Semantic Approach to Decision Making in Comparison of Complex 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 analyze models and methods that can be used to compare complex information objects as components of decision-making in intelligent information systems. The proposed approach is based on the use of knowledge acquired from domain ontologies and metrics to determine the semantic proximity of ontology components. We propose an algorithm for the semantic comparison of objects with a similar structure defined by the common ontology. This allows for the generation of a finite set of comparison criteria that depend on the definition of the user task and domain ontology. The significance of all criteria for the current situation is evaluated dynamically by expert statements about their hierarchy. The proposed method considers an integrated approach to multi-criteria decision-making in conditions where the set of criteria itself is determined by the requirements of the task and domain knowledge, but at different points of time the relative importance of particular criteria can change dynamically. Practical use of proposed approach is demonstrated on task of the formation of unmanned aerial vehicles groups.

Keywords
Complex information object, decision-making, ontology, semantic similarity, multi-level hierarchical structure

1. Introduction

One of the key issues in the development of complex systems is to increase the efficiency of decision-making in problematic situations, such as risk management, management of complex projects and the allocation of human resources. *Decision Support Systems* (DSSs) are a specific subclass of *intelligent information systems* (IISs) that help specialists to form and choose the right alternative among a set of acceptable options for making responsible decisions, and often combine mathematical approaches of decision-making with logical and linguistic models that use knowledge-based methods and heuristics acquires from accumulated experience of domain experts.

In such systems, decision-making problems concern the selection of the most informative features used to compare potential solutions. Processing of complex solutions with big number of connected sub-elements is complicated by the need to analyze a significant number of parameters of these objects contained into potential solutions, and the necessity of unification of their structure. In an open environment, the lack of time for processing of information and its rapid changes limits the possibility of using traditional decision-making methods that ensure finding the optimal solution. One

XXII International Scientific and Practical Conference "Information Technologies and Security (ITS-2022)", November 16, 2022, Kyiv, Ukraine
EMAIL: ladamandraka2010@gmail.com (J.Rogushina); glanat@yahoo.com (A.Gladun)
ORCID: 0000-0001-7958-2557 (J.Rogushina); 0000-0002-4133-8169 (A.Gladun);

of the directions for solving this problem is the application of elements of artificial intelligence (AI) and use of domain knowledge from external sources.

Currently, IISs implement knowledge based on ontological analysis [1], which is a special case of semantic analysis [2]. Ontologies provide interoperability and unequivocal interpretation of the domain knowledge, and therefore they can be considered as external sources based on ontological knowledge representation of information for different IIS, including the decision support subsystems.

Decision-making in an open environment requires special means to receive up-to-date information about changes in the environment, including both the objects of the decision and the criteria that influence the decision-making process. In many IISs a decision is a structured set of elements (objects), and these elements have their own structure and can be connected to other elements by various semantic relations. Decision requirements deal not only with such elements, but also with the relations between them. In this case, we have to consider such decisions as *complex information objects* (CIO) [3].

Each CIO is represented as an ordered collection of more than one information object (IO) that are related to each other by relations with defined semantics and meet the requirements for the structure and property values of the IOs defined by the user's task. In each specific case, the CIO structure is determined by the domain knowledge and the specifics of the task. The use of ontologies has become a standard for formalizing domain knowledge in Web-oriented IISs. Therefore, we apply the elements of ontological analysis to describe the model and methods for comparing CIOs.

From the point of view of ontological analysis, any IO is a class instance of domain ontology. Corresponding class of ontology characterizes the IO structure as a set of properties and their characteristics, as well as acceptable relations with other class instances of this ontology. The class instance of ontology associated with an IO also defines values (all or some) of these properties, and some of these properties can define relations with other ontology instances.

Therefore, every CIO can be considered a subset of the domain ontology that is distinguished according to the user's task.

The CIO structure is a set of ontology classes and the relations between them that define the structure of available CIOs and restrictions on the available values of class instances, but it does not contain exact instances.

For example, CIOs that can be generated based on the organizational ontology of a research institute include the team and equipment used to carry out a research project or hold a scientific conference. In this case, decision-making requires the selection of such subset of the institute personnel that can effectively implement the project and has all necessary competencies. Examples of CIOs based on the ontology of an educational institution include the choice of a specialty or the construction of a curriculum to obtain a certain set of competencies. Decision-making in this case is related to obtaining the necessary competencies in the shortest time or with the greatest completeness that can be provided by the selected institution [4].

2. Comparison of complex information objects

In this work we do not consider the problems of decision generation as a sequence of certain actions in a dynamic information environment, but concentrate on other important component of decision-making that deals with comparison of possible solutions represented as CIOs. Our main focus is on conducting a multi-criteria comparison of these CIOs and defining the sources and estimates of these criteria.

