then these CIOs are different.

4. Intelligent decision-making

The main tasks of DM are:

- selection of DM criteria;
- development of compatible metrics for quantitative assessment criteria;
- assessment of the quality and cost of data used for DM process;
- selection of requirements for the quality of the solution;
- analysis of factors affecting the decision-making process;
- selection of representation means for used data and solution;
- optimization and integration of results for group DM;

hierarchy and multi-stage goals in the development of the decision-making context.

Now DM uses various intelligent elements of data science and knowledge management. For example, decision intelligence [2] can be defined as intelligent DM oriented on business problems. It is a new scientific discipline that combines applied data science, artificial intelligence, social sciences and management theory. This theoretical ground provides powerful means for developing goals, metrics and evaluating criteria for decisions in various domains. In this context, decision is considered as any choice between options for any entity – a person, an organization, a software agent, etc. or combination of entities. In this sense, DM can not be reduced to some binary alternative to perform one specific action. Intelligent DM incorporates different decision-making methods like rules-based approaches to ML and AI.

Many of these tasks require the involvement of external sources of knowledge about decision domain, about personifies needs and current interests of users, and about environment where these solutions are implemented. Intelligent DM can process domain knowledge that provide more efficient solutions of these tasks.

Therefore decision intelligence considers some additional DM aspects based on use of knowledge management and logical inference and transforms traditional ones according to requirements of knowledge representation tools. For example, ontological structures of various volume and complexity can be involved.

This is justified by the fact that decision-making strategies that are based only on quantitative assessments without a qualitative knowledge about DM domain are, as a rule, less effective in comparison with approaches that also use elements of semantic analysis.

DM process, as well as matching of other CIOs, depends on the available information. If decisions are made in an open information environment (which is most typical for practical problems), then the information may be incomplete, unclear and contradictory. Moreover, some part of the facts may simply be unreliable. Other important factors influenced on DM results are the representation form of the input information, selection of criteria for matching of particular solutions and evaluations of their values.

5. Ontology-based CIO comparison

Comparison of CIOs at the semantic level can use ontological knowledge for two aims:

- to evaluate the semantic proximity between IOs occupying a certain place in different IOs;
- to evaluate the semantic similarity between relations that connect these IOs.

In both cases, evaluations take into account the semantic distance between the corresponding classes of the domain ontology and the closeness of the property values for class instances.

Such evaluations can use various metrics of semantic proximity and semantic similarity that transform the qualitative knowledge representation of domain ontology into quantitative characteristics of their semantic proximity and affinity.

6. Semantic proximity and semantic similarity evaluations

Ontology can be considered as a hierarchical semantic network where nodes correspond to domain concepts (meaning units), and the directed arcs correspond to various semantic relations between concepts. The meaning of a concept is described by its relations with other concepts. To compare CIOs, it is necessary to identify semantic similarity and semantic proximity between the concepts included to their ontological models.

There is a significant difference between the "semantic similarity" and "semantic proximity" terms: semantic similarity is much broader. Semantic similarity is based on the relations of synonymy and "class-subclass" between concepts, while semantic proximity takes into account all other domain relations between these concepts (for example, the relation of antonymy or meronymy). Choice of semantic similarity estimates depends on task specifics [3].

Semantic similarity is related to information content of concept. The informational content of the concept A is defined as $-\log p(c)$: the higher probability of the concept use causes its lower informativity. Thus, the higher level of the concept abstraction (that is, the higher place into domain taxonomy) causes the less information content. The similarity of concepts A and B is evaluated by finding the maximum information content over such concepts where A and B both can be instances. This approach provides to create sets of semantically close concepts (SCCs), i.e. concepts with semantic distances from selected one less than the selected threshold value.

Analysis of research works related to methods of semantic proximity and semantic similarity of domain SCCs allows to divide them into four groups:

- semantic similarity algorithms based on domain knowledge [4];
- methods based on informational content of concepts [5];
- methods of semantic proximity [6] based on various vector representation of natural language words;
- hybrid and generalized methods [7] that combine various approaches.

Therefore, methods used for estimation the semantic similarity between CIO concepts can be divided into groups:

- based on attributes of IOs of CIO ;
- based on content of CIOs (individuals);
- based on semantic distance between CIOs;
- hybrid methods that take into account structure, values and individuals of CIOs.

Majority of CIO comparison methods evaluate semantic similarity on base of "attribute-value" pairs by quantitative similarity measures of these attribute values. However, a simple vector of attributes does not sufficiently reflect the complexity of CIOs that appear in practice, first of all it is necessary to know the structure of the CIO defined by its ontology [8].

