DOI: 10.15587/978-617-8360-03-0.CH1

Maksym Ievlanov, Nataliya Vasiltcova,
Olga Neumyvakina, Iryna Panforova

USE OF CLUSTERING METHODS TO SOLVE THE PROBLEM OF IDENTIFYING CONFIGURATION ITEMS IN IT PROJECT

ABSTRACT

The object of the research is the IT project configuration management process.

During the research, the task of identifying the configuration items (CI) of the IT product has been solved. Research in this field is mainly aimed at solving the problem of configuration analysis during the refactoring of a monolithic IT product into separate services or microservices. The question of decomposition methods of the description of the architecture of the developed IT product into separate functional SIs remains practically unexplored.

In the course of the research, it has been proposed to use hierarchical and non-hierarchical clustering methods to solve the problem of identifying functional CIs. As examples of hierarchical clustering methods, the agglomerative clustering method (AGNES using the nearest neighbor algorithm) and the divisive clustering method DIANA have been proposed. The k-means algorithm has been proposed as an example of non-hierarchical clustering methods. In addition to them, it has been suggested to use one of the grapho-analytic clustering methods for comparison, which was developed to solve the decomposition problem of the description of the monolithic architecture of the software product into separate microservices.

The starting data for the research is the description of the architecture of the "Formation and management of the individual plan of the scientific and pedagogical worker of the department" functional task at the level of individual functions. 10 functions of this problem have been considered as CI. Descriptions of 12 entities of the problem database have been used to define these functions. The features of the solution have been considered and the results of the solution to the problem of identifying functional CIs using all four selected clustering methods have been obtained.

A comparative analysis of the process of solving and the obtained results of solving the task of identifying functional CIs using all four selected clustering methods has been carried out. It has been established that the best alternative is to use hierarchical clustering methods to solve this problem. This makes it possible to further consider the task of assigning to individual teams of IT project executors a list of functional CIs that require implementation as a sequence of individual single-criteria optimization tasks.

KEYWORDS

Configuration item, identification, IT product, architecture description, AGNES method, DIANA method, k-means algorithm, Chebyshev distance, Hamming distance, function, data flow diagram, ER diagram.

Modern views recognize the process of configuration management as one of the main processes of project management within the life cycle of IT products of various purposes. Such projects will be referred to as IT projects hereafter. And although different points of view define the appropriateness of the configuration management process, the formulation of the purpose of this process, which is a rule, coincides. Thus, in [1], the configuration management process is one of the integrated change control processes. In contrast to this point of view, in [2] the process of configuration management is one of the processes of technical management of an IT project. But the descriptions of the purpose of this process are more similar to each other. In [1], the purpose of this process is to systematically control configuration changes and maintain the integrity and tracking of the configuration throughout the entire product life cycle. In [2], the purpose of this process is to manage and control system items and configurations within the system life cycle. In addition, configuration management ensures meaningfulness between a product and the configuration definition associated with that product [2]. The variants of decomposition of the configuration management process also correspond. Thus, in [1] the main tasks of this process are: planning and management of the configuration management process; configuration identification; configuration control; configuration state accounting; configuration audit and product release and delivery management. In [2], the main actions of this process are: configuration management planning; configuration definition; implementation of configuration change management; performance of configuration status accounting; configuration evaluation; implementation of release control.

Certainly, the key work of the configuration management process is the work of identifying or defining the configuration. The results of the execution of almost all other works of this process depend on the results of this work. In general, the points of view on the tasks that are performed within the scope of this work coincide. Thus, in [1], the work on configuration identification includes the following tasks:

– determination of items that are subject to control;
– establishment of item identification schemes and their versions;
– establishment of tools and methods that will be used to obtain and manage selected items.

In [2], the configuration definition action includes the following tasks:

– determination of system items and information objects that are configuration objects;
– determination of hierarchy and structure of system information;
– setting identifiers of the system, system item and information object;

– determination of baselines during the life cycle;

– obtaining the agreement of the party acquiring the system to establish a baseline with the supplier.

But these points of view on the process of configuration management, its purpose and content are too theoretical and methodical in nature. A significant reason for this is, in particular, the lack of a clear idea of what exactly to consider controllable items within the configuration management process [1], or configuration objects [2]. Usually, such items are called "configuration items" (CI). These items can be IT products or their baselines, individual system items of such products, information objects, software, a set of hardware or software-hardware complexes. It is recognized that CI can be described by a hierarchical scheme of decomposition of products into separate system items, system items into separate software services, etc. [2]. As a result, it should be assumed that at different stages of the life cycle of an IT product, different descriptions of system items may appear as CI, depending on the current level of presentation of this IT product. An example of such a multi-level representation is proposed in [3].

Based on this assumption, it becomes possible to conduct research on methods of solving the problem of determining the optimal set of CIs at the early stages of the life cycle of IT products. Optimal here should be understood as such a CI set that will require minimal expenditure of time and resources for the implementation of relevant IT projects. Among these IT projects, special attention should be paid to IT projects of creation or development of information systems (IS) management of enterprises and organizations. As shown in [4], most cases of negative impact of local decisions on the overall design and quality of a large software and technical system are eliminated, in particular, by early division of the system into separate items. Solving the task of determining the optimal set of CI provides an assessment of the possibility of implementing an IT project of creating an IS at the stages of project initiation and planning. Such an assessment will make it possible to make more informed decisions about the possibility and feasibility of implementing such projects. Therefore, such studies should be recognized as relevant.

1.1 ANALYSIS OF MODERN RESEARCH IN THE FIELD OF IDENTIFYING CONFIGURATION ITEMS IN IT PRODUCT

Solving the problem of CI identification is of particular importance when designing large systems (the so-called "System of Systems"), which include a significant number of IS management of enterprises and organizations. Thus, in [4] it is shown that negative cases of the influence of local decisions on system-wide decisions arose mainly as a result of incorrect interpretation of requirements or bias of personal experience. This allows to draw a conclusion about the feasibility of using to solve the problem of CI identification during the identification and planning of an IT project for the creation of artificial intelligence methods IS. In such methods, the subjective influence of an individual specialist is minimized. At the same time, the basic description of the IS, which can be

decomposed into separate CIs, should be considered the description of the architecture of the IS being created. It is this description that combines the descriptions of individual requirements for IS and its individual functions [2].

The vast majority of modern research on solving the problem of decomposition of the description of the system architecture into separate CIs is based on the results of theoretical studies considered in [5]. From these results it follows that:

a) most of the existing approaches to the decomposition of a monolithic architecture into a set of individual microservices are applicable only under certain conditions;

b) there are probably no universal approaches to solving this problem.

However, the research results given in [5] practically do not take into account the transfer of the solution of this problem to a higher level of abstraction. Such a transfer is, in particular, the selection of CIs that implement individual IS functions from the description of the architecture of this system as a coherent set of functions.

The first of the above statements is supported by modern research on refactoring the code of a monolithic software application into individual microservices [6, 7]. At the same time, a density-based clustering algorithm is used to solve this problem in [6]. For such a special case of the decomposition problem of the description of the system architecture into separate CIs, the proposed solutions give positive results. However, a serious limitation of the application of these results is the need to observe the assumption of invariance of the functional decomposition of the original description of the software application architecture. This assumption may be violated as a result of a change in an existing or the appearance of a previously forgotten functional requirement for an IT product.

Theoretical studies, aimed at verifying the second of the above statements, made it possible to establish that the solution of most IT service selection problems requires an a priori definition of a set of functionally equivalent IT services [8]. This means that the decomposition problem of the description of the system architecture, created on the basis of a set of functional system requirements, into individual IT services must be solved separately for abstract descriptions of these services. Such abstract descriptions should not depend on the implementation specifics of these services and the non-functional requirements put forward to the services.

This conclusion is confirmed by the results of solving the problem of identification based on the description of the meta-architecture of the designed software system of individual software artifacts that ensure the reliable operation of this system [9]. This work shows that the use of abstractions to describe the architecture of a software system simplifies the solution of the decomposition problem of the description of the system architecture into separate items. However, the selection of such abstractions by functional feature was not considered in [9].

The application of the approaches proposed in [8] to the implementation of the functional decomposition of the description of the architecture of the software system into separate items is shown in [10]. In this work, at the first stage, the description of the architecture of the software system is divided into separate items, taking into account the necessary evolutionary changes.

The second stage of solving the decomposition problem in [10] is considered as the selection of such decomposition options that satisfy the constraints on the cost of the necessary changes. However, this approach to solving the decomposition problem leaves open the issue of assigning individual CIs to IT project teams. The solution to this issue requires the assignment to each individual team or performer of such a subset of CIs that would contain the most similar functional CIs among themselves.

The decomposition of the description of the system architecture into individual microservices using an object-oriented language was studied in [11]. The proposed Silvera language and its compiler make it possible to automate the decomposition of the system-wide description of the architecture into descriptions of individual microservices during the design of the software system. But the use of this solution is limited by the following features [11]:

– focusing exclusively on the decentralized development of microservices;

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

Also, in [3], the issue of the optimal (from the point of view of IT project costs) number of teams of microservices developers that are allocated is neglected.