A specific feature of our proposed approach is that we only need to compare a relatively small number of CIOs: we analyze not all theoretically possible CIOs that can fit within CIO structure, but only those ones that can be constructed from existing IOs in the current situation.

Therefore, the problem is not finding the optimal (according to certain criteria) solution, but rather choosing an acceptable solution from the set of available options. For example, to carry out a research project, you need to select a group of employees from a specific department, rather than from all individuals in general. Therefore for some situations all possible solutions are unsatisfactory and can change this situation only for the worse. For example, the execution of project by insufficiently competent employees can lead to a loss of time and resources without obtaining of the desired result. At the same time, the significance of the criteria can be changed over time due to changes in the dynamic information environment, and these changes can cause obtaining of the available result on base of the same IOs. The most common example of changing priorities is the cost of work and the speed of obtaining results. For example, the use of some energy sources (such as diesel generators) is not acceptable for routine activity by ecological criterion, but people can use them during blackout because health reasons and safety criteria become more significant.

Comparison of CIO has to provide justification for information search at the meaningful level. Semantic search differs from traditional one by use domain concepts and relations instead of keywords to describe user needs. CIO structure can be considered as such requests. Results of semantic search can be represented not only by list of documents or other IOs but also as structured sets of such IOs that satisfy requirements for components of this structure. Examples of semantic search: to find a group of people with certain qualifications working in the same organization that meets the conditions of the competition; to determine the countries where scientific research on a certain topic was carry out with publishing of the results in a selected set of journals for a certain period of time.

While semantic search, usually, results in a set of IOs of one class (for example, people or goods) or a group of classes with some similar properties (for example, research and educational organizations), where search requirements are used only for selection of acceptable options of IO individuals, CIO comparison is a more complex problem where every result is represented by the set of IOs of different types linked by specified relations. For example, for some task we need a specified set of equipment, qualified personal for operating this equipment, appropriate means for transportation of this equipment and personal to the desired place in a certain time.

Some CIOs can contain several levels of the hierarchy of IOs. For these tasks, we need to generate a criteria hierarchy that determines what criteria should be used for each level of CIO components, and which of them are applicable to IOs of all levels. An example of such task is Hierarchical Aggregate Assessment (HAA) that provides a comparison of teams with several levels of the hierarchy by analyzing individual and collective test results [5]. In HAA, the evaluation process is based on the use of specific tests for each level of the team hierarchy, where each member has an individual hierarchical position. The process of grouping team members into different hierarchical levels and defining their positions within the hierarchy is called aggregation. Therefore, this problem is referred to as aggregated evaluation of the team.

HAA provides assessments of IOs of various levels simultaneously in one assessment session, by interpreting test results differently for each level.

3. Stages of CIO comparison

In the general case, the task of comparing CIOs that have different structures and are based on different ontologies consists of three subtasks:

- alignment of base ontologies and generation of some common ontology;
- analysis of similarities between their structural elements and retrieval of sub-CIOs that can be matched;
- matching of retrieved sub-CIOs with similar structure based on common ontology.

In this work, we consider the last subtask where all compared CIOs are based on a single ontology and have a similar structure. This subtask requires the following stages:

- creation of the CIO reference model of that reflects user requirements (CIO structure);
- generation of a set of existing CIOs that structurally correspond to the reference model, based on information about the current state of the environment and semantic similarity between CIO elements [4];
- choosing among the available CIOs those ones that do not contradict the user's requirements;
- search for CIO evaluation criteria with use of knowledge from the domain ontology;
- determination of the level of individual significance of each criterion at the current moment by the comparison of expert evaluations and domain heuristics with use of the analytic hierarchy process;
- determination of the quantitative assessment of each CIO based on the selected set of criteria and coefficients of their significance.

Comparing CIOs can be seen as a special case of a multi-criteria decision-making (MCDM) problem in recommendation systems [6].

The main steps of the MCDM methodology proposed by Roy [7] can be adapted for the specifics of CIO comparison:

- Defining the decision object involves identifying the CIO structure, which is represented as a set of IOs with relations between them and individuals who require this structure for decision-making purposes.
- Defining a consistent set of criteria involves identifying and specifying a set of functions based on CIO characteristics that represent user preferences among various alternatives.
- Developing a global preference model involves determining the relative significance of partial preferences to define general evaluations of CIOs based on the set of criteria.
- Selecting the decision support process involves choosing the methods of using the selected CIO for a particular task (this step is not analyzed in our proposed work).