Many researchers suggest that semantic similarity should take into account the hierarchical structure of the CIO ontology. A content-based similarity algorithm determines the similarity of two classes by comparing the content information contained in the common parent node of the classes and ignores the content information contained in the class itself. The main idea of the distance-based semantic similarity algorithm is to calculate the semantic distance between two concepts in the classification tree based on ontology [9]. The main drawback of this method is the assumption that the distance of all edges in the system is equal.

Resnik [10] offers an alternative way for evaluation of similarity in the semantic structures that is also not sensitive to the different sizes of distances between relations: such similarity can be considered as a taxonomic relations with ignoring other ontological relations. This approach is suitable for many practical tasks but it leads to the loss of some potentially useful information.

Another important term in semantic matching in data mining is *semantic correlation* that differs significantly from semantic similarity. Semantic correlation is related to the degree of interconnection between two concepts. Semantic similarity aggregates concepts and relations, while semantic correlation is a combination of concepts. For example, automobiles have semantic correlation with fuel, but automobiles and bicycles are semantic similar as subclasses of transport, but are not similar, where automobiles and bicycles are more similar semantically, but have not semantic correlation.

Domain ontologies can be used as a base of semantic similarity evaluations [11]. Ontologies contain formalized knowledge about relations between domain concepts and their properties. This knowledge van be acquired from ontology according to parameters used by particular evaluation. For example, various ontology-based evaluations can define semantic similarity between concepts by analyzes relation "is-a", and correlation between two concepts can be defined by any other type of ontological relations, for example, "part-of".

A special case of ontologies is taxonomies. They are a fairly common and convenient source of knowledge for analyzing the semantic closeness of NL concepts and words. The most popular way of evaluation of semantic similarity by taxonomy is based on measuring the distance between net nodes that correspond to the elements being compared: the shorter path from one node to another means their higher similar. If elements are connected by multiple paths between them then the shortest path length is used

It is important to understand that semantic similarity and correlation both depend on interpretation or context, and therefore these measures depend on selected ontology and analyzed set of relation.

Evaluations of the semantic similarity between the domain concepts help in formalization of the information needs of users represented by natural language texts [12-14] and describe the structure and properties of the desired solutions. They can be used to build a formalized thesaurus of the problem, which becomes a source of information about the structure of CIO [15].They can be used to find CIOs that are the most similar to the reference CIO.

7. Problem definition

If we consider CIO as a decision of some user task, then we need in means for formalization of user requirements for relevant decisions (some reference model of CIO) and for criteria of selection the most satisfactory from them. The metrics of semantic similarity discussed above allow to quantify the semantic proximity between elements of different CIOs and between different CIOs as a whole as an instrument of their comparison. In this work we define main stages of this comparison and propose methods for their execution.

8. Stages of CIO comparison

In general, the task of comparing CIOs with arbitrary structure that are based on different ontologies requires the alignment of these ontologies and the search for similarities between their structural elements – IOs and their combinations. In this work, we consider a subtask of such problem: compared CIOs that are based on a single ontology and have the same (or similar) structure. Differences between compares CIOs are represented by used IOs, the sets of defined IO properties and values of these properties. Processing of CIOs with arbitrary structure includes this subtask at the last stage of comparison, but many practical IISs reduce CIO comparison by these restrictions.

Considered type of comparing is typical for tasks solved by retrieval services, for decision-making tools, and for various recommender and advisory systems. In practice, many of such systems can compare CIOs with different structures (such as resumes and vacancies), but semantic matching is performed for subsets of CIO elements that have a similar structure – for example, for a set of competencies or project descriptions.

The task of comparing CIO with a similar structure is divided into the following stages:

- creation of a reference formalized structural model of CIO $c_{refer} \in C$, which reflects the main requirements and limitations determined by the user's task, with structure according to (2);
- generation of the set of CIOs, $C_{current} = \{c_i \in C, i = \overline{1,n}\}$ that can be constructed from current IOs (according to information about instances of the real environment at a certain point in time) and correspond structurally to this reference model;
- selection of the subset $C_{avail} = \{c_j \in C, f(c_j, c_{refer}) \leq A_{task}, j = \overline{1,m}\} \subseteq C_{current}$ of available CIOs that meet the user's requirements at the semantic level, i.e. are at a semantic distance from the reference model defined with the help of some estimation $f(c_1, c_2), c_1 \in C, c_2 \in C$ no more than a certain value A_{task};
- generation of the non-empty criteria set $D = \{d_k = g(T_{cl}, R), k = \overline{1,p}\}$ for CIO comparison on base of domain ontology O defined by (1);
- estimation of the significance level $s_k = h(d_k, t), k = \overline{1,p}$ of each individual criterion of the comparison from D at the current moment t based on expert evaluations and domain heuristics (it is important to consider that the significance of criteria in a dynamic information environment can change significantly over time for the same user task, but the set of criteria that is built for a certain task based on the selected domain ontology, as a rule, does not change at all or is only supplemented with additional criteria);
- determination of the quantitative assessment $e_j, j = \overline{1,m}$ of each CIO from C_{avail} based on the selected set of criteria $d_k = g_k(T_{cl}, R)$ and their significance level $s_k = h(d_k, t)$ of its similarity to the reference model and the selection of the most suitable CIO (it should be taken into account that such a choice of CIO is not optimal in the global sense, and changes in the level of significance of the criteria affect such a choice):

$$e_j = \sum_{k=1}^{p} g_k(T_{cl}, R) * h(d_k, t), j = \overline{1,m} \qquad (3).$$

Estimation (3) is used for ordering available CIOs according to their semantic similarity to reference model in consideration of current user priorities.

Every stage of this process needs in relevant algorithms and data representations. Below we consider the most important characters of these stages that are general for all their practical realizations.

9. Generation of the CIOs comparison criteria set

We consider comparison criteria as a mean to find CIOs that are the most semantically similar to reference model proposed by user. This model has to represent all main aspects that are important for current user task. The review of methods for determining the semantic similarity between the domain concepts shows that parameters for determining the semantic proximity between two ontology class individuals are defined by:

- the semantic distance between their classes (defined by some subset of ontological relations with certain characteristics such as hierarchical and synonymous ones);
- the semantic proximity between the property values of the same attributes.

The task is greatly simplified for matching instances of the same class. Then the first group of parameters can be ignored, and the second one does not need in aligning the properties of different classes by analysis of their semantics (for example, a property of the type "Year" can characterize both the year of birth of a person and the year of the start of education). The problem of comparison of CIOs with a different structure is more complex and requires additional stages of information gathering. In this work we consider situation of matching CIOs with the same or similar structure, and therefore alignment can be reduced to search of subclasses and superclasses for analyzed IOs.

We propose the following algorithm for the preliminary generation of the criteria set for matching CIOs with a similar structure:

- each IO $t_i, i = \overline{1,p}$ based on the formal model (2) defines a set of its properties K_{0_i} divided in data properties $T\_data_{0_i} = \{t\_data_{i_m}, m = \overline{0, y_i}\}$ (their values of ontology class attributes $t\_d_{i_m}, i = \overline{1,p}, m = \overline{0, y_i}$ are constants of various types – number, text, date, etc.), and object properties $T\_o_{0_i} = \{t\_o_{i_k}, m = \overline{0, z_k}\}$ (their values $t\_o_{i_k}, i = \overline{1,p}, m = \overline{0, z_k}$ are individuals of different classes of ontology O that are used as attribute values of other individuals in domain ontology for IO class): $K_{0_i} = T\_d_0 \cup T\_o_0$.
- for data properties, the analysis ends here, and for object properties, if necessary, it can be repeated recursively for each IO class that can be a property value for t_i for replenishment of the set K_{0_i} with the corresponding properties considered as additional similarity criteria for CIO matching;
- K_0 is generated by of combining the information from the sets K_{0_i} with clear fixation of IOs used as evaluation criteria: $K_0 = \{(t, i), t \in K_{0_i}, i = \overline{1, p}\}$;
- The user analyzes the constructed set K_0 and can explicitly remove some criteria that she/he considers irrelevant for current task.

At this point, the work of the algorithm can be completed or continued for construction of the criteria set K_{sem} enriched by other classes of the domain O ontology selected with use of various measures of semantic closeness and semantic similarity.

The set K_{sem} is constructed as follows:

- a subset T_{C_cl} of classes of ontology O that contains the appropriate criterion in K_0 is defined: $T_{C\_cl} \subseteq T_C, T_{C\_cl} \subseteq K_0$;
- a set S_j of semantically close or semantically similar concepts of domain ontology (according to the selected measure $f(t_a, t_b)$ and constant L) for each element $t_j \in T_{C\_cl}, t = \overline{1, x}$, is defined: $S_j = \{t \in T_{C\_cl}, f(t, t_j) \leq L\}$;
- the criteria set of K_{sem_i} for each set $S_j, j = \overline{1, x}$, is built by the same algorithm as sets K_{0_i} are built;

- the sets K_{sem_i} are combined into single set K_{sem} (in the same way as K_0 is built) where is also clearly fixed what IOs of CIO are used for the evaluation criteria: $K_{sem} = \{(t, i), t \in K_{sem_i}, j = \overline{1, x}\}$.