The analysis of the considered publications allows to formulate the following conclusions:

a) for large IT projects of creation or development of IS for management of enterprises and organizations, the task of CI identification is an important task;

b) solving the problem of CI identification in similar IT projects is usually considered as solving the decomposition problem of the IS architecture description into separate CI;

c) solving this problem requires its division into two sequentially solved sub-problems:

– the subtask of forming a set of options for decomposing the description of the system architecture into separate functional CIs;

– the subtask of selecting from a set of separate functional CIs a subset that will satisfy the selection conditions in the best way.

At the same time, it should be remembered that the peculiarities of solving the problem of forming a set of options for the decomposition of the description of the system architecture into separate functional CIs have been studied very poorly.

These conclusions make it possible to recognize the need for research in the field of finding special methods for solving the subtask of forming a set of options for the decomposition of the description of the system architecture into separate functional CIs. Such methods should be aimed at forming a set of all possible options for decomposition of the IT product architecture description into individual CIs.

The aim of this research is to identify the advantages and disadvantages of using the simplest clustering methods to solve the subtask of forming a set of options for the decomposition of the description of the IS architecture into separate functional CIs. For this purpose, it is proposed to compare the progress and results of solving this subtask using hierarchical and non-hierarchical clustering algorithms. It is expected that the results of the research will allow to draw a conclusion about the expediency of using clustering methods to solve the problem of identifying functional CIs and reasonably recommend the best of the considered algorithms.

To achieve this aim, it is proposed to solve the following research problems:

– to determine the features of hierarchical and non-hierarchical clustering algorithms that will be used in the research;

– to formulate a description of the initial data of the identification task of functional CIs, which will be used to compare different solutions;

– to consider the progress and results of solving the problem of identifying functional CIs using selected representatives of hierarchical and non-hierarchical clustering algorithms;

– to conduct a comparative analysis of the results of the received decisions.

1.2 METHODS OF RESEARCH

Hierarchical clustering methods are usually divided into two large classes: agglomerative and divisive. As an example of agglomerative methods, it is proposed to use the AGNES hierarchical agglomerative clustering method. To solve the subtask of forming a set of options for the decomposition of the description of the system architecture into separate functional CIs, this method can be represented as a sequence of the following stages [12, 13]:

Stage 1. The entire set CI is represented as a set of clusters C, each of which contains one item CI_i, $i=1,\ldots,n$, where n is the number of items in the CI set. Calculate the distance matrix D between the items of the set C.

Stage 2. Select two clusters c_p and c_q, the distance between which will be minimal, and combine them into a new cluster c_r, entering it instead of clusters c_p and c_q into the set of clusters C.

Step 3. Calculate the values of the distance matrix D using the rule:

$$d_{rs} = \alpha_p \times d_{ps} + \alpha_q \times d_{qs} + \beta \times d_{pq} + \gamma \times \left| d_{ps} - d_{qs} \right|, \; r \neq s, \; r \neq p, \; r \neq q, \tag{1.1}$$

where d_{ps} – the distance between the centers of clusters c_p and c_s; d_{qs} – the distance between the centers of clusters c_q and c_s; d_{pq} – the distance between the centers of clusters c_p and c_q; α_p, α_q, β and γ – parameters whose value is determined based on the selected distance calculation algorithm.

Step 4. Repeat Step 2 and Step 3 until one cluster is formed that includes all items of the CI set.

It is proposed to use the nearest-neighbor clustering algorithm as an algorithm for calculating distances. For this algorithm, the parameters in (1.1) acquire the following values: $\alpha_p=0.5$; $\alpha_q=0.5$; $\beta=0$; $\gamma=-0.5$ [12, 13].

As an example of divisive methods, it is proposed to use the DIANA hierarchical divisive clustering method. In order to solve the subtask of forming a set of options for the decomposition of the description of the system architecture into separate functional CIs, this method can be represented as a sequence of the following stages [12–14]:

Stage 1. Form one cluster C1 consisting of all m items of the original set of clustering objects CI.

Step 2. Select the C1 cluster item with the largest average distance value from other cluster items. The average distance value for the i_p item of cluster C1 can be calculated as follows:

$$D_{C1}(i_p) = \frac{1}{m} \times \sum_{q=1}^{m} d(i_p, i_q) \; \forall i_p, i_q \in C1, \; p \neq q, \tag{1.2}$$

where m – the number of items in cluster C1; i_p – the p-th item of cluster C1; i_q – the q-th item of cluster C1; $d(i_p, i_q)$ – the distance between the items i_p and i_q.

Step 3. Remove the item selected in Step 2 from cluster C1 and include it in the new cluster C2.

Stage 4. Among the remaining items of the cluster, find one for which the difference between the average distance to the items remaining in the cluster C1 and the average distance to the items included in the cluster, C2, is positive and maximal.

Step 5. Delete the item selected in Step 4 from cluster C1 and include it in the new cluster C2.

Step 6. Continue to perform Steps 4, 5 until the differences of the average distances become negative, then terminate the method.

The result of this method is the division of the original cluster C1 into two children – C1 and C2. Next, one of the child clusters is selected and divided into two more child clusters using this method. The separation procedure stops in one of the following cases [11–13]:

a) only one item remains in the child cluster;

b) all items of the child cluster have zero difference from each other.

The results of the use of hierarchical classification methods are dendrograms, which describe the result of the decomposition of the IS architecture description into separate clusters of functional CIs. These results allow to proceed to the solution of the sub-problem of selecting from a set of separate functional CIs a subset that will satisfy the selection conditions in the best way. The solution of this subtask is aimed at grouping selected clusters of functional CIs into lists of tasks (backlogs) for IT project implementation teams. At the same time, it is assumed that each team of performers is homogeneous, that is, it consists of specialists with the same skills and level of quality of work performance.

This approach requires a significant investment of time. These costs could be reduced by using non-hierarchical clustering methods, namely partitioning methods. A classic example of these methods is the k-means algorithm [12–14]. The use of this algorithm makes it possible to determine clusters of functional CIs according to the number of executive teams that may be assigned to the implementation of the IT project of creation or development of this IS. In addition, the k-means algorithm is one of the simplest clustering methods and is the basis for a sufficiently large set of non-hierarchical algorithms [12–14].

Conducting research requires comparing the course and results of the proposed clustering methods. Therefore, it is necessary that the above methods are based on the same method of determining the distance between descriptions of functional CIs. For this purpose, the research introduces an assumption according to which descriptions of functional CIs are sets of entities or classes. Therefore, it is suggested to use the methods of determining the distance for objects

whose descriptions contain qualitative features. Among such descriptions, the most frequently used are [15]:

– Hamming distances $dH(x_i, x_j) = \sum_{l=1}^{n} \left| x_{il} - x_{jl} \right|$;

– distances based on the Hauer coefficient;

– distances based on different correlation coefficients (for example, Pearson, Spearman or Kendall).

However, these methods do not sufficiently take into account the peculiarities of the IS CI description. Therefore, the use of these methods in their pure form to solve the given sub-task is ineffective.

In order to establish a better way of determining the distance between descriptions of functional CIs, let's introduce the second assumption into the study. According to this assumption, the description of each individual IS function as a functional CI should be unified and represent a set of the following groups of descriptions [16]:

a) a group of descriptions of entities or classes that characterize a specific functional CI;

b) a group of descriptions of entities or classes of input data flows of a specific functional CI;

c) a group of descriptions of entities or classes of output data flows of a specific functional CI.

This assumption allows to state that the descriptions of two functional CIs should be considered different if they do not match at least in one of the above groups. Based on this, it is recommended to use the Chebyshev distance to determine the distance between descriptions of functional CIs based on the introduced assumptions. This distance is determined by the formula:

$$d_{\infty}(i_p, i_q) = \max_{1 \le l \le m} \left| i_{pl} - i_{ql} \right|, \tag{1.3}$$

where i_p, i_q – objects between which the distance is determined; i_{pl} – the l-th coordinate of the object i_p, $1 \le l \le m$; i_{ql} – the l-th coordinate of the object i_q, $1 \le l \le m$.

In this study, the objects i_p, i_q will be descriptions of the i-th and j-th functional CIs. The coordinates of each specific object will be the groups of function descriptions, input and output flows of the corresponding CIs discussed above. Based on this, the number m will be equal to 3 for the task of decomposition of the description of the IS architecture into separate clusters of functional CIs.

However, to determine the Chebyshev distance, it is necessary that the coordinate values be numerical for all objects. For the sub-task of decomposition of the description of the IS architecture into separate clusters of functional CIs, groups of descriptions, which are selected as coordinates, are sets of character strings. At the same time, the entities or classes, the descriptions of which form these groups, are considered as sets. The items of these sets are individual attributes (and, in the case of classes, also methods). Therefore, it is suggested to introduce additional assumptions:

a) the descriptions of each specific entity or class do not change depending on their inclusion in the descriptions of the function, input or output flows of different CIs;

b) each entity or class description can be matched with an identifier.

