The object of this study is the IT project configuration management process.

During the research, the problem of early identification of configuration items (CI) in the information system (IS) of enterprise management was solved. Research in this field is mainly aimed at solving the task of early identification of services and microservices during the refactoring of software systems. The issue of the application of artificial intelligence methods for the detection of CI has not been sufficiently investigated.

During the study, the Chameleon hierarchical clustering method was adapted to solve the problem of early identification of CI IS. This method takes into account both the internal similarity and the connectivity of individual functions of the studied IS.

The adapted Chameleon method was used when solving the task of early identification of CI in the functional task "Formation and maintenance of an individual plan of a scientific and pedagogical employee of the department". 10 functions and 12 essences of the problem database were considered as the initial CIs. The result of the solution is a dendrogram with all possible options for decomposition of the description of the task architecture into individual CIs.

Based on the results, a comparative analysis of the use of Chameleon, DIANA, and AGNES methods for solving the problem of early identification was carried out. According to the criteria "Number of vertices of the dendrogram", "Number of levels of decomposition of the dendrogram", and "Evenness of filling the elements of the dendrogram", the results from using the Chameleon method are the best.

Using the research results allows automating the procedure of forming backlogs of IT project implementation teams. This makes it possible to improve the quality of IS development by assigning IS containing similar functions to the same IT project executor

Keywords: information system, configuration item, hierarchical clustering, Chameleon method, Chebyshev distance

UDC 004.65; 004.91

DOI: 10.15587/1729-4061.2024.303526

COMPARISON OF SOLUTIONS TO THE TASK OF IT PRODUCT CONFIGURATION ITEMS EARLY IDENTIFICATION USING HIERARCHICAL CLUSTERIZATION METHODS

Maksym Ievlanov
Corresponding author
Doctor of Technical Sciences, Professor*
E-mail: maksym.ievlanov@nure.ua
Nataliya Vasiltcova
PhD, Associate Professor *
Iryna Panforova
PhD, Associate Professor *
Borys Moroz
Doctor of Technical Sciences, Professor**
Andrii Martynenko
PhD**
Dmytro Moroz
PhD**
*Department of Information Control System
Kharkiv National University of Radio Electronics
Nauky ave., 14, Kharkiv, Ukraine, 61166
**Department of Software Engineering
Dnipro University of Technology
Dmytra Yavornytskoho ave., 19, Dnipro, Ukraine, 49005

Received date 14.03.2024
Accepted date 25.04.2024
Published date 28.06.2024

***How to Cite**: Ievlanov, M., Vasyltsova, N., Panforova, I., Moroz, B., Martynenko, A., Moroz, D. (2024). Comparison of solutions to the task of it product configuration items early identification using hierarchical clusterization methods. Eastern-European Journal of Enterprise Technologies, 3 (2 (129)), 20–33. https://doi.org/10.15587/1729-4061.2024.303526*

1. Introduction

The configuration management process is one of the important processes in the life cycle of an IT product. Although this process is recommended to be attributed to the processes of integrated change control [1], it is no less important for other aspects of managing the organization and its IT services [2]. The purpose of this process is to systematically control configuration changes, maintain integrity, and track configuration throughout the entire product life cycle [1].

The process of configuration management is proposed to be considered as a set of separate works and tasks that are solved during the execution of these activities. However, in different sources, the names, purpose, and description of these operations and tasks often do not coincide. Thus, in [1], the following are indicated as the main tasks of the configuration management process: planning and management of the configuration management process; configuration identification; configuration control; configuration state accounting; configuration audit and product release and delivery management. In [3], the main tasks of this process are indicated: configuration management planning; configuration definition; configuration change management; configuration status accounting; configuration assessment; release control. At the same time, regardless of the variants of the descriptions, the task of identifying the configuration

related to the definition of system elements that are objects or elements of the configuration ("Configuration item", CI) [1–3] is highlighted in the process. Configuration identification gradually establishes and maintains a current basis for monitoring and accounting for the state of an IT product and its CI throughout their entire life cycle [2].

Despite the rapid development of configuration management concepts and tools, the solution to the problem of configuration identification is described mainly at the conceptual level [2]. The reason for this is the established view of CI as a collection of hardware, software, or both. At the same time, conflicting requirements arise for the problem of configuration identification. On the one hand, such a task must cover every element of hardware and software. On the other hand, CIs should be marked at the highest possible level (it is recommended to use as few CIs as possible in the final IT product) [2]. The consequence of this is the emergence of a large number of theoretical and applied problems related to the allocation of the optimal number of CIs in the course of solving the problem of configuration identification. The optimal number of CIs should be understood here as the number of CIs that will require the minimum expenditure of time and resources to carry out further work on the configuration management process of the IT product. Therefore, conducting research in this field should be considered theoretically and practically promising.

2. Literature review and problem statement

The results of contemporary research allow us to state that the theoretical and applied principles of IT product configuration management systems have been generally formed [2, 4]. These bases in [4] are proposed to be represented as a combination of the following three main groups:

– theoretical foundations (in particular, the three-component model);

– best practices (in particular, ISO 20000 and ITIL, as well as their analogs);

– innovation (in particular, DevOps integration, infrastructure as code, containerization technologies).

Theoretical foundations, according to [4], form a fundamental understanding of configuration management. Best practices provide a fairly efficient solution to the task of identifying, controlling, and accounting for CI status. The considered innovations improve the traditional methods of IT product configuration management, providing scalability, automation, and adaptability of application solutions. However, such a representation leaves a number of problems unsolved. Such issues include security problems, issues of compliance with requirements for the IT product, and difficulties arising during the implementation of various options for cooperation and interaction of the IT product with the surrounding world [4].

Solving the task of compliance with IT product requirements is largely determined by the existing concept of research and formal description of service-oriented systems [5]. Thus, according to this concept, the solution to most problems of choosing IT services requires the definition of a set of functionally equivalent IT services before starting to solve similar problems [5]. If we consider IT services as CIs, then this statement can be formulated as follows: the task of identifying IT services as CIs must be solved first for abstract descriptions of these CIs. Such abstract descriptions should not depend on the specificity of the implementation of these services and the non-functional requirements put forward to the services.

The statement discussed in the previous paragraph becomes especially important when managing the configuration of large systems (the so-called System of Systems). An example of such systems is information systems for enterprise and organization management (ISEM). Thus, paper [6] shows the possibility of eliminating most cases of negative impact of local solutions on the overall design and quality of a large software and technical system due to the early division of the system into separate elements.

The approach to functional decomposition of the system architecture description into individual elements is described in [7]. At the same time, the task of identifying the configuration is solved in two stages:

– at the first stage, the functional decomposition of the description of the architecture into separate elements is carried out, taking into account the necessary evolutionary changes;

– at the second stage, those decomposition options are selected that satisfy the restrictions on the cost of the necessary changes.

However, the approach considered in [7] is mainly focused on tracking changes made to the architecture description. At the same time, the following questions remain unresolved:

– the issue of the initial identification of individual functions as CI as a result of the analysis of points of view on the architecture of this system;

– the issue of selecting individual CIs from the formed description of the system architecture as backlogs of the teams implementing the IT project for the creation of this system.

In [8], the issue of selecting individual software artifacts that enable the reliable operation of the designed software system is considered. At the same time, it is proposed to use the description of the meta-architecture of such a system as the basis for such selection. It is emphasized that the use of abstractions to describe the architecture of a software system simplifies the solution to such a problem. However, selection from the description of the CI system architecture at the level of individual system functions is not considered in [8].

The task of dividing the description of the system architecture into some microservices is considered in [9]. To describe the overall system architecture, it is proposed to use the object-oriented language Silvera. However, the application of this solution is limited by the following features [9]:

– focusing exclusively on the decentralized development of microservices;

– the absence of a description of the business logic in the description of the system architecture.

The lack of a solution to the problem of early CI identification by decomposing the system description into small elements can be explained by the fact that there are probably no universal approaches to solving this problem. Some likely confirmation of this statement is study [7]. It shows that most of the existing approaches to the decomposition of a monolithic architecture into a set of individual microservices can be applied only under certain conditions. However, these conditions are not specified in [7].

An attempt to specify similar conditions for the problem of early identification of a CI system was made in [6]. In particular, it was shown that the problems of early division of the system into separate elements arose mainly as a result of misinterpretation of requirements or bias of personal experience. Therefore, it is advisable to use artificial intelli-

gence methods to identify CI of large systems, in which the subjective influence of an individual analyst is minimized.

One of the modern directions of application of artificial intelligence methods for solving the problem of early identification of system's CIs involves the use of machine learning methods. In [10], examples of application for modeling and setting functions of recommender systems and machine learning methods are given. This approach to solving this problem is conditioned in [10] by the fact that a complete enumeration of all possible configurations is impossible and may cause serious performance problems. And although the considered examples take into account the factor of variability in the descriptions of individual functions, the issues of connectivity of these functions remain unresolved.