Then the user analyzes the criteria set K_{sem} and, if it necessary, explicitly removes from it those criteria that she/he considers insignificant for current task. In addition, the user can manually add those criteria that she/he considers important, but which are not represented by the domain ontology or were not included to K_{sem} by the algorithm proposed above. Some criteria can be lost because of unsuccessfully chosen ontology or estimation of semantic closeness, as well as an insufficient number of the algorithm iterations.

Another reason of manual editing of K_{sem} can be the user's expert knowledge about analyzed domain: it is easier for her/him to clearly indicate the important criteria than to look for them in the ontology structure. But it should be assumed that quite often such expert knowledge relates only to some particular aspects of the problem, and the use of the proposed algorithm ensures that other elements of domain knowledge are taken into account.

The next step of CIO matching deals with a hierarchy of criteria from K_{sem} that represents the current needs of the user and their relative importance. Such hierarchy can be determines with the help of analysis of the semantic proximity estimates between pairs of CIOs (including between the evaluated CIOs and the reference CIO, which is built according to the user's task description). Values of these estimates can be defined by user (or group of users) and external domain experts or acquired from pertinent knowledge bases. Selection of methods used for it depends on task specifics, user qualification and dynamics of user preferences.

10.Generation of the CIO reference model

The generation of the CIO reference model provides some formalized description of the user requirements in terms of domain ontology O. This model can be represented as an instance of this CIO with specified attribute values of its IOs. This CIO is acceptable for user needs, but real CIOs can contain various values for attributes that are not defined unambiguously in reference model.

The basis for the reference CIO construction is the natural language task description compared with domain ontology that defines the CIO structure. This comparison can be based on the task thesaurus, as described in [15].

In some cases, a desirable situation that satisfies the user can be described by more than one CIO (at the same time, combinations of such CIOs are unsatisfactory). The simplest example of such a situation is that you can use nails and a hammer or screws and a screwdriver to perform certain repair tasks, but you cannot use a combination of hammer and screws. Then the user task is transformed into a set of variant tasks defined by different the reference CIOs, and each of them is processed separately.

Task thesaurus can be constructed as a combination of thesauri of natural language documents selected by the user or obtained from the relevant domain ontology. Formal model of task thesaurus is based on formal model of ontology (1):

$$Th = < T_{th}, R_{th}, I > \quad (4),$$

where $T_{th} \subseteq T$ is a finite set of the ontological concepts; and $R_{th} \subseteq R$ is a finite set of the relations between these concepts, and set I represents additional information about concepts that depends on specifics of thesaurus goals and can contain, for example, weight of term or it's definition).

There is important to understand that task thesaurus Th based on domain ontology O is a special case of ontology but is not a subset of O. It has another structure that is simplified by reducing of arbitrary ontological relations (all information about these relations that is important for task is used for construction of T_{th} but is not included into it). T_{th} contains additional information about every concept – it's weight $w_i \in W, i = \overline{1, n}$. Therefore, formal model of task thesaurus is defined as set of ordered pairs $Th_{task} =< \{(t_i \in T_{th}, w_i \in W)\}, \varnothing, I >$ with additional information in I about source ontologies.

In general case, the user task can be described by some structured, semi-structured or non-structured document that uses one or several natural languages. If we have no additional information about structure of this document that can be used for its analysis, then we have to consider it as non-structured one. The task thesaurus built by the natural language description of task is based on linguistic analysis of its content and metadata matched with elements of domain ontology, i.e. task thesaurus does not include all available ontological concepts, but only their subset related to the current task. This approach reduces the volume of processing information and time of its analysis.

11.Conclusions

The comparison of CIOs with similar structure is a necessary component in the comparison of CIOs with different structures based on different ontologies that provides the theoretical background for decision making in open information environment. Comparison of two CIOs with arbitrary structure requires some additional steps. We have to align domain ontologies that define the structure of these CIOs and find correspondences between their concepts and relations that are used in CIO elements. Then we have to find some elements of different (IOs or IO groups) with similar structure (this similarity can be defined by property sets and by property values). At last we have to compare such similar elements according, and this comparison can be executed according to the algorithm discussed above.