These assumptions make it possible to represent groups of descriptions of any functions, input and output flows of CIs as a set, the items of which are binary variables. These variables take the value 1 if the description of the entity or class with the corresponding identifier is present in the description of the function, input or output flow of the functional CI, and 0 otherwise. The proposed representation allows to modify the descriptions of the Hamming distance determination to determine the distance between separate groups as follows:

$$d_{H_m}(i_{pl}, i_{ql}) = \sum_{k=1}^{n} i_{plk} \oplus i_{qlk}, \tag{1.4}$$

where i_{plk} – a binary variable describing the presence of the identifier of the k-th entity or class, which are included in the description of the function, input or output flow i_{pl} of CI i_p; i_{qlk} – a binary variable that describes the presence of the identifier of the k-th entity or class that is included in the description of the function, input or output flow i_{ql} of CI i_q; n – the maximum number of identifiers that participate in the descriptions of the compared functions, input or output flows of CI i_p and i_q; $\oplus$ – operation "sum modulo 2".

The Hamming distance modified in this way will be equal to 0 in the case when the compared descriptions of functions, input or output flows consist of sets of identical entities or classes. In other cases, this distance will be equal to the number of entities or classes that will be present in only one of the two compared descriptions.

The use of the modified Hamming distance allows, in turn, to adapt the Chebyshev distance (3) to solve the subproblem of decomposition of the description of the IS architecture into separate clusters of functional CIs as follows [16]:

$$d_{\infty}(i_p, i_q) = \max_{1 \le l \le m} \sum_{k=1}^{n} i_{plk} \oplus i_{qlk}. \tag{1.5}$$

Using the modified Chebyshev distance (1.5) allows to determine the proximity degree of compared functional CIs based on representations of these CIs in the form of visual models.

1.3 SOLVING THE PROBLEM OF IDENTIFYING CONFIGURATION ITEMS USING THE DECLARED CLUSTERING METHODS

1.3.1 DESCRIPTION OF THE INITIAL DATA OF THE IDENTIFICATION PROBLEM OF CONFIGURATION ITEMS

The starting data for solving the problem of CI identification of an IT product is the CI of the "Formation and maintenance of an individual plan of a scientific and pedagogical employee of the department" functional task. This task was implemented as a separate IT product for the development of the capabilities of the "University" information and analytical system of the Kharkiv

National University of Radio Electronics. Previously, this system implemented the "Distribution of educational load between teachers of the department" functional task, the main source document of which is one of the sections of the "Individual plan of a scientific and pedagogical employee of the department" document.

A detailed description of the names and designations of the functions and data flows forming separate CIs is given in the **Table 1.1** [16, 17]. The abbreviations adopted in **Table 1.1**: IP – individual plan; KPI – key performance indicators. As numerical numbers in the **Table 1.1** shows the numbers of works, input and output flows that were generated by the AllFusion Process Modeler CASE tool during the creation of the data flow diagram.

● **Table 1.1** Description of the configuration items of the "Formation and management of the individual plan of the scientific and pedagogical employee of the department" functional task (based on the data flow diagram)

Work		**Input flow**		**Output flow**	
No.	**Name**	**No.**	**Name**	**No.**	**Name**
1	**2**	**3**	**4**	**5**	**6**
CI1	Conversion of the "Educational work" section	1	The teacher's educational load for the academic year	2	Information from the IP section "Educational work"
CI2	Formation of the "Scientific work" section	2	Information about the teacher	3	Information from the IP section "Scientific work"
		3	Information about planned work		
		5	Information about recommended work		
		8	Information from the IP section "Scientific work"		
		12	Remaining hours		
CI3	Formation of the "Methodical work" section	2	Information about the teacher	4	Information from the IP section "Methodical work"
		3	Information about planned work		
		5	Information about recommended work		
		9	Information from the IP section "Methodical work"		
		12	Remaining hours		
CI4	Formation of the "Organizational Work" section	2	Information about the teacher	5	Information from the IP section "Organizational work"
		3	Information about planned work		
		5	Information about recommended work		
		10	Information from the IP section "Organizational work"		
		12	Remaining hours		

Continuation of Table 1.1

1	2	3	4	5	6
CI5	Formation of the first list of positions and long-term assignments	4 11	Information about positions and long-term assignments Information from the IP section "List of positions and long-term assignments"	6	Information from the IP section "List of positions and long-term assignments"
CI6	Formation of the list of recommended works	5	Information about recommended work	1	Information about recommended work
CI7	Formation and maintenance of normative and reference information about KPI	6	Information about the department's key KPIs	7	Information about the department's key KPIs
CI8	Formation of the teacher's KPI and part of the department's KPI	8	Information from the IP section "Scientific work"	9	Information about the teacher's KPI and parts of the department's KPI
CI9	Forming a summary table for the academic year	9 7 8 10	Information from the IP section "Methodical work" Information from the IP section "Educational work" Information from the IP section "Scientific work" Information from the IP section "Organizational work"	8	Information about the number of hours by IP sections
CI10	Formation of the source document "IP"	9 7 8 10 11	Information from the IP section "Methodical work" Information from the IP section "Educational work" Information from the IP section "Scientific work" Information from the IP section "Organizational work" Information from the IP section "List of positions and long-term assignments"	10	IP

The entity description of the "Formation and maintenance of an individual plan of a scientific and pedagogical employee of the department" functional task was performed in the form of an ER diagram, which is shown in **Fig. 1.1** [16]. The description of entity names and their numerical identifiers is given in the **Table 1.2** [16, 17]. As numerical identifiers in the **Table 1.2** indicate the entity numbers that were generated using the AllFusion ERwin Data Modeler CASE tool during the development of the one shown in **Fig. 1.1** ER diagrams and imported into the AllFusion Process Modeler CASE tool to link with the functional task data flow diagram.

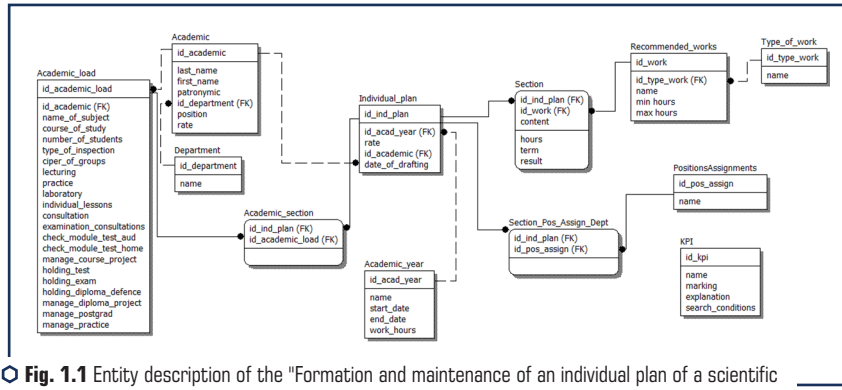

Fig. 1.1 Entity description of the "Formation and maintenance of an individual plan of a scientific and pedagogical employee of the department" functional task in the form of an ER diagram

Table 1.2 Set of descriptions of the entities of the "Formation and maintenance of an individual plan of a scientific and pedagogical employee of the department" functional task

Numerical identifier	Name
1	Academic_load
2	Academic
3	Department
4	Individual_plan
5	Academic_section
6	Academic_year
7	Section
8	Recommended_works
9	Type_of_work
10	Section_Pos_Assign_Dept
11	PositionsAssignments
12	KPI

Supplementing the data flow diagram of the "Formation and maintenance of an individual plan of a scientific and pedagogical employee of the department" functional task with information from ER diagrams allows to fix the corresponding subsets of entities with ER diagrams for each work, input and output flows of the data flow diagram. Therefore, it is proposed to consider each specific work of a data flow diagram with input and output data flows that are associated with this work as functional CIs [16, 17]. The received descriptions of the "Formation and management of the

individual plan of the scientific and pedagogical worker of the department" CI functional task are shown in the **Table 1.3** [17]. Items with the value "1" indicate the facts of the use of the entity with the identifier, which is given in the column of the **Table 1.3**, to describe the operation, input or output data flow with the identifier given in the row of the **Table 1.3**. Items with a value of "0" indicate the opposite facts.