In [11], it is proposed to use a sampler based on deep learning with reinforcement to solve the problem of choosing a system configuration. This sampler allows for the efficient identification of a minimal subset of configurations that covers all possible interactions of features while minimizing redundancy. However, such a solution is attractive only for loosely connected ISEMs designed to manage unstable processes of enterprises and organizations. For ISEM designed to manage stable processes (processes whose data flows are resistant to changes), such a solution is excessively complex.

Another direction that offers simpler solutions is the study of the application of clustering methods to solve the problem of early CI identification. Thus, in [12], the application of clustering algorithms to solve the problem of microservices selection during refactoring of the source code of a monolithic software application is considered. It is shown that a similar method of solving this problem produces more connected microservices with fewer intermediate interactions. A similar situation is observed for cases of refactoring of complex multi-level monolithic software systems during their decomposition into microservices [13]. However, such options for the selection of microservices do not foresee the possible dependence of microservices on the functions selected in these software products. It is assumed that the functional decomposition of the description of the architecture of the original software system was carried out during its design and changes during refactoring.

The study of the application of clustering methods to solve the problem of early identification of backlogs of executive teams as system's CIs was considered in [14]. However, to solve this problem, the simplest method of divisional clustering was used, which did not take into account the relationship between the identified functions.

Our review of the literature [4–14] allows us to conclude that there is a need to conduct research in the field of clustering methods that take into account the connectivity of the studied functions, to solve the problem of early identification of these functions as Cis of an IT project for constructing ISEM. These studies will allow us to highlight the advantages and disadvantages of using similar methods in comparison with other common clustering methods.

3. Literature review and problem statement

The purpose of our study is to determine the possibility of early identification of the CIs of ISEM using the clustering method, which takes into account the connectivity of the investigated functions within the ISEM. As an example of such a method, it is proposed to apply the Chameleon hierarchical clustering method. The application of this method makes it possible to form separate backlogs for teams implementing the IT project to create the system, taking into account the connections discovered during the formation and analysis of the functional requirements for this system. This method for solving the problem of early identification of CIs of the information system makes it possible to enable the delivery of ISEM releases in which the management of individual processes and works of the enterprise will not be impaired due to the breakdown of such connections.

To achieve the goal, the following tasks were set:

– to adapt the Chameleon method to the peculiarities of solving the task of early identification of Cis in ISEM;

– to check the possibility of solving the task of early identification of CIs in ISEM using the Chameleon method;

– to conduct a comparative analysis of the results with the results from solving the task of early identification of ISEM CIs by other methods of hierarchical clustering.

4. The study materials and methods

The object of this study is the configuration management of ISEM. The subject of the study is the task of early identification of ISEM CIs.

The main hypothesis of this study assumes the possibility of improving the solution to the problem of early identification of ISEM CIs using the Chameleon hierarchical clustering method in comparison with the solutions to the same problem obtained using other methods of hierarchical clustering.

The choice of a group of hierarchical clustering methods is due to the following considerations [14]:

– due to the application of these algorithms, each CI will belong to only one cluster at each specific level of the cluster tree;

– the application of these algorithms does not impose restrictions on the form of the final clusters;

– the result of the application of these algorithms is a tree of cluster associations, on the basis of which the system architect can choose the final solution, taking into account factors that are weakly formalized.

However, hierarchical and, in particular, agglomerative methods of clustering have a number of disadvantages, among which the following were highlighted in [15]:

– these methods do not use information about the nature of individual merging clusters;

– one group of these methods (CURE and related to it) ignores information about the cumulative relationship of elements in two clusters;

– another group of these methods (ROCK, group averaging method and related ones) ignore information about the proximity of two clusters, which is determined by the similarity of the nearest elements in two clusters.

To eliminate these shortcomings, Chameleon agglomerative clustering method was developed in [15]. The choice of the Chameleon clustering method is also due to the fact that it has greater capabilities for detecting clusters of arbitrary form of high quality than other hierarchical clustering methods (for example, such as BIRCH and DBSCAN) [16].

The generalized scheme for using the Chameleon method to search for clusters on the data set is shown in Fig. 1 [15].

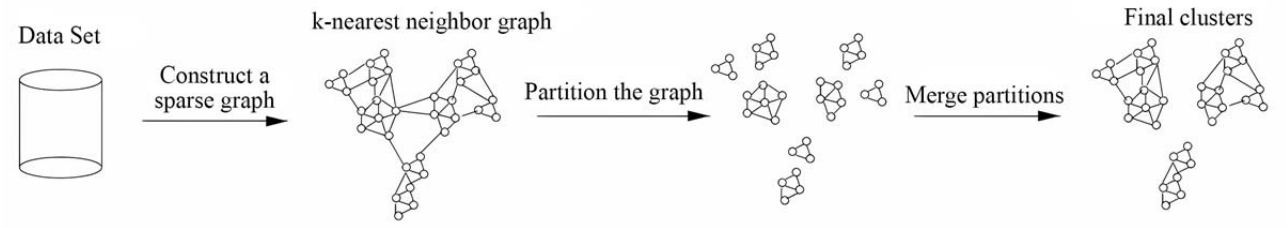

Fig. 1. A generalized scheme for using the Chameleon method to find clusters on a data set

In [15] it is proposed to consider the Chameleon method as a sequence of two stages directly related to the processing of the graphical representation of the analyzed data set. But a generalized approach to solving the problem of clustering using the Chameleon method, illustrated in Fig. 1, considers it appropriate to represent this method as a sequence of the following three stages:

– Stage 1: construction of a graph by adding edges according to the principle of nearest neighbors;

– Stage 2: selecting a set of relatively small connected subgraphs on the graph constructed in Stage 1 [15];

– Stage 3: agglomerative formation of a set of clusters; as the initial data, the set of subgraphs obtained in Stage 2 [15] is taken.

Stage 1 can be represented as a sequence of the following steps:

– Step 1: form a matrix of distances between objects of multidimensional space, the analysis of which is carried out;

– Step 2: set the value k for further search of k nearest neighbors of each of the objects of the multidimensional space;

– Step 3: determine on the distance matrix k the smallest distances from each object to all other objects of the multidimensional space;

– Step 4: form subgraphs that describe the connectivity of each of the objects of the multidimensional space with its k nearest neighbors;

– Step 5: Combine the subgraphs obtained in Step 4 into a final graph or set of graphs and complete Step 1.

The result of Stage 1 is a graph or a set of graphs that describe the investigated points of the multidimensional space. The number of graphs depends on the value of k, which is set by the analyst in the process of determining the nearest neighbors. The larger the value, the more likely it is that the result will be a single graph rather than multiple unconnected graphs.

Stage 2 can be represented as a sequence of the following steps:

– Step 1: determine the weight of each edge of a graph or a set of graphs, which is the result of Step 1;

– Step 2: determine the graph or subgraph with the largest number of vertices;

– Step 3: divide the graph or subgraph defined in Step 2 into two subgraphs according to a predetermined condition;

– Step 4: determine the number of vertices in each subgraph that are the result of Step 3; if this quantity is equal to the predetermined value, then stop the further division of these subgraphs;

– Step 5: if there are any graphs or subgraphs that require further partitioning, then return to Step 2. Otherwise, form a set of clusters, each of which includes one of the subgraphs whose partitioning was stopped in Step 4, and complete the execution of Step 2.

The weights of the edges of a graph or set of graphs, which are the result of Step 1, are defined as density characteristics of the region in which the nearest neighbors of each of the objects of the multidimensional space are defined. If these areas have a high density and the neighbors are located close to the object under study, then the weight should be large. If these areas are sparse, then the weight should be small. The physical meaning of the weight of the edge in this case is to quantify the degree of similarity of the vertices that describe the neighboring objects of the multidimensional space.

The condition under which a graph or subgraph is divided into two new subgraphs at Step 3 of Stage 2 is the simultaneous observance of two simple conditions:

– the separator of the edges of the graph or subgraph, the division of which is carried out, must be minimal;

– each of the subgraphs resulting from the division must contain at least 25 % of the vertices of the original graph or subgraph.

The edge separator is the number of edges, the destruction of which divides the investigated graph or subgraph into two subgraphs that are not connected to each other.

To stop dividing a graph or subgraph into separate subgraphs in Step 4 of Stage 2, it is necessary to determine in advance the number of vertices that should remain in the new subgraph. Usually, this number is equal to the number that is in the range of 1...5 % of the initial number of objects in the multidimensional space.

The result of Stage 2 will be a set of clusters, each of these clusters containing a minimum number of elements. These elements are relatively small connected subgraphs that do not intersect with each other.

The sequence of steps that can be used to represent Stage 3 is generally determined based on the result of the selection of the agglomerative clustering strategy. Two such strategies are defined in [15]:

– the first strategy offers the merging of two clusters if their indicators exceed the set threshold values;

– the second strategy proposes merging two clusters if their indicators maximize the benefit function.