3.1. Creation of the CIO reference model of that reflects user requirements

We propose using knowledge from the domain ontology to define the structure and properties of CIO components that reflect user requirements for the decision in question. This structure can serve as a reference model for evaluating available solutions. This process consists of the following steps:

- selecting the domain ontology relevant to the user's task and required solutions;
- analyzing the natural language (NL) description of the user's task and matching it with the domain ontology through various semantic similarity evaluations [8];
- selecting a non-empty subset of ontology classes that correspond with the components of the required solution (we can use approach proposed for generation of the task thesaurus by its NL description [9]);
- selecting a non-empty subset of ontology relations that link the components of the required solution based on task specifics (it is important that such relations have to link all IOs by some chain of relations) according to task specifics that can be made by requests to domain ontology or manually on base of its visual representation in ontology editor Protégé [10];
- describing the properties and their values for CIO components that are important for the solution;
- fixing the CIO structure as an OWL ontology [11].

3.2. Generation and check of the set of CIOs that are relevant to CIO scheme

This subtask involves using combinatorial analysis to first identify all individuals belonging to CIO scheme classes, and then generate all possible combinations of these individuals based on the reference model. For example, we analyze the joint publications of researchers working in the same organization and build the CIO structure of such co-authorship. This structure consists of two components of the class «Person», three of the class «Publications», and one of the class «Organization». We then generate all possible 6-element sequences, where the first and second elements are individuals of the class «Person», the third, fourth, and fifth elements are individuals of the class «Publications», and the sixth element is an individual of the class «Organization». Place of every individual in generated combination is significant, and combinations with the same elements for different positions are considered as different CIOs.

Each individual in the CIO is checked against the user's requirements, which deal with the values of IO properties and relations with other components of the CIO defined by the structure. In example proposed above we search for presence of publication for person and for coauthors of this publication that work in the same organization. We propose sequences that satisfy all these requirements for future analysis. The selected CIOs can be used as solutions to the user's problem, but we must choose only one of them to be implemented. Therefore, we try to find the CIO that is most similar to the reference model described by the CIO structure. For this purpose, we need a method that proposes evaluations of this similarity, taking into account different criteria.

3.3. Generation of the CIOs comparison criteria set with use of knowledge from the domain ontology

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

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

Matching instances of the same class greatly simplifies the task. In such cases, the first group of parameters can be ignored, and the second one does not require aligning the properties of different classes by analyzing 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 comparing CIOs with a different structure is more complex and requires additional stages of information gathering. In this work, we consider the situation of matching CIOs with the same or similar structure, and therefore alignment can be reduced to searching for sub-classes and super- classes 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_{0i} divided in data properties $T\_data_{0i} = \{t\_data_{i_m}, m = \overline{0, y_i}\}$ (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 $T\_o_{0i} = \{t\_o_{i_k}, m = \overline{0, z_k}\}$ (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): $K_{0i} = T\_d_0 \cup T\_o_0$.
- 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: $S_j = \{t \in T_{C\_cl}, f(t, t_j) \le L\}$;

- 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. Additionally, the user can manually add criteria that they consider important, even if they are not represented in the domain ontology or were not included in K_{sem} by the algorithm proposed above. Certain criteria may be lost due to unsuccessfully chosen ontology, inaccurate estimation of semantic closeness, or insufficient number of 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 involves creating a hierarchy of criteria from K_{sem} that represents the user's current needs and their relative importance. This hierarchy can be determined by analyzing the semantic proximity estimates between pairs of CIOs, including those between the evaluated CIOs and the reference CIO, which is built based on 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 the methods used depends on task specifics, user qualifications, and the dynamics of user preferences.

4. Analytic hierarchy process and ontologies

Methods of the multi-criteria problem solving are supported by various approaches (see Figure 1). [12, 13]. One of them is an *analytic hierarchy process* (AHP) that can be improved by integration with use of external domain knowledge. Some researchers that process knowledge on base of ontological analysis try to combine AHP with acquisition of task knowledge from domain ontologies, but mostly they are aimed at creating an ontology that characterizes the main elements of AHP variation or at building an ontological representation of the obtained results. The construction of such AHP ontology ensures the construction of an ontological model of the results of the AHP work: a glossary of basic terms (concepts and their instances, attributes, actions, etc.), a classification tree of concepts and their binary relations.

In proposed here approach we consider elements of domain ontology relevant to solved task that can be used on various stages of AHP application.

The most known method aimed at analyzing hierarchies was proposed by Saati [14]. A large number of improvements to this method and other approaches that achieve similar goals for other variants of the problem statement are now in use. More information with multiple levels of criteria could potentially help to better understand user preferences. Here, we focus on the task of analyzing hierarchies for matching CIOs, which utilizes a domain ontology as a source of knowledge about CIO structure and the relationships between their concepts.

The main task in the construction of the hierarchy is the assessment of higher levels that is based on the interaction of different levels of the hierarchy, and not only on the direct dependences between the elements at these levels. Exact methods of hierarchical construction are used in the natural and social sciences, particularly in the problems of general systems theory related to planning and constructing social systems.