● **Table 1.3** Descriptions of the configuration items of the "Formation and maintenance of an individual plan of a scientific and pedagogical employee of the department" functional task

Description of the CI function												
CI	**Entities IDs**											
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
CI1	1	1	1	1	1	1	0	0	0	0	0	0
CI2	1	1	1	1	1	1	1	1	1	0	0	0
CI3	1	1	1	1	1	1	1	1	1	0	0	0
CI4	1	1	1	1	1	1	1	1	1	0	0	0
CI5	0	1	0	1	0	1	1	0	0	1	1	0
CI6	0	0	0	0	0	0	0	1	1	0	0	0
CI7	0	0	0	0	0	0	0	0	0	0	0	1
CI8	1	1	0	1	1	1	1	1	1	0	0	1
CI9	1	1	0	1	1	1	1	1	1	0	0	0
CI10	1	1	0	1	1	1	1	1	1	1	1	0
Description of CI input data flows												
CI	**Entities IDs**											
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
CI1	1	1	1	0	0	0	0	0	0	0	0	0
CI2	1	1	1	1	1	1	1	1	1	0	0	0
CI3	1	1	1	1	1	1	1	1	1	0	0	0
CI4	1	1	1	1	1	1	1	1	1	0	0	0
CI5	0	1	0	1	0	1	1	0	0	1	1	0
CI6	0	0	0	0	0	0	0	1	1	0	0	0
CI7	0	0	0	0	0	0	0	0	0	0	0	1
CI8	1	1	0	1	0	1	1	1	1	0	0	0
CI9	1	1	0	1	1	1	1	1	1	0	0	0
CI10	1	1	0	1	1	1	1	1	1	1	1	0

● **Continuation of Table 1.3**

Description of CI output data flows												
CI	**Entities IDs**											
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
CI1	1	1	0	1	1	1	0	0	0	0	0	0
CI2	0	1	0	1	0	1	1	1	1	0	0	0
CI3	0	1	0	1	0	1	1	1	1	0	0	0
CI4	0	1	0	1	0	1	1	1	1	0	0	0
CI5	0	1	0	1	0	1	1	0	0	1	1	0
CI6	0	0	0	0	0	0	0	1	1	0	0	0
CI7	0	0	0	0	0	0	0	0	0	0	0	1
CI8	1	1	0	1	1	1	1	0	0	0	0	1
CI9	1	1	0	1	1	1	1	1	0	0	0	0
CI10	1	1	0	1	1	1	1	1	1	1	1	0

1.3.2 DESCRIPTION OF SOLVING THE PROBLEM OF IDENTIFYING CONFIGURATION ITEMS USING THE AGNES AGGLOMERATIVE CLUSTERING METHOD

As a result of the implementation of Stage 1, 10 clusters were formed, each of which contained one functional CI of the researched task. The distance matrix *D* was calculated for these clusters, which is shown in the **Table 1.4** [17]. Distances were calculated according to formula (1.5).

● **Table 1.4** Initial distance matrix *D* (AGNES method, nearest neighbor algorithm)

Clusters	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
C1	0	6	6	6	7	8	7	6	7	9
C2	6	0	0	0	7	7	10	4	3	4
C3	6	0	0	0	7	7	10	4	3	4
C4	6	0	0	0	7	7	10	4	3	4
C5	7	7	7	7	0	8	7	7	6	4
C6	8	7	7	7	8	0	3	7	7	9
C7	7	10	10	10	7	3	0	8	9	11
C8	6	4	4	4	7	7	8	0	2	4
C9	7	3	3	3	6	7	9	2	0	3
C10	9	4	4	4	4	9	11	4	3	0

As a result of the first iteration of Stage 2, a pair of clusters C2 and C3, closest to each other, was selected. The distance between these clusters is equal to 0. The choice was made when viewing the matrix of distances (**Table 1.4**) from left to right and from top to bottom. A new cluster C11 = {C2, C3} was formed [17].

During the first iteration of Stage 3, the distance matrix *D* was recalculated taking into account the presence of the new C11 cluster. The result of the calculation is shown in **Table 1.5** [17].

● **Table 1.5** Distance matrix *D* with cluster C11

Clusters	C1	C11	C4	C5	C6	C7	C8	C9	C10
C1	0	6	6	7	8	7	6	7	9
C11	6	0	0	7	7	10	4	3	4
C4	6	0	0	7	7	10	4	3	4
C5	7	7	7	0	8	7	7	6	4
C6	8	7	7	8	0	3	7	7	9
C7	7	10	10	7	3	0	8	9	11
C8	6	4	4	7	7	8	0	2	4
C9	7	3	3	6	7	9	2	0	3
C10	9	4	4	4	9	11	4	3	0

During the second iteration of Stage 2, a pair of clusters C11 and C4, which are closest to each other, were selected. The distance between these clusters is equal to 0. A new cluster C12 = {C11, C4} was formed.

During the first iteration of Stage 3, the distance matrix *D* was recalculated taking into account the presence of the new cluster C12. The result of the calculation is shown in **Table 1.6** [17].

● **Table 1.6** Distance matrix *D* with cluster C12

Clusters	C1	C12	C5	C6	C7	C8	C9	C10
C1	0	6	7	8	7	6	7	9
C12	6	0	7	7	10	4	3	4
C5	7	7	0	8	7	7	6	4
C6	8	7	8	0	3	7	7	9
C7	7	10	7	3	0	8	9	11
C8	6	4	7	7	8	0	2	4
C9	7	3	6	7	9	2	0	3
C10	9	4	4	9	11	4	3	0

During the third iteration of Stage 2, a pair of clusters C8 and C9, closest to each other, was selected. The distance between these clusters is equal to 2. A new cluster C13 = {C8, C9} was formed.

During the third iteration of Stage 3, the distance matrix *D* was recalculated taking into account the presence of the new C13 cluster. The result of the calculation is shown in **Table 1.7** [17].

● **Table 1.7** Distance matrix *D* with cluster C13

Clusters	C1	C12	C5	C6	C7	C13	C10
C1	0	6	7	8	7	6	9
C12	6	0	7	7	10	3	4
C5	7	7	0	8	7	6	4
C6	8	7	8	0	3	7	9
C7	7	10	7	3	0	8	11
C13	6	3	6	7	8	0	3
C10	9	4	4	9	11	3	0

During the fourth iteration of Stage 2, a pair of clusters C12 and C13 closest to each other was selected. The distance between these clusters is 3. A new cluster C14 = {C12, C13} was formed.

During the fourth iteration of Stage 3, the distance matrix *D* was recalculated, taking into account the presence of the new C14 cluster. The result of the calculation is shown in **Table 1.8** [17].

● **Table 1.8** Distance matrix *D* with cluster C14

Clusters	C1	C14	C5	C6	C7	C10
C1	0	6	7	8	7	9
C14	6	0	6	7	8	3
C5	7	6	0	8	7	4
C6	8	7	8	0	3	9
C7	7	8	7	3	0	11
C10	9	3	4	9	11	0

During the fifth iteration of Stage 2, a pair of clusters C14 and C10 closest to each other was selected. The distance between these clusters is 3. A new cluster C15 = {C14, C10} was formed.

During the fifth iteration of Stage 3, the distance matrix *D* was recalculated taking into account the presence of the new C15 cluster. The result of the calculation is shown in **Table 1.9** [17].

● **Table 1.9** Distance matrix *D* with cluster C15

Clusters	C1	C15	C5	C6	C7
C1	0	6	7	8	7
C15	6	0	4	7	8
C5	7	4	0	8	7
C6	8	7	8	0	3
C7	7	8	7	3	0

During the execution of the sixth iteration of Stage 2, a pair of clusters C6 and C7, which are closest to each other, were selected. The distance between these clusters is 3. A new cluster C16 = {C6, C7} was formed.

During the execution of the sixth iteration of Step 3, the distance matrix *D* was recalculated taking into account the presence of the new C16 cluster. The result of the calculation is shown in **Table 1.10** [17].

● **Table 1.10** Distance matrix *D* with cluster C16

Clusters	C1	C15	C5	C16
C1	0	6	7	7
C15	6	0	4	7
C5	7	4	0	7
C16	7	7	7	0

During the execution of the seventh iteration of Stage 2, a pair of clusters C15 and C5, closest to each other, was selected. The distance between these clusters is 4. A new cluster C17 = {C15, C5} was formed.

During the execution of the seventh iteration of Stage 3, the distance matrix *D* was recalculated, taking into account the presence of the new C17 cluster. The result of the calculation is shown in **Table 1.11** [17].

● **Table 1.11** Distance matrix *D* with cluster C17

Clusters	C1	C17	C16
C1	0	7	7
C17	7	0	7
C16	7	7	0

During the execution of the eighth iteration of Stage 2, a pair of clusters C1 and C17, closest to each other, was selected. The distance between these clusters is 7. A new cluster C18 = {C1, C17} was formed.

During the eighth iteration of Stage 3, the distance matrix *D* was recalculated taking into account the presence of a new cluster C18. The result of the calculation is shown in **Table 1.12** [17].

Table 1.12 Distance matrix *D* with cluster C18

Clusters	C18	C16
C18	0	7
C16	7	0

During the ninth iteration of Stage 2, cluster C19 was formed on the basis of clusters C18 and C16, which includes all the original clusters. This completes the execution of the nearest neighbor algorithm.

The result of applying the AGNES agglomerative clustering method (using the nearest neighbor algorithm) is a dendrogram that looks like the one shown in **Fig. 1.2** [17].