Let's consider these strategies in more detail. The first strategy requires user-defined T_{RI} and T_{RC} thresholds. Then the decision to merge two adjacent clusters C_i and C_j is made if the following conditions are met for these clusters [15]:

$$RI\left(C_i,C_j\right)\geq T_{RI}; \quad (1)$$

$$RC\left(C_i,C_j\right)\geq T_R\ , \quad (2)$$

where $RI(C_i,C_j)$ is the relative mutual connectivity of a pair of clusters C_i and C_j; $RC(C_i,C_j)$ is the relative mutual similarity between a pair of clusters C_i and C_j.

If conditions (1) and (2) are met by more than one cluster C_j, adjacent to cluster C_i, the most connected cluster (subgraph) is selected for unification, i.e., such cluster C_j, with which cluster C_i has the largest absolute mutual connectivity $EC_{\{C_i,C_j\}}$. It is defined as the sum of the weights of the edges connecting the vertices belonging to C_i with the vertices belonging to C_j.

The second strategy proposes to combine at each step those clusters C_i and C_j, for which the profit function f_{profit}, which is calculated according to the following formula [15], attains its maximum value:

$$f_{profit} = RI\left(C_i,C_j\right) * RC\left(C_i,C_j\right)^{\alpha}, \quad (3)$$

where α is a value chosen by the user. If α>1, then the method places more value on relative mutual similarity, and if α<1, then the method places more value on relative mutual connectivity.

The value of $RI(C_i,C_j)$ in formulas (1) and (3) is calculated according to the following expression [15, 16]:

$$RI\left(C_i,C_j\right) = \frac{\left|EC_{\{C_i,C_j\}}\right|}{\frac{1}{2}\left(\left|EC_{C_i}\right| + \left|EC_{C_j}\right|\right)}, \quad (4)$$

where $EC_{\{C_i,C_j\}}$ is the absolute mutual connectivity of a pair of clusters C_i and C_j; EC_{C_i} is the internal connectivity of the cluster C_i, which is defined as the weighted sum of edges included in the edge separator, which divides C_i into two absolutely equal subgraphs; EC_{C_j} is the internal connectivity of the cluster C_j, which is defined similarly to EC_{C_i}.

The value of $RC(C_i,C_j)$ in formulas (2) and (3) is calculated according to the following expression [15, 16]:

$$RC\left(C_i,C_j\right) = \frac{\overline{S}_{EC_{\{C_i,C_j\}}}}{\frac{\left|C_i\right|}{\left|C_i\right|+\left|C_j\right|}\overline{S}_{EC_{C_i}} + \frac{\left|C_j\right|}{\left|C_i\right|+\left|C_j\right|}\overline{S}_{EC_{C_j}}}, \quad (5)$$

where $\overline{S}_{EC_{\{C_i,C_j\}}}$ is the average weight of the edges that connect the vertices of the cluster C_i with the vertices of the cluster C_j, and the edges must belong to the separator of the edges $EC_{\{C_i,C_j\}}$; $\overline{S}_{EC_{C_i}}$ is the average weight of the edges that belong to the separator EC_{C_i}; $\overline{S}_{EC_{C_j}}$ is the average weight of the edges that belong to the separator EC_{C_j}.

Then Stage 3 can be represented as a sequence of the following steps:

– Step 1: get the set of clusters that was formed as a result of Step 5 in Step 2;

– Step 2: determine the length of each cluster C_i, which is an element of the obtained set;

– Step 3: choose a C_i cluster of minimum length;

– Step 4: choose cluster C_j, which must be adjacent to cluster C_i, if there are no adjacent clusters, then proceed to Step 9;

– Step 5: determine $EC_{\{C_i,C_j\}}$, EC_{C_i}, EC_{C_j} and calculate $\overline{S}_{EC_{\{C_i,C_j\}}}$, $\overline{S}_{EC_{C_i}}$, $\overline{S}_{EC_{C_j}}$, $RI(C_i,C_j)$ values (according to formula (4)) and $RC(C_i,C_j)$ (according to formula (5)) for clusters C_i and C_j;

– Step 6: if not all C_j clusters are considered, then return to Step 4;

– Step 7: if the first strategy is chosen, then from the considered pairs of clusters (C_i,C_j), merge the pair for which conditions (1) and (2) are fulfilled and the $EC_{\{C_i,C_j\}}$ value is maximum;

– Step 8: if the second strategy is chosen, then from the considered pairs of clusters (C_i,C_j), merge the pair for which the value of function (3) is maximal;

– Step 9: if not all C_i clusters are considered, then exclude the C_i cluster from further consideration in Step 3, adjust the obtained set according to the results of Step 7 or Step 8 and return to Step 3, otherwise go to Step 10;

– Step 10: if the merge did not occur, then complete Step 3.

The results of Stage 3 will depend on the strategy chosen during this stage. If the first strategy is chosen, the results will be a set of distinct clusters, each of which defines a separate CI and contains a subset of interrelated CI functions. It can be said that in this way functional tasks are distinguished in ISEM as a set of functions that are similar in the descriptions of their structural elements and are strongly connected with each other by data flows. If the second strategy was chosen, the result will be a dendrogram of clusters, which displays all possible options for dividing the studied system into individual CIs. In this case, the choice of the final option will be carried out by the analyst on the basis of weakly formalized factors that affect the design of the ISEM and the IT project of its construction.

5. Results of solving the task of early identification of configuration items in the information system

5. 1. Results of adapting the Chameleon method to the peculiarities of solving the task of early identification

Constructing a distance matrix in Step 1 of Step 1 requires establishing a way to determine the distances between objects in the data sample to be analyzed. To this end, it is first necessary to establish a technique of formalized description of individual functions of ISEM as CIs to be identified.

As such, it is suggested to use a data flow diagram. This visual model makes it possible to consider each function of ISEM as a tuple consisting of the following groups of descriptions:

– a group of descriptions of entities or classes that characterize a function;

– a group of descriptions of entities or classes that form input data streams;

– a group of descriptions of entities or classes that form the output data streams.

This description of the functions of ISEM allows us to use the modified Chebyshev distance to determine the distances between the functions. This distance is calculated according to the following formula [14]:

$$d_{\infty}\left(i_p,i_q\right) = \max_{1\le l\le m}\sum_{k=1}^{n} i_{plk} \oplus i_{qlk}, \quad (6)$$

where i_{plk} is the value of the variable (0 or 1), which determines the absence or presence of a description of the k-th essence or class in the description of the function, input or output stream i_{pl} CI i_p; i_{qlk} is the value of the variable (0 or 1), which determines the absence or presence of a description of the k-th essence or class in the description of the function, input or output flow i_{ql} CI i_q; n is the maximum value of the identifier of the essence or class participating in the descriptions of the compared functions, input or output streams of CI i_p and i_q; $\oplus$ – operation "sum modulo 2".

A detailed description of the features of the formation of the modified Chebyshev distance is given in [14].

To perform Step 2 of Stage 1, it is suggested to determine the value of k=1. Such a value allows taking into account the presence of only the nearest neighbors of individual functions and will not lead to recognition as a separate CI of the functional module with a large number of functions and a monolithic architecture.

To determine the weights of the edges of the graphs in Step 1 of Stage 2, it was taken into account that the chosen method for determining the distance according to formula (6) is based on establishing the degree of similarity between CIs. In this case, the smaller the distance between two CIs, the closer these CIs are to each other and the denser the area in which these nearest neighbors are defined. Therefore, the weight of each edge of the graphs resulting from Stage 1 was determined by the formula:

$$w_{ij} = (d_{\max} + 1) - d_{ij}, \quad (7)$$

where w_{ij} is the weight of the edge that connects the vertices of clusters C_i and C_j; d_{max} is the maximum distance between CIs that were selected during the search for k nearest neighbors; d_{ij} is the distance between the peaks of clusters C_i and C_j.

To perform Step 3 of Stage 2, it is proposed to define the following separation conditions:

– the separator of the edges of the graph or subgraph, the division of which is carried out, must be minimal;

– each of the subgraphs resulting from the division contains 50 % of the vertices of the original graph or subgraph (50 %±1 vertex for cases when the number of vertices of the original graph or subgraph is odd);

– the minimum number of vertices of the subgraph resulting from division must be at least 2 (if the number of vertices is equal to 1, then it is impossible to determine the value of EC_{C_i} and EC_{C_j}).

5. 2. Checking the possibility of solving the task of early identification of configuration items using the Chameleon method

In order to check the possibility and identify the features of using the Chameleon method when solving the task of early identification of CI, it is suggested to use the descriptions of the functional task "Formation and maintenance of an individual plan of a scientific and pedagogical employee of the department". This task was implemented as a separate IT product for the development of the capabilities of the "University" information and analytical system at the Kharkiv National University of Radio Electronics. Previously, this system implemented the functional task "Distribution of educational load between lecturers at the department", the main source document of which is one of the sections of the document "Individual plan of a scientific and pedagogical employee at the department".

Detailed descriptions of the functional task "Formation and maintenance of an individual plan of a scientific and pedagogical employee of the department" are given in [14].