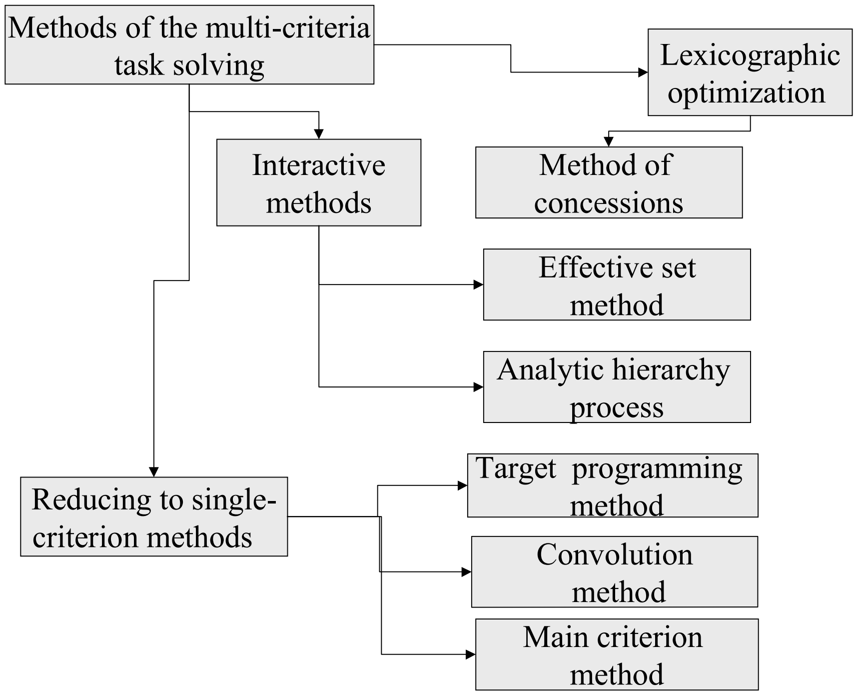

Figure 1: Methods of the multi-criteria task solving

Each element of a hierarchy can functionally belong to several other different hierarchies. For example, the same person may be considered together with other employees of an organization, education participants or patients of a hospital. The same element can belong to high level of one hierarchy as a controlling component, but at another hierarchy simply it can be an element of lower level. For example, a person can be the head of a certain organization as the most authoritative specialist and occupy the highest level of this organizational hierarchy, and in his/her family the same person occupies the middle level of the hierarchy compared to older relatives. Information on the principles of establishing the level of an element in a certain hierarchy is contained into the domain ontology as a set of relations of transitive relations between elements of the corresponding class.

The interpretation of the main elements of AHP depends on the complex tasks it solves and the elements of those tasks. The task of multi-criteria choice (decision-making) for some subject area that is described by the domain ontology is an example of such task, and its interpretation has to take into account the structure of required decision, component of this decision (its elements and relations between them) and more general rules and restrictions of domain on collateral execution of such decisions.

A solution is an instance of a class of ontology P, for which a set of semantic properties $C_1, C_2, ..., C_n$ is defined in this ontology. These properties can be both object properties (with values from instances of other classes) and data properties (with quantitative, qualitative constant values of various types). Additional information from P can be used in $C_1, C_2, ..., C_n$ by considering of semantically similar (from the task point of view) domain concepts.

$C_1, C_2, ..., C_n$ is a subset of the domain ontology concepts of the that are considered as criteria for evaluating the compliance of instances of class P with the defined goal. Therefore values of all properties from $C_1, C_2, ..., C_n$ should be evaluated in the range from "completely unacceptable" to "very successful".

In this case, the AHP is used to determine the hierarchy of individual decision-making criteria, which are compared (evaluated) in pairs by one or more domain experts.

Decision-making consists in selection of the most suitable solution among a limited non-empty set X instances of class P: $X \subseteq P$.

The set X is created based on external criteria that are outside the scope of the AHP approach, such as purchase price, availability of access, or time interval of use. It should be taken into account that in most practical problems of CIO matching set X is a very small subset of P, and therefore there is no need to search for generalized rules for classifying instances of X by AHP categories “completely unacceptable”, “acceptable”, “very successful”, etc., as is done in various methods of machine learning (ML).

Rather than a general classification, we aim to solve a problem with lower computational complexity by using a traductive rule “from the individual to the individual” instead of an inductive rule “from the individual to the general”.

In addition, unlike MN, where instances of P are classified according to an arbitrary set of categories without formalized semantic relations between them (for example, a recommended position or a probable diagnosis), instances of X in AHP belong to an ordered set of categories with a fixed number of values and can be connected by various relations. The main differences between ML and AHP are represented in the table 1.