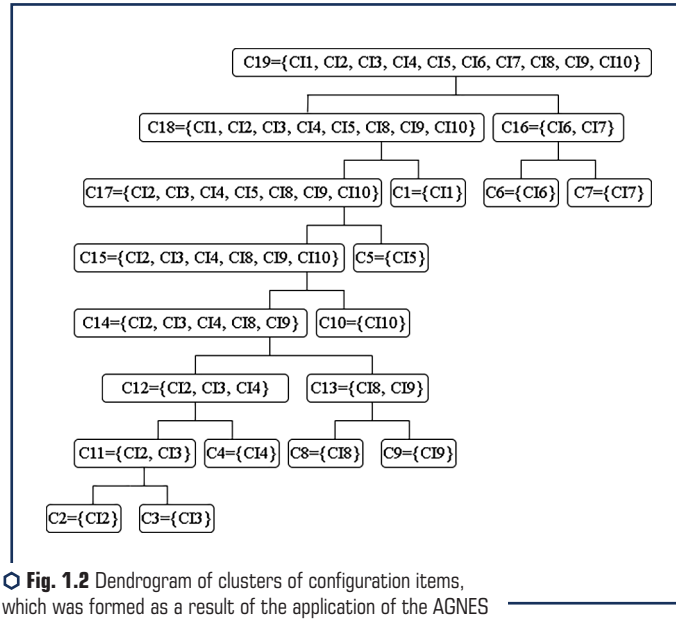

Fig. 1.2 Dendrogram of clusters of configuration items, which was formed as a result of the application of the AGNES agglomerative clustering method

1.3.3 DESCRIPTION OF THE SOLUTION TO THE PROBLEM OF IDENTIFYING CONFIGURATION ITEMS USING THE DIANA DIVISIVE CLUSTERING METHOD

Let's consider the features of using the DIANA divisive clustering method on the example of using this method to decompose the initial cluster C1, which includes all ten CIs defined in **Table 1.1** and **Table 1.3**. During Stage 1 and Stage 2, the value of the modified Chebyshev distance (1.5) was calculated for each pair of items of the initial cluster C1. These values are given in **Table 1.13** [16].

Table 1.13 Values of modified Chebyshev distances for each pair of items of the initial cluster C1

CI	CI1	CI2	CI3	CI4	CI5	CI6	CI7	CI8	CI9	CI10
CI1	0	6	6	6	7	8	7	6	7	9
CI2	6	0	0	0	7	7	10	4	3	4
CI3	6	0	0	0	7	7	10	4	3	4
CI4	6	0	0	0	7	7	10	4	3	4
CI5	7	7	7	7	0	8	7	7	6	4
CI6	8	7	7	7	8	0	3	7	7	9
CI7	7	10	10	10	7	3	0	8	9	11
CI8	6	4	4	4	7	7	8	0	2	4
CI9	7	3	3	3	6	7	9	2	0	3
CI10	9	4	4	4	4	9	11	4	3	0

During Stage 2, the value of the average modified Chebyshev distance was determined for each item of the initial cluster C1. These values are given in **Table 1.14** [16].

Table 1.14 Values of average modified Chebyshev distances for each item of cluster C1

CI	CI1	CI2	CI3	CI4	CI5	CI6	CI7	CI8	CI9	CI10
Average distance	6.2	4.1	4.1	4.1	6	6.3	7.5	4.6	4.3	5.2

As a result of Stage 2, item CI7 was selected. As a result of Stage 3, this item was excluded from the original cluster C1 and included in the formed child cluster C2.

During Step 4, new mean modified Chebyshev distances were determined for each item remaining to be considered in cluster C1, as well as the difference between these distances. The performance results are shown in **Table 1.15** [16].

As a result of Stage 5, the item CI6 was transferred to the daughter cluster C2. After that, the calculations of Stage 4 were repeated for items CI1, CI2, CI3, CI4, CI5, CI8, CI9 and C10, which remained in cluster C1. The results of these calculations are shown in the **Table 1.16** [16].

● **Table 1.15** Values of the difference between the average Chebyshev distances for each item of the cluster C1

Chebyshev distances										Average by items C1	Averages by items C2	Difference
CI	CI1	CI2	CI3	CI4	CI5	CI6	CI8	CI9	CI10			
CI1	0	6	6	6	7	8	6	7	9	6.11	7	–0.89
CI2	6	0	0	0	7	7	4	3	4	3.44	10	–6.56
CI3	6	0	0	0	7	7	4	3	4	3.44	10	–6.56
CI4	6	0	0	0	7	7	4	3	4	3.44	10	–6.56
CI5	7	7	7	7	0	8	7	6	4	5.89	7	–1.11
CI6	8	7	7	7	8	0	7	7	9	6.67	3	3.67
CI8	6	4	4	4	7	7	0	2	4	4.22	8	–3.78
CI9	7	3	3	3	6	7	2	0	3	3.78	9	–5.22
CI10	9	4	4	4	4	9	4	3	0	4.56	11	–6.44

● **Table 1.16** Values of the difference of the average Chebyshev distances for each item that remained in the cluster C1

Chebyshev distances									Average by items C1	Averages by items C2	Difference
CI	CI1	CI2	CI3	CI4	CI5	CI8	CI9	CI10			
CI1	0	6	6	6	7	6	7	9	5.875	7.5	–1.625
CI2	6	0	0	0	7	4	3	4	3	8.5	–5.5
CI3	6	0	0	0	7	4	3	4	3	8.5	–5.5
CI4	6	0	0	0	7	4	3	4	3	8.5	–5.5
CI5	7	7	7	7	0	7	6	4	5.625	7.5	–1.875
CI8	6	4	4	4	7	0	2	4	3.875	7.5	–3.625
CI9	7	3	3	3	6	2	0	3	3.375	8	–4.625
CI10	9	4	4	4	4	4	3	0	4	10	–6

Since all the differences in the **Table 1.8** are negative, then a new child class C3 was formed. This class includes items CI1, CI2, CI3, CI4, CI5, CI8, CI9 and CI10.

As a result of performing Step 6 of the method, it was established that the stopping conditions for clusters C2 and C3 are not fulfilled. Each of these clusters is not a singleton, and the distances between the items of each of these clusters are not equal to 0. This concludes the first iteration of the DIANA method. For the second iteration of the method, the proposed cluster is C2.

As a result of iterative execution of the DIANA method for the given problem, a dendrogram of clusters was formed, which is shown in **Fig. 1.3** [16].

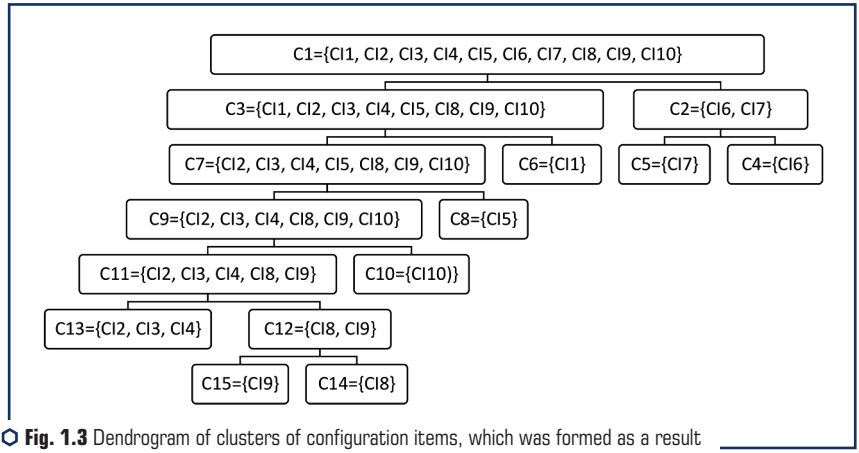

Fig. 1.3 Dendrogram of clusters of configuration items, which was formed as a result of applying the DIANA divisive clustering method

1.3.4 DESCRIPTION OF SOLVING THE PROBLEM OF IDENTIFYING CONFIGURATION ITEMS USING THE K-MEANS CLUSTERING ALGORITHM

Before proceeding with the task of analyzing the configuration of an IT product using the k-means algorithm, let's assume that the number of teams performing this IT project is three. Therefore, as centers for the initial division of the description of the CI of the functional problem considered in 1.3.1 into clusters, vectors describing the items of CI1, CI3 and CI7 are proposed. The choice of CI data is driven by the following considerations:

a) CI1 describes the function of forming the "Educational Work" section, which is an important initial item for drawing up the final individual plan of the employee;

b) CI3 describes the function "Formation of the "Methodical work" section ", which is an example of a number of typical functions for the formation of separate sections of the individual plan of the employee;

c) CI7 describes the "Formation and maintenance of normative reference information about KPI" function, which is the most isolated from the description of the final individual plan of the employee.

The value of the accuracy coefficient δ in the k-means algorithm is 0.1.

In the course of preliminary calculations, a matrix of values of modified Chebyshev distances (1.5) between individual CIs of the functional problem is formed. This matrix is shown in **Table 1.17**.

Based on the data in the **Table 1.17**, the matrix of CI functional task membership to the clusters of the initial partition (partition matrix) is calculated. This matrix is shown in **Table 1.18**.

Since the distances between CI5 and the centers of the initial partition were the same, the membership of CI5 to the cluster with the center CI1 was determined by the first comparison from left to right during the analysis of the distance matrix (**Table 1.17**).