As a description of individual functions, it is proposed to consider each individual work of a data flow diagram with input and output data flows that are associated with this work [14, 17]. The obtained descriptions of the functions of the task "Formation and maintenance of an individual plan of a scientific and pedagogical employee of the department" are given in Table 1 [17]. The letters "CI" in Table 1 denote the IDs of the individual works that describe the relevant task functions as individual initial *CI*s are indicated. Elements with the value "1" indicate the facts of the use of the essence with the identifier, which is given in the column of Table 1, to describe the operation, input or output data flow with the identifier given in the row of Table 1. Elements with the value "0" indicate the opposite facts.

Table 1

Descriptions of configuration items for the task "Formation and maintenance of an individual plan of a scientific and pedagogical employee at the department"

Description of function *CI*												
CI	Essence IDs											
	1	2	3	4	5	6	7	8	9	10	11	12
*CI*1	1	1	1	1	1	1	0	0	0	0	0	0
*CI*2	1	1	1	1	1	1	1	1	1	0	0	0
*CI*3	1	1	1	1	1	1	1	1	1	0	0	0
*CI*4	1	1	1	1	1	1	1	1	1	0	0	0
*CI*5	0	1	0	1	0	1	1	0	0	1	1	0
*CI*6	0	0	0	0	0	0	0	1	1	0	0	0
*CI*7	0	0	0	0	0	0	0	0	0	0	0	1
*CI*8	1	1	0	1	1	1	1	1	1	0	0	1
*CI*9	1	1	0	1	1	1	1	1	1	0	0	0
*CI*10	1	1	0	1	1	1	1	1	1	1	1	0
Description of incoming data streams *CI*												
CI	Essence IDs											
	1	2	3	4	5	6	7	8	9	10	11	12
*CI*1	1	1	1	0	0	0	0	0	0	0	0	0
*CI*2	1	1	1	1	1	1	1	1	1	0	0	0
*CI*3	1	1	1	1	1	1	1	1	1	0	0	0
*CI*4	1	1	1	1	1	1	1	1	1	0	0	0
*CI*5	0	1	0	1	0	1	1	0	0	1	1	0
*CI*6	0	0	0	0	0	0	0	1	1	0	0	0
*CI*7	0	0	0	0	0	0	0	0	0	0	0	1
*CI*8	1	1	0	1	0	1	1	1	1	0	0	0
*CI*9	1	1	0	1	1	1	1	1	1	0	0	0
*CI*10	1	1	0	1	1	1	1	1	1	1	1	0
Description of outgoing data streams *CI*												
CI	Essence IDs											
	1	2	3	4	5	6	7	8	9	10	11	12
*CI*1	1	1	0	1	1	1	0	0	0	0	0	0
*CI*2	0	1	0	1	0	1	1	1	1	0	0	0
*CI*3	0	1	0	1	0	1	1	1	1	0	0	0
*CI*4	0	1	0	1	0	1	1	1	1	0	0	0
*CI*5	0	1	0	1	0	1	1	0	0	1	1	0
*CI*6	0	0	0	0	0	0	0	1	1	0	0	0
*CI*7	0	0	0	0	0	0	0	0	0	0	0	1
*CI*8	1	1	0	1	1	1	1	0	0	0	0	1
*CI*9	1	1	0	1	1	1	1	1	0	0	0	0
*CI*10	1	1	0	1	1	1	1	1	1	1	1	0

Titles of activities in the data flow diagram from Table 1 are given in Table 2.

The result of Step 1 of Stage 1 is the distance matrix D, which is given in Table 3. Distances were calculated according to formula (6).

Table 2

Titles of the activities in the data flow diagram for the functional task "Formation and management of the individual plan of the scientific and pedagogical employee at the department"

Activity	
Designation in Table 1	Title
*CI*1	Conversion of the section «Educational work
*CI*2	Formation of the section «Scientific work»
*CI*3	Formation of the section «Methodical work»
*CI*4	Formation of the section «Organizational work»
*CI*5	Formation of a list of positions and long-term assignments
*CI*6	Formation of a list of works recommended for implementation
*CI*7	Formation and maintenance of regulatory and reference information on KPIs
*CI*8	Formation of KPI of the lecturer and part of the KPI of the department
*CI*9	Formation of a summary table for the academic year
*CI*10	Formation of the resulting document «Individual plan»

Table 3

Matrix of distances *D* (Chameleon method) between configuration items in the functional task "Formation and maintenance of an individual plan of a scientific and pedagogical employee at the department"

CI	*CI*1	*CI*2	*CI*3	*CI*4	*CI*5	*CI*6	*CI*7	*CI*8	*CI*9	*CI*10
*CI*1	0	6	6	6	7	8	7	6	7	9
*CI*2	6	0	0	0	7	7	10	4	3	4
*CI*3	6	0	0	0	7	7	10	4	3	4
*CI*4	6	0	0	0	7	7	10	4	3	4
*CI*5	7	7	7	7	0	8	7	7	6	4
*CI*6	8	7	7	7	8	0	3	7	7	9
*CI*7	7	10	10	10	7	3	0	8	9	11
*CI*8	6	4	4	4	7	7	8	0	2	4
*CI*9	7	3	3	3	6	7	9	2	0	3
*CI*10	9	4	4	4	4	9	11	4	3	0

For each *CI*, as a result of Step 2 and Step 3 of Stage 1, the nearest neighbors were established:

– for *CI*1 – *CI*2, *CI*3, *CI*4 and *CI*8;
– for *CI*2 – *CI*3 and *CI*4;
– for *CI*3 – *CI*2 and*CI*4;
– for *CI*4 – *CI*2 and *CI*5;
– for *CI*5 – *CI*10;
– for *CI*6 – *CI*7;
– for *CI*7 – *CI*6;
– for *CI*8 – *CI*9;
– for *CI*9 – *CI*8;
– for *CI*10 – *CI*9.

As a result of Step 4 and Step 5 of Stage 1, the final partially directed graphs were formed, which are shown in Fig. 2.

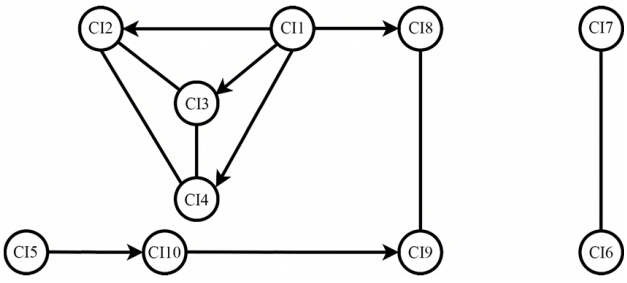

Fig. 2. Partially directed graphs resulting from Step 1 of the Chameleon method

Thus, the result of Step 1 of the Chameleon method for the case of one nearest neighbor is a set of two disconnected graphs.

As a result of performing Step 1 of Stage 2 for the graphs shown in Fig. 2, the weights of each of their edges were determined. The weighted graphs are shown in Fig. 3.

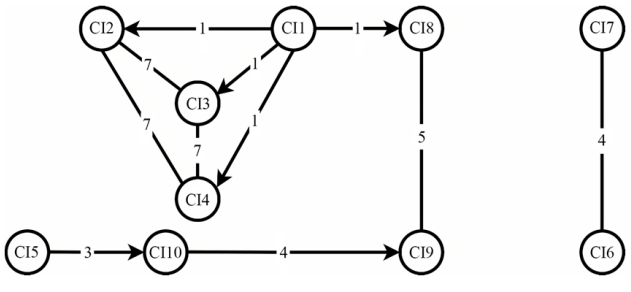

Fig. 3. Weighted partially directed graphs resulting from Step 1 of Step 2 of the Chameleon method

During the first iteration of Step 2 of Stage 2, the graph with vertices *CI*1, *CI*2, *CI*3, *CI*4, *CI*5, *CI*8, *CI*9, *CI*10 was chosen as the graph with the largest number of vertices.

During the first iteration of Step 3 of Stage 2, the original graph that was selected in the first iteration of Step 2 of Stage 2 was split into two subgraphs. The result of separation is shown in Fig. 4.

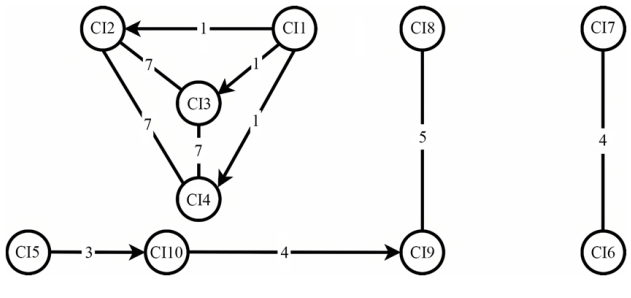

Fig. 4. Result of partitioning a weighted partially directed graph on the first iteration of Step 3 of Stage 2 of the Chameleon method

During the first iteration of Step 4 of Stage 2, it is established that the subgraphs obtained as a result of Step 3 of Stage 2 contain the following number of vertices:

– subgraph with vertices *CI*1, *CI*2, *CI*3, *CI*4 – four vertices;

– subgraph with vertices *CI*5, *CI*8, *CI*9, *CI*10 – four vertices.