Table 1

Comparison of the main characteristics of AHP and MN

	AHP	MN
The object of analysis	Properties of the domain objects and their meaning	Properties of the domain objects and their meaning
Data for analysis	Expert evaluations of properties	Information on instances of domain objects of a certain class and their characteristics
Result of analysis	Hierarchy of significance of the properties of the domain objects	Classification rule for instances of the domain objects of the selected class
Representation of analysis results	Weight vector of dimension n	Decision tree, linear or non-linear expression, neural network, etc.
Range of definition	Ordering of a fixed subset of instances of objects of the selected class according to their compliance with the goal	Attribution of all instances of objects of the selected class to one of the categories
Computational complexity	Depends on the dimension of C and the volume of X	Depends on the dimension of C and the volume of P
Complexity of the decision space	An ordered set of evaluations for fixed decision class	An arbitrary set of non-overlapping categories to which instances of the decision class can be assigned
The role of domain ontology	Source of C and structure of class P	Source of C and structure of class P, their characteristics

Suppose a group of experts considers n types of actions or objects and has two goals: 1) to make statements about the relative significance of these objects, and 2) to develop a process of statement obtaining that allows for quantitative evaluations of these statements for all objects.

It is clear that achieving the second goal requires developing a suitable method for obtaining a set of weights associated with individual objects from the quantitative judgments of the expert group, which are derived from the relative values associated with pairs of objects.

This approach transforms qualitative information into a quantitative form that is more convenient for processing without losing any information.

Let $C_1, C_2, ..., C_n$ is a set of objects (or possible actions). Quantitative statements about pairs of objects $(C_i, C_j), i = \overline{1,n}, j = \overline{1,n}$ are represented by the matrix A of $n \times n$ size: $A = (a_{ij}), i = \overline{1,n}, j = \overline{1,n}$.

The elements of matrix A are determined according to the following rules:

1. if $a_{ij} = x$ and $x \neq 0$, then $a_{ji} = 1/x$;

2. if C_i and C_j have the same relative importance, then $a_{ij} = a_{ji} = 1$ (, $\forall i = \overline{1,n} : a_{ii} = 1$).

Thus, this matrix A has the following form:

$$A=\begin{bmatrix} a_{11}=1 & \dots & a_{1i} & \dots & a_{1n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{i1}=1/a_{1i} & \dots & a_{ii}=1 & \dots & a_{in} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1}=1/a_{1n} & \dots & a_{ni}=1/a_{in} & \dots & a_{nn}=1 \end{bmatrix}$$

After obtaining quantitative estimates for all pairs (C_i, C_j), the task is reduced to matching n possible actions $C_1, C_2, ..., C_n$ with a set of n quantitative estimates $w_1, w_2, ..., w_n$. For this, it is necessary to more clearly formalize the problem in terms of an abstract mathematical structure.

Therefore it is desirable to describe the main stages of the process of task formulation and describe each stage in more detail, so that the potential user can assess the feasibility of its application to a certain practical problem. We have to define clearly how the weights $w_i, i=\overline{1,n}$ depend on expert opinions $a_{ij}, i, j=\overline{1,n}$. We propose to separate three steps of task formalization process, from the simplest individual case to the most general situation.

Step 1. If statement $a_{ij}, i, j=\overline{1,n}$ are the results of a comparison of exact measurements $w_1, w_2, ..., w_n$, then the estimate $a_{ij}=w_i/w_j$, $a_{ji}=w_j/w_i$, , , i.e.

$$A=\begin{bmatrix} w_1/w_1 & \dots & w_1/w & \dots & w_1/w_n \\ \dots & \dots & \dots & \dots & \dots \\ w_1/w_1 & \dots & w_i/w_i & \dots & w_i/w_n \\ \dots & \dots & \dots & \dots & \dots \\ w_n/w_1 & \dots & w_n/w_i & \dots & w_n/w_n \end{bmatrix}$$

But for most practical problems, such a situation is unattainable, because even in the presence of direct measurements of the values these values are not accurate and have certain deviations, which are determined by the statistical dispersion of the measurement results.

Step 2. Determination of deviations in values can be defined as the average value for all measurements, that is:

$$w_i=\frac{1}{n}\sum_{j=1}^{n} w_j a_{ij}, i=\overline{1,n}.$$

But even such relaxation of conditions raises the questions about the existence of a unique solution to the definition of $w_1, w_2, ..., w_n$ for the given $a_{ij}, i, j=\overline{1,n}$.

As it follows from the above remarks, the proximity of a_{ij} is determined by the value of n. Thus, for a more generalized case, it is necessary to estimate such a dependence.

Step 3. We define by l_{max} the largest value of n for which $w_i=\frac{1}{l_{max}}\sum_{j=1}^{n} w_j a_{ij}, i=\overline{1,n}$ have a unique solution (eigenvalue problem).

Then we can find $a_{ij}, i, j=\overline{1,n}$ for this value. If analyzed data contains more then l_{max} compared elements we can consider the found solution is the unique one.