Table 1.17 Matrix of values of modified Chebyshev distances between individual configurational items of the functional problem (k-means algorithm)

Configuration items	CI1	CI2	CI3	CI4	CI5	CI6	CI7	CI8	CI9	CI10
CI1	0	6	6	6	7	8	7	6	7	9
CI2	6	0	0	0	7	7	10	4	3	4
CI3	6	0	0	0	7	7	10	4	3	4
CI4	6	0	0	0	7	7	10	4	3	4
CI5	7	7	7	7	0	8	7	7	6	4
CI6	8	7	7	7	8	0	3	7	7	9
CI7	7	10	10	10	7	3	0	8	9	11
CI8	6	4	4	4	7	7	8	0	2	4
CI9	7	3	3	3	6	7	9	2	0	3
CI10	9	4	4	4	4	9	11	4	3	0

Table 1.18 Partition matrix at the zeroth iteration of the k-means algorithm for a functional problem

Configuration items	CI1	CI3	CI7
CI1	1	0	0
CI2	0	1	0
CI3	0	1	0
CI4	0	1	0
CI5	1	0	0
CI6	0	0	1
CI7	0	0	1
CI8	0	1	0
CI9	0	1	0
CI10	0	1	0

Next, cluster centers were determined on the first iteration of the k-means algorithm. The calculation was carried out according to the formula [12–14]:

$$c_l^{(i)} = \sum_{j=1}^{d} u_{ij}^{(l-1)} \times m_j \Big/ \sum_{j=1}^{d} u_{ij}^{(l-1)}, 1 \le i \le c, \tag{1.6}$$

where $c_l^{(i)}$ – the designation of the center of the i-th cluster on the l-th iteration of the algorithm; $u_{ij}^{(l-1)}$ – designation of the item of the partition n matrix for the j-th CI and the i-th cluster at the (l–1)-th iteration of the algorithm; m_j – designation of the vector describing the j-th CI; c – the number of clusters selected on the (l–1)-th iteration of the algorithm. For this problem $c=3$.

During the calculations, new cluster centers were discovered, which were described as conventional items CI11, CI12 and CI13. The vectors of their descriptions are shown in **Table 1.19**.

● **Table 1.19** Vector descriptions of items of conditional configuration CI11, CI12 and CI13

Description of CI function												
CI	**Entities IDs**											
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
CI11	0.5	1	0.5	1	0.5	1	0.5	0	0	0.5	0.5	0
CI12	1	1	0.5	1	1	1	1	1	1	0.17	0.17	0.17
CI13	0	0	0	0	0	0	0	0.5	0.5	0	0	0.5
Description of CI input data flows												
CI	**Entities IDs**											
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
CI11	0.5	1	0.5	0.5	0	0.5	0.5	0	0	0.5	0.5	0
CI12	1	1	0.5	1	0.83	1	1	1	1	0.17	0.17	0
CI13	0	0	0	0	0	0	0	0.5	0.5	0	0	0.5
Description of CI output data flows												
CI	**Entities IDs**											
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
CI11	0.5	1	0	1	0.5	1	0.5	0	0	0.5	0.5	0
CI12	0.5	1	0	1	0.5	1	1	0.83	0.67	0.17	0.17	0.17
CI13	0	0	0	0	0	0	0	0.5	0.5	0	0	0.5

Then, during the first iteration of the k-means algorithm, the distance matrix was recalculated according to formula (1.5) taking into account the newly introduced cluster centers in the form of conditional items CI11, CI12 and CI13. To perform the modulo 2 summation operation, the fractional values of the coordinates of the cluster center vectors were rounded to the nearest whole number (0 or 1). The results of calculating the distances to the new cluster centers are shown in the **Table 1.20**.

Based on the values obtained from the **Table 1.18**, the partition matrix was updated. The result of the update is shown in **Table 1.21**.

To check the stopping condition of the k-means algorithm, the partition matrix at the first iteration (**Table 1.21**) was subtracted from the partition matrix at the zero iteration (**Table 1.18**) of the k-means algorithm. The result is a zero partition matrix. This means that the stopping condition of the k-means algorithm is fulfilled.

● **Table 1.20** Matrix of modified Chebyshev distances between configuration items and new centers of clusters CI11, CI12 and CI13 at the first iteration of the algorithm

Configuration items	CI11	CI12	CI13
CI1	5	6	9
CI2	6	2	8
CI3	6	2	8
CI4	6	2	8
CI5	3	7	9
CI6	11	7	1
CI7	10	10	2
CI8	6	3	8
CI9	6	1	8
CI10	4	3	9

● **Table 1.21** Partition matrix on the first iteration of the k-means algorithm for a functional problem

Configuration items	CI11	CI12	CI13
CI1	1	0	0
CI2	0	1	0
CI3	0	1	0
CI4	0	1	0
CI5	1	0	0
CI6	0	0	1
CI7	0	0	1
CI8	0	1	0
CI9	0	1	0
CI10	0	1	0

Thus, the solution to the problem of clustering using the k-means algorithm for CI of the considered functional problem will be three clusters consisting of the following items:

a) cluster 1 (C1), the items of which are CI1 and CI5 with the center in the conditional item CI11;

b) cluster 2 (C2), the items of which are CI2, CI3, CI4, CI8, CI9 and CI10 with the center in the conditional item CI12;

c) cluster 3 (C3), the items of which are CI6 and CI7 with the center in the conventional item CI13.

1.3.5 DESCRIPTION OF SOLVING THE PROBLEM OF IDENTIFYING CONFIGURATION ITEMS USING THE GRAPHOANALYTIC METHOD

Let's consider the solution of the problem of CI identification according to the conditions outlined in 1.3.1, by the method outlined in [10]. According to this method, it is proposed to transform the description of the studied system into a set of the following descriptions:

– description of data structures defined state variables;

– descriptions of operations performed on state variables.

At the same time, interactions between operations and state variables are divided into two main groups [10]:

– read operation of the state variable;

– recording of the value in the state variable by the operation.

Then in [10] it is proposed to transform the set of these descriptions into an undirected graph. The vertices of this graph represent state variables and operations, and the arcs represent interactions between operations and state variables. At the same time, each arc has one of the following weights:

– weight equal to 1, if reading is performed by the state variable operation;

– weight equal to 2, if the value operation is written to the state variable.

System decomposition is performed on the formed graph by selecting individual clusters. Each such cluster is a part of the graph that is connected to other parts by the minimum possible number of arcs with minimum weights. The rationale for this method of cluster selection is detailed in [10].

It should be noted that the method described in [10] is focused on the use of UML diagrams that describe the software system being created. In our case, the description of the functional problem is a DFD. Therefore, as a description of operations, let's use descriptions of works from the **Table 1.1**. As a description of the state variables, let's use the descriptions of the input and output flows from the **Table 1.1**. At the same time, let's accept the following assumption: the output flows of any work from the **Table 1.1** are the input streams of other works from the **Table 1.1** if the names of these streams match. The results of selection of operations and state variables of the "Formation and maintenance of an individual plan of a scientific and pedagogical employee of the department" functional task are shown in the **Table 1.22** [16]. The description of interactions between operations and state variables is given in the **Table 1.23** [16].

The result of graph construction according to the method described in [10] is shown in **Fig. 1.4** [16]. Its weight is indicated on each edge. At the same time, it is considered that the weight of the edge that describes both reading and writing is equal to 3 ($(r=1)+(w=2)=3$).

Most of the vertices of the constructed graph are strongly connected. This means that the studied description of the architecture of the functional task is highly monolithic. Therefore, during the selection of clusters on the constructed graph, the following will be formed:

– relatively small clusters that include one operation;

– one large cluster, which includes a large number of operations that will have to be implemented as a monolithic software module.

● **Table 1.22** Description of operations and state variables of the functional task (based on the data flow diagram)

Designation	Name
Operations	
O1	Conversion of the "Educational work" section
O2	Formation of the "Scientific work" section
O3	Formation of the "Methodical work" section
O4	Formation of the "Organizational work" section
O5	Formation of the first list of positions and long-term assignments
O6	Formation of the list of recommended works
O7	Formation and maintenance of normative and reference information about KPI
O8	Formation of the teacher's KPI and part of the department's KPI
O9	Formation of a summary table for the academic year
O10	Formation of the source document "IP"
State variables	
V1	Teaching load for the academic year
V2	Information about the teacher
V3	Information about planned work
V4	Information about positions and long-term assignments
V5	Information about recommended work
V6	Information about the department's key KPIs
V7	Information from the "Educational work" IP section
V8	Information from the "Scientific work" IP section
V9	Information from the "Methodical work" IP section
V10	Information from the "Organizational work" IP section
V11	Information from the "List of positions and long-term assignments" IP section
V12	Remaining hours
V13	Information about the number of hours by IP sections
V14	Information about the teacher's KPI and parts of the department's KPI
V15	IP