This means that the division of these subgraphs should be continued because one of the conditions for stopping the division is not fulfilled (the number of vertices in the subgraphs is greater than 2).

As a result of the second and third iteration of Steps 2–4 of Stage 2, the subgraphs shown in Fig. 5 were obtained.

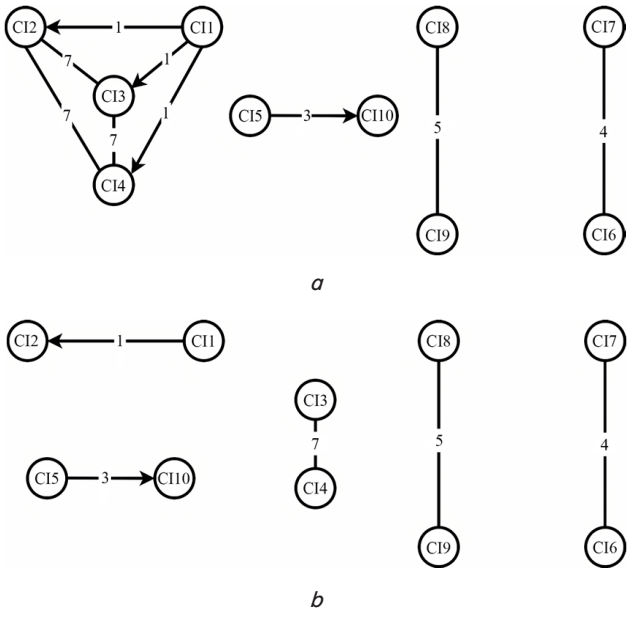

Fig. 5. Result of the division of the weighted partially directed graph in the second and third iterations of Step 2 and Step 3 of Stage 2 of the Chameleon method: *a* – result of division in the second iteration; *b* – result of dividing at the third iteration

As a result of the second iteration of Step 4 of Stage 2, it was established that the subgraph obtained as a result of the first iteration of Step 3 of Stage 2 contains four vertices (CI1, CI2, CI3, CI4). Therefore, its division should be continued on the third iteration of Steps 2–4 of Stage 2. As a result of the third iteration of Step 4 of Stage 2, it was established that each of the subgraphs obtained from the results of the first iteration of Steps 2–4 contains two vertices and therefore their separation should be stopped.

As a result of Step 5 of Stage 2, it was established that each of the resulting graphs and subgraphs of the original graph contains two vertices and cannot be divided. Therefore, the set of clusters resulting from Step 2 of the Chameleon method takes the following form:

$$C=\left\{\begin{matrix} C1=\{CI1,CI2\},C2=\{CI5,CI10\}, \\ C3=\{CI3,CI4\}, \\ C4=\{CI8,CI9\},C5=\{CI6,CI7\} \end{matrix}\right\}, \quad (8)$$

where C is a set of clusters as a result of Stage 2 of the Chameleon method; C1, ..., C5 – designation of individual clusters as elements of the set C.

Thus, the result of Step 2 of the Chameleon method for the single nearest neighbor case is a set (8) of five clusters.

As a result of performing Step 1 and the first iteration of Step 2 of Stage 3 of the Chameleon method, the set (8) was obtained, and it was determined that the length of each of the clusters that are elements of this set is equal to 2. Therefore, at the first iteration of Step 3 of Stage 3, it is proposed to choose cluster C1 as the minimum-length cluster C_i because it is the first in the list of elements of minimum length of set (8).

During the first and second iterations of Step 4 of Stage 3, it was established that clusters $C3$ and $C4$ are adjacent to cluster $C1$. The results of calculations performed on the first and second iterations of Step 5 of Stage 3 are given in Table 4.

Table 4

Results of calculations of indicator values for pairs of clusters ($C1$, $C3$) and ($C1$, $C4$)

Indicator	Pair (C1, $C3$)	Pair (C1, $C4$)
$EC_{\{C_i,C_j\}}$	$EC_{\{C1,C3\}}$=1+1+7+7=16	$EC_{\{C1,C4\}}$=1
EC_{C_i}	$EC_{\{C1\}}$=1	$EC_{\{C1\}}$=1
EC_{C_j}	$EC_{\{C3\}}$=7	$EC_{\{C4\}}$=1
$\overline{S}_{EC_{\{C_i,C_j\}}}$	$\overline{S}_{EC_{\{C1,C3\}}}=4$	$\overline{S}_{EC_{\{C1,C4\}}}=1$
$\overline{S}_{EC_{C_i}}$	$\overline{S}_{EC_{\{C1\}}}=1$	$\overline{S}_{EC_{\{C1\}}}=1$
$\overline{S}_{EC_{C_j}}$	$\overline{S}_{EC_{\{C3\}}}=7$	$\overline{S}_{EC_{\{C4\}}}=5$
$RI(C_i,C_j)$	$RI(C1,C3)$=4	$RI(C1,C4)$=1
$RC(C_i,C_j)$	$RC(C1,C3)$=1	$RC(C1,C4)$≈0,334

In order to analyze the peculiarities of using the Chameleon method when solving the task of early identification of CIs in ISEM, it is proposed to consider the progress and results of the application of each of the strategies. For the case of the first strategy, it is suggested to set the following values for conditions (1) and (2): T_{RI}=1, T_{RI}=0.75. The choice of these values is due to the following reasons:

– the selection of the T_{RI}=1 value is due to the need for the existence of at least one data flow that would connect the functions whose descriptions are in the merging clusters;

– the choice of the value T_{RC}=0.75 is due to the desire to combine into a single cluster those functions whose descriptions of structural elements coincide by at least 75 %.

During Step 7 of Stage 3, it was established that conditions (1) and (2) are fulfilled only for the pair of clusters ($C1$, $C3$). Therefore, there is no need to calculate the maximum $EC_{\{C_i,C_j\}}$ value. Based on the results of Step 7 of Stage 3, it is proposed to merge clusters $C1$ and $C3$.

For the case of the second strategy, it is proposed to consider two options for the value of the α indicator: α=0.75 and α=1.25. This makes it possible to appreciate the specificity of using the Chameleon method to recognize the greater importance of both relative mutual connectivity and relative mutual similarity.

The result of calculating the value of function (3) for pairs of clusters ($C1$, $C3$) and ($C1$, $C4$) at Step 8 of Stage 3 is given in Table 5.

Table 5

Results of calculating the indicator values for pairs of clusters (C1, $C3$) and (C1, C4)

Indicator	Pair (C1, $C3$)	Pair (C1, $C4$)
α=0.75	f_{profit}=4	f_{profit}≈0.147
α=1.25	f_{profit}=4	f_{profit}≈0.085

Based on the results of Step 8 of Stage 3, it is proposed to merge clusters $C1$ and $C3$.

As a result of merging clusters $C1$ and $C3$, the resulting set (8) will take the following form:

$$C=\left\{\begin{array}{c} C6=\{CI1,CI2,CI3,CI4\}, \\ C2=\{CI5,CI10\}, \\ C4=\{CI8,CI9\}, C5=\{CI6,CI7\} \end{array}\right\}, \quad (9)$$

where $C6$ is the designation of the cluster that arose as a result of the merger of clusters $C1$ and $C3$.

In the second iteration of Step 3 of Stage 3, it is proposed to choose cluster $C2$ as the cluster of minimum length C_i because it is the first in the list of elements of minimum length of the set (9).

During the first iteration of Step 4 of Stage 3, it was established that cluster $C4$ is adjacent to cluster $C2$. The results of our calculations performed on the first iteration of Step 5 of Stage 3 are given in Table 6.

Table 6

Results of calculating the indicator values for a pair of clusters (C2, C4)

Indicator	Pair (C2, $C4$)
$EC_{\{C_i,C_j\}}$	$EC_{\{C2,C4\}}=4$
EC_{C_i}	$EC_{\{C2\}}=3$
EC_{C_j}	$EC_{\{C4\}}=5$
$\overline{S}_{EC_{\{C_i,C_j\}}}$	$\overline{S}_{EC_{\{C2,C4\}}}=4$
$\overline{S}_{EC_{C_i}}$	$\overline{S}_{EC_{\{C2\}}}=3$
$\overline{S}_{EC_{C_j}}$	$\overline{S}_{EC_{\{C4\}}}=5$
$RI(C_i,C_j)$	$RI(C1,C4)=1$
$RC(C_i,C_j)$	$RC(C1,C4)=1$

During the implementation of Step 7 of Stage 3, it was established that conditions (1) and (2) are fulfilled for a pair of clusters ($C2$, $C4$). Based on the results of Step 7 of Stage 3, clusters $C2$ and $C4$ are proposed to be merged.

The result of calculating the value of function (3) for a pair of clusters ($C2$, $C4$) at Step 8 of Stage 3 is given in Table 7.

Table 7

Results of calculating the indicator values for a pair of clusters (C2, C4)

Indicator	Pair (C2, $C4$)
α=0.75	$f_{profit}=1$
α=1.25	$f_{profit}=1$

Based on the results of Step 8 of Stage 3, clusters $C2$ and $C4$ are proposed to be merged.