To obtain a quantitative assessment of matched objects, the matrix A must be multiplied by the transposed vector of the characteristic weights. The sum of the assessments for each of the eigenvectors obtained above is then determined, taking into account the priority of the corresponding characteristic.

The main difference between the proposed variant of AHP and traditional ones is the use of external knowledge about the domain. This knowledge does not replace expert assessments but provides them with a more structured initial set of information. For example, experts do not need to manually enter the basic elements of the domain (concepts, relations, etc.) that can be obtained from the corresponding ontology. Generating a set of examples that satisfy certain user requirements is easier with semantic queries to the ontology than having experts select them directly from existing examples. It is important to note that ontological information alone is not the basis for making decisions regarding the hierarchy of criteria. This information only becomes relevant after experts confirm its relevance and pertinence to the task.

5. Practical example of the multi-criteria comparison of CIOs

Proposed approach to matching of CIOs can be used for generation of specialized subgroups of heterogeneous unmanned aerial vehicles (UAVs). Today, UAVs, or drones are used in various fields of application – both military (surveillance and attack) and civilian, as they are suitable for solving a large the number of tasks, and the reduction of their cost, various technical solutions and the expansion of the management functionality make their use expedient. UAVs are effective in dynamically uncertain environments with hard- to-reach areas. However, in such cases, they usually require special sensors to facilitate the task. Managing the coordination of drones is a complex task that requires integrating research from multiple fields [15].

Use of UAVs is very urgent and increases interest in both military, industrial and research fields due to the large number of scenarios and applications they can support and execute in dynamic/uncertain environments. One of the great challenges of today is the application of UAV swarms to take advantage of the benefits that the coordinated actions of drones [16]. The concept of "swarm of drones", which the American Institute of Modern Warfare associates with the technology of swarms of mass destruction originally developed for ballistic missiles, can significantly increase the efficiency of UAVs. But management of such complex system need in it division of such devices on subparts with selected set of functions. Each UAV can be considered as a separate intelligent agent (IA) [17] according to approach of agent-oriented programming (AOP) [18] that has specific goals, resources and strategies. Action planning and environmental assessment by drones can use elements of artificial intelligence (AI). We can analyze the joint activity of a group of UAVs as a multi-agent system (MAS) [19] where individual rational agents of individual drones can exchange information and perform a joint task, optimally allocating goals.

MASs solve problems actual for organization of the drone swarm working together:

- *coordination*: agents have to make joint decisions to cooperate effectively and achieve a common goal;
- *communication*: agents need in information exchange to learn about the state of the environment and make decisions;
- *trust*: often agents do not have complete information about the state of the environment and have to use information provided by other agents;
- *ambiguity*: agents often have to make decisions under conditions of ambiguity and unpredictability;
- achieving a *common goal*: common goal can contradict local goals of individual agents, and need in means of their harmonization.

However, there are various methods and technologies that help solve these problems and improve the performance of multi-agent systems. For example, coordination techniques such as joint planning can be used to help agents coordinate their actions and make decisions together. Communication methods, such as formalized information exchange protocols, can also be used to improve the transfer of information between agents. To deal with ambiguity, techniques such as modeling and uncertainty assessment can be used to give agents a better understanding of the state of the environment.

For example, as a result of such information exchange, two drones can attack different objects to inflict maximum damage, or vice versa, jointly attack one object for its guaranteed damage. At the same time, the selection of objects is carried out dynamically based on built-in rules and analysis of the current state.

To facilitate the cooperation of drones into the swarm, they need to share information with an unambiguous understanding of it, but the heterogeneity of the information representation used by different UAVs is a major obstacle. Therefore, we propose to use ontology of pertinent domain to solve information heterogeneity and allow UAVs to use information equally in the interaction process.

In some cases, the use of a single UAV is not enough for task that need in joint use of several drones with different functionality or in different places. The use of UAV swarms that group of individual vehicles in a group with limited human intervention for common goal can be used for division and paralleling of task.

The use of UAV groups provides a number of advantages: 1) the total cost of acquiring and maintaining several small commercial UAVs is lower than the total cost of one large unit; 2) scalability is a key feature of UAV swarms; 3) increased fault tolerance, since the malfunction of one drone has a limited effect on the swarm; 4) faster operations thanks to the distribution of work. But the organization of the interaction of UAV groups requires the creation of appropriate theoretical models of their communication and interaction and the software implementation of algorithms that ensure decision-making by individual UAVs based on the available information and taking into account the common goal.

First step of such organization requires generation of UAV teams as subgroups from existing ones according to actual goals that are represented as a set of requirements for all drones into group, for at least one (or other number) of group elements and for combination or disjointness of some properties. Such problem definition is very similar to matching of CIOs with reference one.