Table 1.23 Description of interactions between operations and state variables

Operations	State variables														
	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15
01	*r*	–	–	–	–	–	*w*	–	–	–	–	–	–	–	–
02	–	*r*	*r*	–	*r*	–	–	*r*, *w*	–	–	–	*r*	–	–	–
03	–	*r*	*r*	–	*r*	–	–	–	*r*, *w*	–	–	*r*	–	–	–
04	–	*r*	*r*	–	*r*	–	–	–	–	*r*, *w*	–	*r*	–	–	–
05	–	–	–	*r*		–	–	–	–	–	*r*, *w*	–	–	–	–
06	–	–	–	–	*r*, *w*	–	–	–	–	–	–	–	–	–	–
07	–	–	–	–	–	*r*, *w*	–	–	–	–	–	–	–	–	–
08	–	–	–	–	–	–	–	*r*	–	–	–	–	–	*w*	–
09	–	–	–	–	–	–	*r*	*r*	*r*	*r*	–	–	*w*	–	–
010	–	–	–	–	–	–	*r*	*r*	*r*	*r*	*r*	–	–	–	*w*

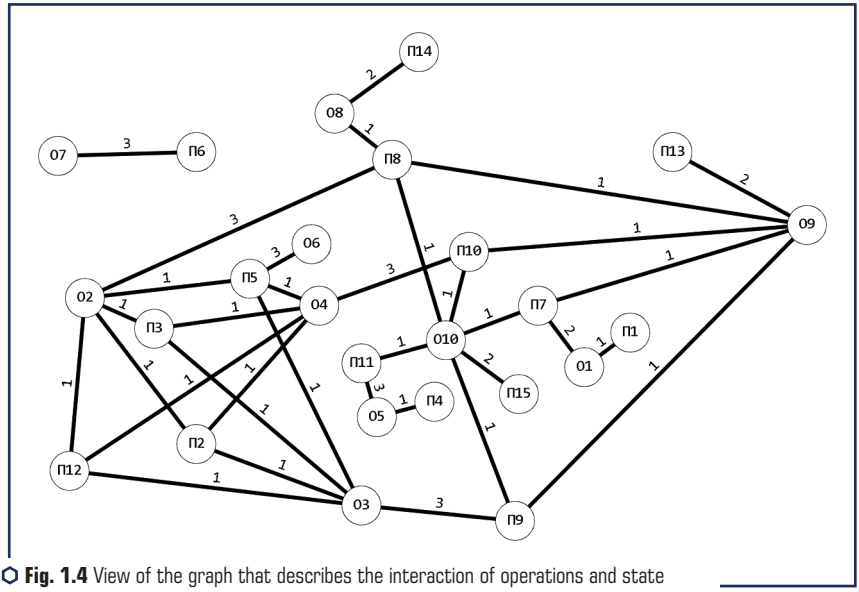

Fig. 1.4 View of the graph that describes the interaction of operations and state variables of the functional task

Solutions for selecting clusters on the constructed graph (**Fig. 1.4**) are given in **Table 1.24** [16].

The method described in [10] requires minimizing the number of intersecting arcs. However, the fulfillment of this condition (see solution 1 in **Table 1.24**) leads to issuing the task of creating

a large monolithic software module (cluster C2). Such a decision is inefficient for the further division of work between IT project implementation teams.

● **Table 1.24** Description of the results of solving the problem by the method described in [10]

Cluster designation	Designation of the vertices included in the cluster	Designation of functional task CIs that correspond to operations
1	**2**	**3**
Solution 1 (the number of cut arcs of minimum weight is 0)		
C1	O7, V6	CI7
C2	O1, O2, O3, O4, O5, O6, O8, O9, O10, V1, V2, V3, V4, V5, V7, V8, V9, V10, V11, V12, V13, V14, V15	CI1, CI2, CI3, CI4, CI5, CI6, CI8, CI9, CI10
Solution 2 (the number of cut arcs of minimum weight is 1)		
C1	O7, V6	CI7
C2	O8, V14	CI8
C3	O1, O2, O3, O4, O5, O6, O9, O10, V1, V2, V3, V4, V5, V7, V8, V9, V10, V11, V12, V13, V15	CI1, CI2, CI3, CI4, CI5, CI6, CI9, CI10
Solution 3 (the number of cut arcs of minimum weight is 1)		
C1	O7, V6	CI7
C2	O5, V4, V11	CI5
C3	O1, O2, O3, O4, O6, O8, O9, O10, V1, V2, V3, V5, V7, V8, V9, V10, V12, V13, V14, V15	CI1, CI2, CI3, CI4, CI6, CI8, CI9, CI10
Solution 4 (the number of cut arcs of minimum weight is 2)		
C1	O7, V6	CI7
C2	O8, V14	CI8
C3	O5, V4, V11	CI5
C4	O1, O2, O3, O4, O6, O9, O10, V1, V2, V3, V5, V7, V8, V9, V10, V12, V13, V15	CI1, CI2, CI3, CI4, CI6, CI9, CI10
Solution 5 (the number of cut arcs of minimum weight is 2)		
C1	O7, V6	CI7
C2	O1, V1, V7	CI1
C3	O2, O3, O4, O5, O6, O8, O9, O10, V2, V3, V4, V5, V8, V9, V10, V11, V12, V13, V14, V15	CI2, CI3, CI4, CI5, CI6, CI8, CI9, CI10
Solution 6 (the number of cut arcs of minimum weight is 3)		
C1	O7, V6	CI7
C2	O8, V14	CI8
C3	O1, V1, V7	CI1
C4	O2, O3, O4, O5, O6, O9, O10, V2, V3, V4, V5, V8, V9, V10, V11, V12, V13, V15	CI2, CI3, CI4, CI5, CI6, CI9, CI10

● **Continuation of Table 1.24**

1	2	3
Solution 7 (the number of cut arcs of minimum weight is 3)		
C1	O7, V6	CI7
C2	O5, V4, V11	CI5
C3	O1, V1, V7	CI1
C4	O2, O3, O4, O6, O8, O9, O10, V2, V3, V5, V8, V9, V10, V12, V13, V14, V15	CI2, CI3, CI4, CI6, CI8, CI9, CI10
Solution 8 (the number of cut arcs of minimum weight is 4)		
C1	O7, V6	CI7
C2	O8, V14	CI8
C3	O5, V4, V11	CI5
C4	O1, V1, V7	CI1
C5	O2, O3, O4, O6, O9, O10, V2, V3, V5, V8, V9, V10, V12, V13, V15	CI2, CI3, CI4, CI6, CI9, CI10
Solution 9 (the number of cut arcs of minimum weight is 4)		
C1	O7, V6	CI7
C2	O6, V5	CI6
C3	O1, O2, O3, O4, O5, O8, O9, O10, V1, V2, V3, V4, V7, V8, V9, V10, V11, V12, V13, V14, V15	CI1, CI2, CI3, CI4, CI5, CI8, CI9, CI10

In [10], the stopping condition for solving the clustering problem on the graphical description of the system architecture is not specified. So, the **Table 1.24** shows those division options for which the number of intersecting graph arcs does not exceed 4.

1.4 COMPARATIVE ANALYSIS OF THE OBTAINED RESULTS

Shown in **Fig. 1.2, 1.3** dendrograms obtained as a result of solving the problem by the AGNES agglomerative clustering method and DIANA divisive clustering method, respectively, coincide almost completely. An exception is clusters C2, C3, C4, and C11, selected during the construction of the dendrogram shown in **Fig. 1.2**. The appearance of these clusters is due to the fact that the AGNES agglomerative clustering method (using the nearest neighbor algorithm) is unable to recognize CIs which descriptions completely match. This shortcoming is the reason why the first two iterations of Stage 2 of the AGNES method were aimed at merging clusters that contained CIs with completely identical descriptions. In practice, this shortcoming of the AGNES method can lead to IT project executors being allocated separate sprints to implement CIs with overlapping

descriptions (in this case, CI2, CI3 and CI4) during the planning of their work. This will lead to an unjustified overestimation of labor costs and time spent on the implementation of these CIs.

To eliminate the mentioned shortcoming, it is proposed to adjust the AGNES agglomerative clustering method by supplementing it with operations to adjust the set of initial clusters formed at Stage 1. As a result of the addition, the modified AGNES agglomerative clustering method will be represented by a sequence of the following stages [16]:

Stage 1. The entire set CI is represented as a set of clusters C, each of which contains one item CI_i, $i=1,\ldots,n$, where n is the number of items in the set CI. Calculate the distance matrix D between the items of the set C.

Step 2. Calculate the distance matrix D between the items of the set of clusters C and adjust the set C by combining in a new cluster each pair of clusters c_p and c_q for which $d_{pq}=0$, then exclude the clusters c_p and c_q from further consideration and do not display them on the final dendrogram.

Stage 3. Select two clusters c_p and c_q, the distance between which will be minimal, and combine them into a new cluster c_r, entering it instead of clusters c_p and c_q into the set of clusters C.

Step 4. Calculate the values of the distance matrix D using rule (1.1).

Step 5. Repeat Step 2, Step 3, and Step 4 until one cluster is formed that includes all items of the CI set.

This modification makes it possible to eliminate the above-mentioned shortcoming and does not lead to a serious increase in the computational complexity of the AGNES method.