As a result of merging clusters $C2$ and $C4$, the adjusted resulting set (9) will take the following form:

$$C=\left\{\begin{array}{c} C6=\{CI1,CI2,CI3,CI4\}, \\ C7=\{CI5,CI8,CI9,CI10\}, \\ C5=\{CI6,CI7\} \end{array}\right\}, \quad (10)$$

where $C7$ is the designation of the cluster that arose as a result of the merger of clusters $C2$ and $C4$.

At the third iteration of Step 3 of Stage 3, it is proposed to choose cluster $C5$ as the cluster of minimum length C_i because it is the first in the list of elements of minimum length of the set (10). But during the first and second iterations of Step 4 of Stage 3, it was established that there are no adjacent clusters for cluster $C5$. Therefore, the adjusted set (10) obtained at the previous iteration of Step 3 execution remains unchanged, the $C5$ cluster will not be considered in subsequent iterations of Step 3, and the execution of Step 3 of the Chameleon method continues.

At the fourth iteration of Step 3 of Stage 3, it is proposed to choose cluster $C6$ as the cluster of minimum length C_i because it is the first in the list of the remaining elements of minimum length of the set (10).

During the first iteration of Step 4 of Stage 3, it was established that cluster $C7$ is adjacent to cluster $C6$. The results of the calculations performed on the first iteration of Step 5 of Stage 3 are given in Table 8.

Table 8

Results of calculating the indicator values for a pair of clusters (C6, C7)

Indicator	Pair (C6, $C7$)
$EC_{\{C_i,C_j\}}$	$EC_{\{C6,C7\}}=1$
EC_{C_i}	$EC_{\{C6\}}=16$
EC_{C_j}	$EC_{\{C7\}}=4$
$\overline{S}_{EC_{\{C_i,C_j\}}}$	$\overline{S}_{EC_{\{C6,C7\}}}=1$
$\overline{S}_{EC_{C_i}}$	$\overline{S}_{EC_{\{C6\}}}=4$
$\overline{S}_{EC_{C_j}}$	$\overline{S}_{EC_{\{C7\}}}=5$
$RI(C_i,C_j)$	$RI(C6,C7)=0.1$
$RC(C_i,C_j)$	$RC(C6,C7)\approx 0.223$

During Step 7 of Stage 3, it was established that conditions (1) and (2) for a pair of clusters ($C6$, $C7$) are not fulfilled.

The result of calculating the value of function (3) for a pair of clusters ($C6$, $C7$) at Step 8 of Stage 3 is given in Table 9.

Table 9

Results of calculating the indicator values for a pair of clusters (C2, C4)

Indicator	Pair (C6, $C7$)
α=0.75	$f_{profit}=0.0325$
α=1.25	$f_{profit}=0.0153$

According to the results of Step 8 of Stage 3, clusters $C6$ and $C7$ are proposed to be combined (to construct a dendrogram of clusters).

According to the results of Step 8 of Stage 3 and the merging of clusters $C6$ and $C7$, the adjusted resulting set (10) will take the following form:

$$C=\left\{\begin{array}{c} C8=\left\{\begin{array}{l} CI1,CI2,CI3,CI4, \\ CI5,CI8,CI9,CI10 \end{array}\right\}, \\ C5=\{CI6,CI7\} \end{array}\right\}, \quad (11)$$

where *C*8 is the designation of the cluster that arose as a result of the merger of clusters *C*6 and *C*7.

At the fifth iteration of Step 3 of Stage 3, it is proposed (in the case of the second strategy) to choose cluster *C*8 as the cluster of minimum length C_i because it is the first in the list of the remaining elements of minimum length of the set (11). But during the first iteration of Step 4 of Stage 3, it was established that there are no adjacent clusters for cluster *C*8. Therefore, the adjusted set (11) obtained at the previous iteration of Step 3 execution remains unchanged, the *C*8 cluster will not be considered in subsequent iterations of Step 3, and the execution of Step 3 of the Chameleon method is completed.

The result of performing Stage 3 according to the first strategy is the adjusted set (10), which consists of the following clusters:

– cluster C6={*CI*1, *CI*2, *CI*3, *CI*4};

– cluster C7={*CI*5, *CI*8, *CI*9, *CI*10};

– cluster C5={*CI*6, *CI*7}.

These clusters establish the result of solving the task of early identification of CI for the functional task "Formation and maintenance of an individual plan of a scientific and pedagogical employee at the department" in the form of the following team backlogs:

– backlog No. 1, which consists of functions marked as *CI*1, *CI*2, *CI*3 and *CI*4;

– backlog No. 2, which consists of functions marked as *CI*5, *CI*8, *CI*9 and *CI*10;

– backlog No. 3, which consists of functions marked *CI*6 and *CI*7.

The result of performing Stage 3 according to the first strategy is the set (8) and the adjusted sets (9) to (11), which allow us to construct the cluster dendrogram shown in Fig. 6.

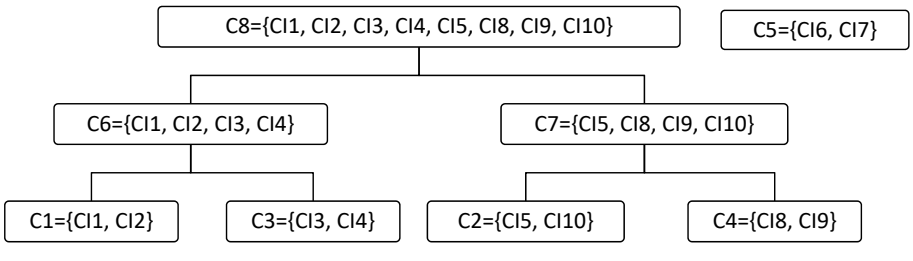

Fig. 6. Dendrogram of clusters, which is the result of solving the task of early identification of CIs in the functional task "Formation and maintenance of an individual plan of a scientific and pedagogical employee at the department" by the Chameleon clustering method

Based on this dendrogram, the analyst can make decisions regarding the formation of backlogs of teams of IT project developers for designing the functional task "Formation and maintenance of an individual plan of a scientific and pedagogical employee at the department." In particular, it should be recognized that the backlog, which includes CI6 and CI7 functions, will in any case be separated from other backlog options.

5. 3. Comparative analysis of features in using the Chameleon method and other methods of hierarchical clustering

In order to identify the features (advantages and disadvantages) of the Chameleon clustering method, it is proposed to conduct an analysis of the results obtained when solving the task of early identification of ISEM CIs. To this end, it is proposed to compare these results with the results of solving the same problem, obtained as a result of using other methods of hierarchical clustering.

The following results are considered:

– the solution obtained as a result of using the method based on the DIANA divizyme clustering method;

– the solution obtained as a result of using the AGNES agglomerative clustering method [17].

In both cases, the same problem that was solved during the Chameleon method testing in this study was used to test these methods.

The solution obtained as a result of using the DIANA method is the cluster dendrogram shown in Fig. 7 [14].

The result of applying the AGNES agglomerative clustering method (using the nearest neighbor algorithm) is the cluster dendrogram shown in Fig. 8 [17].

It is proposed to evaluate the received solutions using the following criteria:

– "The number of vertices of the dendrogram" characterizes the complexity of perception and further processing of the received solutions by humans or machine tools;

– "Number of levels of decomposition of the dendrogram" characterizes the number of alternatives that are subject to analysis for further decision-making regarding the distribution of backlogs between the teams of IT project performers for the development of a functional task;

– "Evenness of filling the elements of the dendrogram" characterizes the evenness of the subsequent loading of the executors of the IT project of the development of a functional task with tasks for the development of individual functions of this task.

According to the criteria "Number of dendrogram vertices" and "Number of dendrogram decomposition levels", the analyzed dendrogram will be considered better if the value of these criteria for this dendrogram is smaller than for other dendrograms.

According to the criterion "Evenness of filling the elements of the dendrogram", the dendrogram will be considered better if the number of CIs in different clusters at the studied level of dendrogram decomposition is the same.

The results of the comparative analysis according to the criterion "Number of vertices of the dendrogram" are given in Table 10.

Table 10

Results of the comparative analysis of dendrograms constructed using the Chameleon, DIANA, and AGNES methods, according to the criterion "Number of vertices of the dendrogram"

Dendrogram variant	Number of vertices	Qualitative assessment
Dendrogram by the Chameleon method	8	Best
Dendrogram by the DIANA method	15	Medium
Dendrogram by the AGNES method	19	Worst

The results of the comparative analysis according to the criterion "Number of levels of decomposition of the dendrogram" are given in Table 11.