Using drone teams for a specific task requires effective communication and coordination between groups of UAVs and individual drones.

Drones are grouped both by purpose (and existing competencies) and by low-level subordination into territorially close groups focused on the execution of separate subtasks.

Personal agents of devices can be classified by the hierarchical set of criteria: for example, by functionality, by manufacturers or with a similar value of domain-specific parameters (flight range, weight, availability of certain devices, functions or weapons). Another criterion for the construction of subgroups deal with cooperation where the subgroup has to contain some fixed set of functions provided by different devices: for example, one drone has advanced surveillance tools and several another ones can carry larger load.

For more effective UAV management, it is advisable to support several levels of group hierarchy that provide (by analogy with the reference model of open systems) interaction of drones either with devices of their own level of hierarchy (members of their micro-group), with devices one level higher (direct commanders or coordinators) or with devices one level below (direct subordinates). Elements of AHP can be used to choose the most successful structure of the MAS.

In order to determine the functionality of individual devices (both the drones themselves and their equipment and weapons), it is advisable to develop an ontological model of their competencies, where various potentially acceptable elements of functionality are characterized both qualitatively (availability, type of image perception, etc.) and quantitatively (speed, weight, range, etc.). In addition, this ontology should reflect the semantics of interaction between individual types of drones, operators and external equipment. The apparatus of atomic competences developed for the assessment of learning outcomes and education planning [3] can be easily adapted to describe the available technical characteristics of independently operating drones and directions for their improvement (see Table 2).

Table 2
Correspondences of competence approach with UAV management parameters

Competence approach	UAV management
Potential employees	Individual drones
Available staff	UAV swarm
Project team	UAV subgroup
Employee competencies	Drone equipment
Ability to learn additional competencies	Possibility of completion with other equipment
Set of vacancies for the implementation of the project	task to be performed by a group of drones
Atomic competency	UAV equipment property
Matching of competencies	Comparison of UAV subgroups

This approach provides possibility of evaluation and comparison of different UAV sets and their compliance with actual tasks and environment parameters.

UAV swarm tasks often require multi-criteria decision-making.

For example of such task is organization of defense of a critical infrastructure object with the help of UAVs (drones) against enemy air raids. There is a network of drones with various functions, such as air defense equipment and monitoring equipment for the airspace and ground area around the object to detect mobile enemy objects, communication means. We need to find a solution for selecting the minimum number of drones that can ensure an appropriate defense level for the object.

Here we don't consider planning of drone actions and distribution of function because we analyze the previous stage where we have to generate this swarm subset that would be able to solve practical task, and elements of this task are considers as criteria of group generation. Examples of these criteria are protection of different types of dangers for the object of critical infrastructure, potential damages of enemy (live military forces, mobile equipment, weapons, etc.) and own losses of UAV. However, simply listing the criteria is not sufficient. We also need to evaluate their relative significance. For example, safety of drones is important and necessary condition for future defense, but some objects (such as nuclear power plants) are so critical that their safety is much more significant.

Therefore, we are dealing with a multi-criteria problem that requires the simultaneous optimization of conflicting objectives. It is impossible to find a solution that would be the best for all criteria at the same time, because in general the improvement of the values of some criteria subset leads to the deterioration of the values of the other ones and solution can only be a compromise solution

6. Conclusion and prospects for further work

Comparing CIOs with similar structures is a necessary component of the more general problem of analyzing and comparing CIOs built on different ontologies and having different structures. When analyzing CIOs with heterogeneously defined semantics, it is necessary to: 1. align the basic domain ontologies and identify correspondences between their concepts and relations; 2. identify subsets of CIOs with similar structures in the compared CIOs; 3. compare these subsets using the algorithm discussed above.

The approach based on hierarchical aggregate evaluation of CIOs focuses on integrated multi-criteria decision-making in conditions where the set of criteria depends on the specifics of the task and can be created based on domain knowledge. However, the relative importance of these criteria can change dynamically at different points in time. We intend to expanded this approach in the future with the use of tools and methods such as knowledge management, intelligent data analysis, and machine learning, which can be employed to obtain competent knowledge.

The theoretical models and methods proposed in this work can be used to support tasks relevant to the state of war, such as risk management, rapid adjustment of industry for the production of essential products, restorative construction, and dynamic adaptation of teams with a multi- level hierarchical structure (military units, expert commissions, rapid response medical teams) to perform critical operational tasks in the absence of sufficient competencies, skills, and experience [5]. This approach enables the identification of strengths and weaknesses of various teams and their members for specific tasks, facilitating adjustments to their composition or additional training as needed.

In the future, we propose to use matching of complex objects with a hierarchy of criteria as a scaling instrument. This approach transforms complex tasks into a set of simpler subtasks linked by various hierarchical relations. Existing standards and domain ontologies used in the semantic structure require additional research on alignment methods.