The proposed theoretical models of CIOs and methods of their comparison can be used to support various actual practical tasks. For example, such matching of object with complex structure under dynamic requirements would be useful for risk management, rapid adjustment of industry for the production of important products, restorative construction, dynamic adaptation of teams, organizations, collectives with a complex structure (research groups, military units, expert commissions, rapid response medical teams).

In general, it can be used to perform various operational tasks in the absence of sufficient competencies, skills and experience for situations where decision making is based on big number of criteria, and their relative importance can be changed for the same task in different moments of time. We consider tasks that are not oriented on optimal decision but with significant restrictions for processing time and used resources because such approach reduces the number of compared elements but needs in quick algorithms and adaptive solutions.

12.References

[1] J. Rogushina, A. Gladun, Semantic processing of metadata for Big Data: Standards, ontologies and typical information objects. CEUR Vol-2859, ITS 2020, Information Technologies and Security, 2020, pp.114-128. URL: http://ceur-ws.org/Vol-2859/paper10.pdf

[2] C. Kozyrkov, Introduction to Decision Intelligence, 2019. URL: https://towardsdatascience.com/introduction-to-decision-intelligence-5d147ddab767 [Accessed 12/07/2022].

[3] J. Rogushina, Use of Semantic Similarity Estimates for Unstructured Data Analysis. CEUR Vol-2577, ITS 2019, Information Technologies and Security, pp.246-258. URL: http://ceur-ws.org/Vol-2577/paper20.pdf.

[4] J. J. Lastra-Díaz, J. Goikoetxea, et al. A reproducible survey on word embeddings and ontology-based methods for word similarity: linear combinations outperform the state of the art. Engineering Applications of Artificial Intelligence, 85, 2019, pp.645-665. URL: https://doi.org/10.1016/j.engappai.2019.07.010. [Accessed 11/07/2022].

[5] J. J. Lastra-Díaz, A. García-Serrano, A new family of information content models with an experimental survey on WordNet. Knowledge-Based Systems, 89, 2015, pp.509–526.

[6] J.Camacho-Collados, M. T. Pilehvar, From word to sense embeddings: A survey on vector representations of meaning. Journal of Artifcial Intelligence Research 63, 2018, pp.743–788.

[7] M. A. H. Taieb, T. Zesch, M. B. Aouicha, A survey of semantic relatedness evaluation datasets and procedures. Artifcial Intelligence Review 53(6). 2019. pp. 4407–4448. URL: https://doi.org/10.1007/s10462-019-09796-3. [Accessed 11/07/2022].

[8] M. Gan, X. Dou, R. Jiang, From Ontology to Semantic Similarity: Calculation of Ontology-Based Semantic Similarity. Sci. World J. 2013, pp.1–11. .
[9] D. Lin, An Information-Theoretic Definition of Similarity. Proc.of the International Conference on Machine Learning, USA, 1998.
[10] P. Resnik, Using information content to evaluate semantic similarity in a taxonomy. Proc. of the14th international joint conference on Artifcial intelligence, V. 1. 1995, pp.448–453.
[11] D.Sanchez, M. Batet et al., Ontology-based semantic similarity: A new feature-based approach. Expert Systems with Applications 39, 9, 2012, pp.7718 -7728. URL: https://doi.org/10.1016/j.eswa.2012.01.082. [Accessed 4/07/2022].
[12] Z.Wu, M. Palmer, Verbs semantics and lexical selection. Proc. of the 32nd annual meeting on Association for Computational Linguistics. Association for Computational Linguistics, 1994, pp.133–138.
[13] J. Jiang , D. W. Conrath, Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. Proc.of the 10th Research on Computational Linguistics International Conference. 1997, pp.19–33.
[14] J.-B. Gao, B.-W. Zhang, X.-H., Chen A WordNet-based semantic similarity measurement combining edge-counting and information content theory. Engineering Applications of Artifcial Intelligence 39, 2015, pp.80-88. URL: https://doi.org/10.1016/j.engappai.2014.11.009. [Accessed 12/07/2022].
[15] J. Rogushina, A. Gladun, Task Thesaurus as a Tool for Modeling of User Information Needs, Chapter 7 in New Perspectives on Enterprise Decision-Making Applying Artificial Intelligence Techniques, 2021, Springer Verlag . pp. 385-403. DOI:10.1007/978-3-030-71115-3_17.