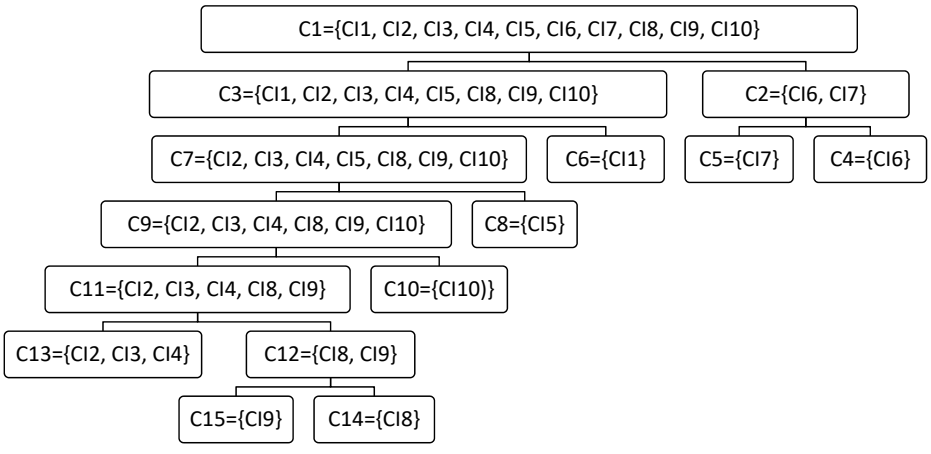

Fig. 7. Dendrogram of clusters of configurational elements, constructed as a result of the application of the DIANA divizyme clustering method

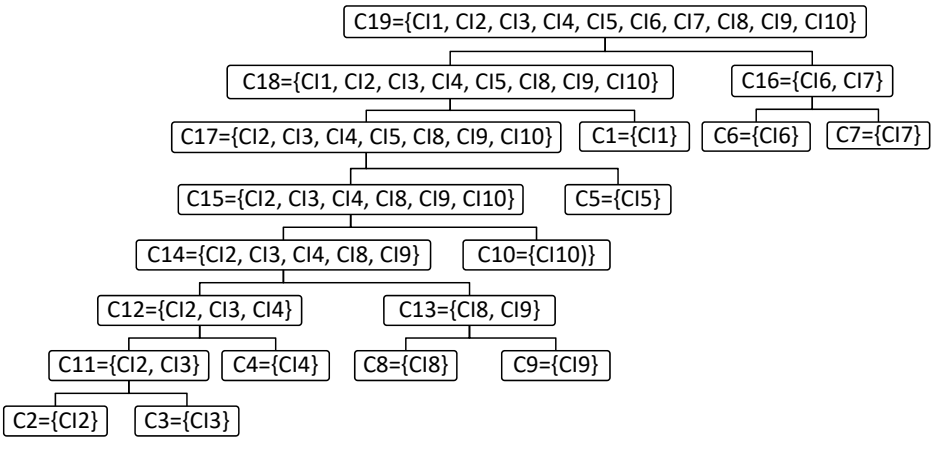

Fig. 8. Dendrogram of clusters of configuration items, constructed as a result of the application of the AGNES agglomerative clustering method

Table 11

Results of the comparative analysis of dendrograms constructed using Chameleon, DIANA, and AGNES methods, according to the criterion "Number of levels of dendrogram decomposition"

Dendrogram variant	Number of decomposition levels	Qualitative assessment
Dendrogram by the Chameleon method	3	Best
Dendrogram by the DIANA method	7	Close to the worst
Dendrogram by the AGNES method	8	Worst

The results of the comparative analysis according to the criterion "Evenness of filling the elements of the dendrogram" are given in Table 12.

In Table 12, decomposition level "0" contains the elements that are the roots of the dendrograms under consideration. If there is only one element at these levels, it does not participate in the evenness evaluation. According to the criterion "Evenness of filling the elements of the dendrogram", it is proposed to consider the dendrogram with the overall average evenness score equal to 1. This score indicates that at any level of decomposition, the number of CIs in the selected clusters is the same.

Now let's compare individual clusters of the dendrogram shown in Fig. 6, with clusters of dendrograms shown in Fig. 7, 8. According to the results of this comparison, the following should be noted:

– cluster $C4$ of the dendrogram constructed using the Chameleon method completely coincides with cluster $C12$ of the dendrogram constructed using the DIANA method and cluster $C13$ of the dendrogram constructed using the AGNES method;

– cluster $C5$ of the dendrogram constructed using the Chameleon method completely coincides with cluster $C2$ of the dendrogram constructed using the DIANA method and cluster $C16$ of the dendrogram constructed using the AGNES method;

– cluster $C8$ of the dendrogram built by the Chameleon method completely coincides with cluster $C3$ of the dendrogram built by the DIANA method and with cluster $C18$ of the dendrogram built by the AGNES method.

Table 12

Results of the comparative analysis of dendrograms constructed using the Chameleon, DIANA, and AGNES methods, according to the criterion "Evenness of filling the elements of the dendrogram"

Decomposition level	Number of clusters at the decomposition level	Ratio of CI quantities in clusters	Assessment of the degree of uniformity
Dendrogram by the Chameleon method			
0	2	8:2	4
1	2	4:4	1
2	4	2:2:2:2	1
General average assessment of the uniformity of filling of the elements of the dendrogram according to the Chameleon method			2 (best)
Dendrogram by the DIANA method			
0	1	–	–
1	2	8:2	4
2	4	7:1:1:1	7
3	2	6:1	6
4	2	5:1	5
5	2	3:2	1,5
6	2	1:1	1
Overall average estimation of the uniformity of filling in the elements of the dendrogram according to the DIANA method			4.083 (worst)
Dendrogram by the AGNES method			
0	1	–	–
1	2	8:2	4
2	4	7:1:1:1	7
3	2	6:1	6
4	2	5:1	5
5	2	3:2	1.5
6	4	2:1:1:1	2
7	2	1:1	1
Overall average assessment of the uniformity of filling in the elements of the dendrogram using the AGNES method			3.786 (close to the worst)

6. Discussion of results of using the Chameleon method when solving the task of early identification of configuration items

In the research process, the Chameleon hierarchical clustering method was adapted to solve the task of early identification. This task is solved during the distribution of elements of the description of the ISEM architecture between the IT project teams for the creation of this system for further implementation. The result of solving the problem is CIs, which determine the backlogs of teams implementing the IT project for creating the ISEM.

The essence of adapting the Chameleon method to the peculiarities of solving the task of early identification of ISEM CIs is as follows:

– to determine the distance between the descriptions of individual functions of the ISEM, it is proposed to use the modified Chebyshev distance (6), which takes into account the similarity of the structural descriptions of such functions;

– the study of only the nearest neighbors of each ISEM function was chosen (by setting the value k=1), which allows us to abandon the creation of systems exclusively with a monolithic architecture and take into account the features of modular and service-oriented architectures;

– the method of calculating the weights (7) of the edges of the graph, which is the result of the implementation of Stage 1 of the Chameleon method, has been refined, which allows taking into account the peculiarities of the chosen method for determining the distance between the ISEM functions;

– the conditions for selecting subgraphs during Stage 2 of the Chameleon method have been specified.

Such adaptation results are explained by the assumptions adopted during the research about the way to describe the architecture of the ISEM being created. According to this assumption, such a description should be carried out using a data flow diagram, which is supplemented with elements of an essence-relationship diagram.

Next, the adapted Chameleon method was used to solve the task of early identification of CIs for the functional task "Formation and maintenance of an individual plan of a scientific and pedagogical employee at the department". When solving this problem, the peculiarities of using each of the two strategies for forming the clustering result, which are offered by the Chameleon method, were considered. It should be noted that the results of using the first strategy, represented as an adjusted set (10), are part of the results of using the second strategy, represented as a dendrogram of clusters, shown in Fig. 6. Consideration of the solution of the task of early identification by all strategies of the Chameleon method is explained by the need for a reasoned choice of which of these strategies would be the best for solving similar problems.

Our results of solving the task of early identification of CIs made it possible to conduct a comparative analysis of the

results obtained using the Chameleon method and similar results obtained using the DIANA and AGNES hierarchical clustering methods. The criteria "Number of vertices of the dendrogram", "Number of levels of decomposition of the dendrogram", and "Evenness of filling the elements of the dendrogram" were proposed for the analysis. In addition, an analysis of the match of individual clusters as elements of dendrograms constructed using the specified methods was carried out.

According to the specified criteria (Tables 10–12), the solution obtained using the Chameleon method is the best from the point of view of its practical application in IT project management. Based on the results of the matching analysis, it was established that the results obtained using the Chameleon method differ from the results obtained using the DIANA and AGNES methods. But these differences can be explained by the fact that they arise only in cases of analysis of fragments of the description of the ISEM, which differ in sufficiently strong connectivity.

It should be noted that the use of methods of hierarchical clustering, such as DIANA or AGNES, when solving the task of early identification of ISEM CIs [14, 17], made it possible to take into account only the internal similarity of individual functions among themselves. This was possible due to the use of the modified Chebyshev distance to determine the distance between these functions [14]. In contrast to them, the solution to the task of early identification of ISEM CIs, obtained using the Chameleon method, takes into account both the internal similarity of individual functions and the degree of their interconnectedness. This makes it possible to reduce the number of levels of decomposition of the dendrogram of clusters (in the case of using the second strategy) or to obtain one variant of the problem solution (in the case of using the first strategy).