7. References

[1] N. Guarino, P. Giaretta, P. Ontologies and knowledge bases. In: Towards very large knowledge bases, 1-2, 1995.

[2] P. Fitsilis, V. Gerogiannis, L. Anthopoulos, Ontologies for Software Project Management. In: Journal of Software Engineering and Applications. No. 7, 2014, pp. 1096-1110. doi.org/10.4236/jsea.2014.713097.

[3] J. Rogushina, S. Pryima, Ontological approach to qualification matching based on competences: model and methods. In: Naukovyi visnyk NGU. No. 6, 2017, pp.162-168. URL: http://elar.tsatu.edu.ua/bitstream/123456789/3194/1/4.pdf.
[4] J. Rogushina, A. Gladun, Development of domain thesaurus as a set of ontology concepts with use of semantic similarity and elements of combinatorial optimization Journal of Problems in programming. No. 2, 2021, pp. 03-15. doi.org/10.15407/pp2021.02.003/.
[5] M.Lesage, G., Raîche, M. Riopel, F. Fortin, D. Sebkhi, An E-assessment website to implement hierarchical aggregate assessment. In: World Academy of Science, Engineering and Technology, International Science Index 86, 2014, 8(2), pp.925-933.
[6] G. Adomavicius, N. Manouselis, Y. Kwon. Multi-criteria recommender systems. In: Recommender systems handbook, 2011, Springer, Boston, MA. pp. 769-803. URL: https://www.academia.edu/download/32978074/Recommender_systems_handbook.pdf#page=774. doi.org/10.1007/978-0-387-85820-3_24.
[7] B. Roy, R. Słowiński, Questions guiding the choice of a multicriteria decision aiding method. In: EURO Journal on Decision Processes, 2013, 1(1-2), pp.69-97. doi.org/10.1007/s40070-013-0004-7.
[8] P. Resnik, Using information content to evaluate semantic similarity in a taxonomy. In: Proc. of the14th international joint conference on Artificial intelligence, volume 1, 1995, pp.448-453.
doi.org/10.48550/arXiv.cmp-lg/9511007.
[9] J. Rogushina, A.Gladun, Task Thesaurus as a Tool for Modeling of User Information Needs. In: New Perspectives on Enterprise Decision-Making Applying Artificial Intelligence Techniques, Springer, Cham. 2021, pp. 385-403. doi.org/10.1007/978-3-030-71115-3_17.
[10] A free, open-source ontology editor and framework for building intelligent systems. URL: protege.stanford.edu.
[11] OWL Web Ontology Language. Overview. W3C Recommendation. URL: www.w3.org/TR/owl-features/.
[12] R.E. Steuer, Multiple Criteria Optimization: Theory, Computations, and Application. New York: John Wiley & Sons, Inc., 1986. ISBN 047188846X. doi.org/10.1002/oca.4660100109.
[13] O. B.Vasyliev, N. S.Vasylieva, O. D.Kichmarenko, Methods of solving multi-criteria optimization problems: tutorial. Odesa: Odesa National University named after I. I. Mechnikov, 2017. 48 p. URL: http://dspace.onu.edu.ua:8080/bitstream/123456789/22586/1/VecOpt.pdf.
[14] T. L. Saaty, Decision making with the analytic hierarchy process. In: International journal of services sciences, 2008, 1(1), pp.83-98.
[15] M. W. Mueller, S. J. Lee, R. D'Andrea, Design and control of drones. In: Annual Review of Control, Robotics, and Autonomous Systems, 5, 2022, pp.161-177. doi.org/10.1146/annurev-control-042920-012045.
[16] Y. Shi, Particle swarm optimization. In: IEEE connections, 2004, 2(1), pp.8-13. doi.org/10.1109/CEC.2001.934374.
[17] M. L. Griss, Software agents as next generation software components. In: Component-based software engineering, 2001, pp.641-657. URL: https://pdfs.semanticscholar.org/839e/b7a89ed502b637fc6ad5a46acd4dca0cad40.pdf.
[18] Y.Shoham, Agent-oriented programming. In: Artificial Intelligence. N 60. 1993, pp.51-92.
[19] M. L. Griss, Software agents as next generation software components. Component-based software engineering, 2001, pp.641-657. URL:
https://pdfs.semanticscholar.org/839e/b7a89ed502b637fc6ad5a46acd4dca0cad40.pdf.
[20] J. Rogushina, A. Gladun, Ontology-based competency analyses in new research domains. Journal of Computing and Information Technology, 2012. 20 (4), pp. 277-291. URL: http://www.scopus.com/inward/record.url?eid=2-s2.0-84878940246&partnerID=40&md5=38cda0f803a838713e8d3d88262604c8.
doi.org/10.2498/cit.1002034.