These advantages of using the Chameleon method allow us to recognize that taking into account the connectivity of ISEM functions is a factor that greatly simplifies the solution to the task of early identification of these functions as CIs of the corresponding IT project. The dendrogram of clusters obtained using the Chameleon method (Fig. 6) offers a smaller number of alternatives for making a decision about the most desirable content of the executive teams' backlogs. The application of the first strategy makes it possible to determine the option of solution (10), which is the best from the point of view of the uniformity of the loading of executive teams during the implementation of the IT project of creating an ISEM. These results make it possible to adopt the main hypothesis of this study as true, according to which the application of the Chameleon method could improve the solution of the task of early identification of ISEM CIs.

The main limitations for applying our research results in further applied work and theoretical studies are:

– use of data flow diagrams and "essence – connection" as the main sources of information about the functions and data structures of the created IS;

– use of the method of determining the distance between CIs proposed in [14] in the clustering method;

– the need to construct, at least at the conceptual level, an "essence-connection" diagram for the analyzed IT product.

These limitations determine the main drawback of this study. This drawback consists in the need for a detailed description of each functional requirement and individual functions of the ISEM being created. This description should be based on the definition of data entities and their attributes. At the same time, the incompleteness of this description or the use of natural language for this description can lead to an inaccurate solution to the task of early identification.

Further development of our study can be carried out in several directions. One of these areas is conducting research aimed at identifying the peculiarities of using the Chameleon method for investigating descriptions of ISEM architectures and their separate fragments of different sizes. These experiments will make it possible to determine the level of scaling of the Chameleon method and to establish the limits of the application of this method for the design management of small, medium, or large ISEMs. The second direction is to improve the Chameleon method to account for the effort spent on the development of defined CIs. Such an improvement will make it possible to equalize the distribution of individual ISEM functions among the backlogs of the executive teams of the corresponding IT project not only in terms of the number, similarity, and connectivity of functions, but also in terms of labor costs for their implementation. Another area of research is the study of the possibilities of using the Chameleon method in those cases when other structural or object-oriented visual models are used to describe the created ISEM.

7. Conclusions

1. The Chameleon hierarchical clustering method has been adapted to the specificity of the task of early identification of ISEM CIs. The essence of this adaptation is to establish a technique for determining the distances between separate functions of ISEM and clarify the elements of the Chameleon method, which depend on the features of the description of the architecture of ISEM. The results of the adaptation make it possible to use the Chameleon method to solve the task of early identification of ISEM CIs, taking into account the relationships existing between individual functions of this system.

2. The adapted Chameleon method was used to solve the task of early identification of CIs. This test was carried out during the planning of the IT project for the development of the functional task "Formation and management of the individual plan of the scientific and pedagogical employee at the department". A data flow diagram and an essence-relationship diagram were used as a description of the problem functions. The results of solving the task of early identification of CI allow us to claim that the use of the Chameleon method makes it possible to obtain not only a dendrogram of clusters, as in the case of using other methods of hierarchical clustering, but also a separate, best solution option. The resulting dendrogram consists of 8 clusters, which are selected at three levels of decomposition. The solution option consists of three clusters, two of which ($C6$ and $C7$) contain 4 CIs each, and one ($C5$) contains 2 CIs.

3. A comparative analysis of the results of solving the task of early identification of CIs, obtained using Chameleon, DIANA, and AGNES hierarchical clustering methods, was carried out. For comparison, the criteria "Number of dendrogram vertices", "Number of levels of dendrogram decomposition", and "Evenness of filling dendrogram elements" have been proposed. For the solution that was obtained using the Chameleon method, the value of the "Number of vertices of the dendrogram" criterion is 8; the value of the

criterion "Number of dendrogram decomposition levels" is 3; the value of the criterion "Evenness of filling the elements of the dendrogram" is 2. These values are the best among the compared solutions. In addition, an analysis of the matching of individual clusters as elements of dendrograms constructed using the specified methods was carried out. According to the results of our comparative analysis, the Chameleon method was recognized as the best of the hierarchical clustering methods that participated in the analysis. The use of this method makes it possible to obtain more balanced solutions regarding the content of backlogs for the teams implementing the IT project for ISEM construction.

Conflicts of interest

The authors declare that they have no conflicts of interest in relation to the current study, including financial, personal, authorship, or any other, that could affect the study and the results reported in this paper.

Funding

The study was conducted without financial support.

Data availability

All data are available in the main text of the manuscript.

Use of artificial intelligence

The authors confirm that they did not use artificial intelligence technologies when creating the current work.

References

1. Bourque, P., Fairley, R. E. (Eds.) (2014). Guide to the Software Engineering Body of Knowledge. Version 3.0. IEEE Computer Society, 335. Available at: https://cs.fit.edu/~kgallagher/Schtick/Serious/SWEBOKv3.pdf
2. Quigley, J. M., Robertson, K. L. (2019). Configuration Management. Auerbach Publications. https://doi.org/10.1201/9780429318337
3. 15288-2015 - ISO/IEC/IEEE International Standard - Systems and software engineering -- System life cycle processes. https://doi.org/10.1109/ieeestd.2015.7106435
4. Farayola, O. A., Hassan, A. O., Adaramodu, O. R., Fakeyede, O. G., Oladeinde, M. (2023). Configuration management in the modern era: best practices, innovations, and challenges. Computer Science & IT Research Journal, 4 (2), 140–157. https://doi.org/10.51594/csitrj.v4i2.613
5. Reiff-Marganiec, S., Tilly, M. (Eds.) (2012). Handbook of Research on Service-Oriented Systems and Non-Functional Properties. IGI Global. https://doi.org/10.4018/978-1-61350-432-1
6. Cadavid, H., Andrikopoulos, V., Avgeriou, P., Broekema, P. C. (2022). System and software architecting harmonization practices in ultra-large-scale systems of systems: A confirmatory case study. Information and Software Technology, 150, 106984. https://doi.org/10.1016/j.infsof.2022.106984
7. Faitelson, D., Heinrich, R., Tyszberowicz, S. (2017). Supporting Software Architecture Evolution by Functional Decomposition. Proceedings of the 5th International Conference on Model-Driven Engineering and Software Development. https://doi.org/10.5220/0006206204350442
8. Shahin, R. (2021). Towards Assurance-Driven Architectural Decomposition of Software Systems. Computer Safety, Reliability, and Security. SAFECOMP 2021 Workshops, 187–196. https://doi.org/10.1007/978-3-030-83906-2_15
9. Suljkanović, A., Milosavljević, B., Inđić, V., Dejanović, I. (2022). Developing Microservice-Based Applications Using the Silvera Domain-Specific Language. Applied Sciences, 12 (13), 6679. https://doi.org/10.3390/app12136679
10. Felfernig, A., Le, V.-M., Popescu, A., Uta, M., Tran, T. N. T., Atas, M. (2021). An Overview of Recommender Systems and Machine Learning in Feature Modeling and Configuration. Proceedings of the 15th International Working Conference on Variability Modelling of Software-Intensive Systems. https://doi.org/10.1145/3442391.3442408
11. Abolfazli, A., Spiegelberg, J., Palmer, G., Anand, A. (2023). A Deep Reinforcement Learning Approach to Configuration Sampling Problem. 2023 IEEE International Conference on Data Mining (ICDM). https://doi.org/10.1109/icdm58522.2023.00009
12. Sellami, K., Saied, M. A., Ouni, A. (2022). A Hierarchical DBSCAN Method for Extracting Microservices from Monolithic Applications. The International Conference on Evaluation and Assessment in Software Engineering 2022. https://doi.org/10.1145/3530019.3530040
13. Krause, A., Zirkelbach, C., Hasselbring, W., Lenga, S., Kroger, D. (2020). Microservice Decomposition via Static and Dynamic Analysis of the Monolith. 2020 IEEE International Conference on Software Architecture Companion (ICSA-C). https://doi.org/10.1109/icsa-c50368.2020.00011
14. Ievlanov, M., Vasiltcova, N., Neumyvakina, O., Panforova, I. (2022). Development of a method for solving the problem of it product configuration analysis. Eastern-European Journal of Enterprise Technologies, 6 (2 (120)), 6–19. https://doi.org/10.15587/1729-4061.2022.269133
15. Karypis, G., Han, E.-H., Kumar, V. (1999). Chameleon: hierarchical clustering using dynamic modeling. Computer, 32 (8), 68–75. https://doi.org/10.1109/2.781637
16. Han, J., Kamber, M., Pei, J. (2012). Data Mining: Concepts and Techniques. Morgan Kaufmann. https://doi.org/10.1016/c2009-0-61819-5
17. Vasyltsova, N. V., Panforova, I. Yu. (2022). Doslidzhennia vykorystannia metodiv ierarkhichnoi klasteryzatsiyi pid chas vyrishennia zadachi analizu konfihuratsiyi IT-produktu. ASU ta prylady avtomatyky, 178, 37–49. Available at: https://www.ewdtest.com/asu/wp-content/uploads/2024/01/ASUiPA_178_37_49.pdf