It should be noted that the AGNES method considered in the study (using the nearest neighbor algorithm) is less complex in terms of computational complexity than the DIANA divisive classification method. In particular, in order to obtain a similar result, the AGNES method does not require quite complex calculations regarding the calculation of average distances and the search for items transferred from the parent cluster to the newly created child cluster. Therefore, it is more appropriate to use agglomerative clustering methods in order to solve the problem of CI identification of an IT product (provided that the above-mentioned shortcoming is eliminated).

Let's compare the clusters formed as a result of the application of the k-means algorithm with the general parts of the dendrograms shown in **Fig. 1.2, 1.3**. As a representation of such a common part, let's use the dendrogram shown in **Fig. 1.3**. According to the results of this comparison, it should be noted:

– cluster C3 obtained as a result of applying the k-means algorithm coincides with cluster C2 highlighted on the dendrogram;

–cluster C2, obtained as a result of applying the k-means algorithm, coincides with the C9 cluster selected on the dendrogram;

– cluster C1, obtained as a result of applying the k-means algorithm, has no direct analogues on the dendrogram.

The absence of a direct analogue for the C1 cluster allows to claim that this cluster is artificial. To verify this statement, let's analyze the distance of individual CIs from the conditional centers of CI11, CI12, and CI13 clusters determined in the first iteration. For this purpose, let's use the data from **Table 1.20**. The results of the analysis are given in **Table 1.25**.

● **Table 1.25** The distance of individual configuration items from the conditional centers CI11, CI12 and CI13 of the clusters determined in the first iteration of the k-means algorithm

Chebyshev modified distance to the center	A list of CIs that are at the appropriate distance to the center
Conditional center CI11	
3	CI5 ∈ C1
4	CI10 ∈ C2
5	CI1 ∈ C1
6	(CI2, CI3, CI4, CI8, CI9) ∈ C2
10	CI7 ∈ C3
11	CI6 ∈ C3
Conditional center in CI12	
1	CI9 ∈ C2
2	(CI2, CI3, CI4) ∈ C2
3	(CI8, CI10) ∈ C2
6	CI1 ∈ C1
7	CI5 ∈ C1, CI6 ∈ C3
10	CI7 ∈ C3
Conditional center in CI13	
1	CI6 ∈ C3
2	CI7 ∈ C3
8	(CI2, CI3, CI4, CI8, CI9) ∈ C2
9	(CI1, CI5) ∈ C1, CI10 ∈ C2

From the **Table 1.25**, it can be seen that the initial decision to select a cluster centered on CI1 was rather erroneous. This is evidenced by the fact that CI10, which is part of cluster C2, is closer to the conventional center of cluster C1 than CI1, which is part of this cluster.

It would be appropriate to assign cluster centers at the beginning of the k-means algorithm, based on the results of the analysis of the matrix of distances between CIs. The purpose of such an analysis is to select the desired combination of CIs that are as far apart as possible. However, such a combination of clusters, identified at the beginning of the k-means algorithm, does not allow obtaining a set of clusters balanced by the main characteristics of the IT project (indicators of labor intensity, time, cost, and quality of project work). At the same time, the dendrogram of clusters formed as a result of the application of the DIANA method makes it possible to select CI clusters that are maximally balanced in terms of such characteristics in the future. Therefore, the use of the k-means algorithm to solve the problem of CI identification in the IT project is less appropriate than hierarchical clustering methods.

A comparison of the results of solving the problem of CI identification in the IT project using the methods of hierarchical clustering AGNES and DIANA and the k-means algorithm with the results obtained using the graphoanalytic method is given in the **Table 1.26**.

● **Table 1.26** Comparison of the results of solving the task of identifying functional configuration items using the graph-analytic method, the DIANA method, and the k-means algorithm

Designation of the cluster by the graphoanalytical method	Designation of CIs of the functional problem, which are included in the cluster according to the graphoanalytic method	Designation of a similar cluster on the dendrogram (Fig. 1.3)	Designation of a similar cluster based on the results of the k-means algorithm
1	**2**	**3**	**4**
Solution 1 (the number of cut arcs of minimum weight is 0)			
C1	CI7	C5	There is no complete analogue, the closest is C3
C2	CI1, CI2, CI3, CI4, CI5, CI6, CI8, CI9, CI10	There is no complete analogue, the closest is C3	There is no complete analogue, the closest is C2
Solution 2 (the number of cut arcs of minimum weight is 1)			
C1	CI7	C5	There is no complete analogue, the closest is C3
C2	CI8	C14	There is no analogue
C3	CI1, CI2, CI3, CI4, CI5, CI6, CI9, CI10	There is no complete analogue, the closest is C3	There is no complete analogue, the closest is C2
Solution 3 (the number of cut arcs of minimum weight is 1)			
C1	CI7	C5	There is no complete analogue, the closest is C3
C2	CI5	C8	There is no complete analogue, the closest is C1
C3	CI1, CI2, CI3, CI4, CI6, CI8, CI9, CI10	There is no complete analogue, the closest is C3	There is no complete analogue, the closest is C2
Solution 4 (the number of cut arcs of minimum weight is 2)			
C1	CI7	C5	There is no complete analogue, the closest is C3
C2	CI8	C14	There is no analogue
C3	CI5	C8	There is no complete analogue, the closest is C3
C4	CI1, CI2, CI3, CI4, CI6, CI9, CI10	There is no complete analogue, the closest are C3, C7, C9	There is no complete analogue, the closest is C2
Solution 5 (the number of cut arcs of minimum weight is 2)			
C1	CI7	C5	There is no complete analogue, the closest is C3
C2	CI1	C6	There is no complete analogue, the closest is C1
C3	CI2, CI3, CI4, CI5, CI6, CI8, CI9, CI10	There is no complete analogue, the closest is C7	There is no complete analogue, the closest is C2

● **Continuation of Table 1.26**

1	2	3	4
Solution 6 (the number of cut arcs of minimum weight is 3)			
C1	CI7	C5	There is no complete analogue, the closest is C3
C2	CI8	C14	There is no analogue
C3	CI1	C6	There is no complete analogue, the closest is C1
C4	CI2, CI3, CI4, CI5, CI6, CI9, CI10	There is no complete analogue, the closest is C7	There is no complete analogue, the closest is C2
Solution 7 (the number of cut arcs of minimum weight is 3)			
C1	CI7	C5	There is no complete analogue, the closest is C3
C2	CI5	C8	There is no complete analogue, the closest is C1
C3	CI1	C6	There is no complete analogue, the closest is C1
C4	CI2, CI3, CI4, CI6, CI8, CI9, CI10	There is no complete analogue, the closest is C9	There is no complete analogue, the closest is C2
Solution 8 (the number of cut arcs of minimum weight is 4)			
C1	CI7	C5	There is no complete analogue, the closest is C3
C2	CI8	C14	There is no analogue
C3	CI5	C8	There is no complete analogue, the closest is C3
C4	CI1	C6	There is no complete analogue, the closest is C1
C5	CI2, CI3, CI4, CI6, CI9, CI10	There is no complete analogue, the closest is C9	There is no complete analogue, the closest is C2
Solution 9 (the number of cut arcs of minimum weight is 4)			
C1	CI7	C5	There is no complete analogue, the closest is C3
C2	CI6	C4	There is no complete analogue, the closest is C3
C3	CI1, CI2, CI3, CI4, CI5, CI8, CI9, CI10	C3	There is no complete analogue, the closest is C2

The following conclusions can be drawn based on the data from the **Table 1.26**:

a) when selecting clusters consisting of one CI, hierarchical clustering methods and the method described in [10] give the same results;

b) when selecting clusters consisting of several CIs, the method described in [10] is less accurate than hierarchical clustering methods;

c) with an increase in the number of connections between CIs, which must be cut when decomposing the graph into separate clusters, hierarchical clustering methods and the method outlined in [10] generally give similar results (solution 9 in **Table 1.26** completely coincides with clusters C3, C4 and C5, highlighted in **Fig. 1.3**);

d) the results obtained using the k-means method and algorithm described in [10] practically do not coincide with each other.

It should be taken into account that the increase in the number of connections between CIs, which must be cut when decomposing the graph into separate clusters, leads to a further increase in problems when integrating the system from separate CIs. Therefore, the methods of hierarchical clustering should be recognized as the best for solving the problem of CI identification in the IT project of creating a new IT product, in particular – IS. In case of re-planning of an ongoing IT project due to changes, the method described in [10] may be more convenient.

CONCLUSIONS

It has been proposed to use methods of hierarchical and non-hierarchical clustering to investigate the peculiarities of solving the problem of CI identification. As examples of hierarchical clustering methods, it has been proposed to use the AGNES agglomerative clustering method (using the nearest neighbor algorithm) and the DIANA divisive clustering method. As an example of non-hierarchical clustering methods, it has been proposed to use the k-means algorithm. In addition, it has been proposed to use the grapho-analytical clustering method considered in [10] to compare the course and results of solving the problem of CI identification. This method was recommended in [10] to solve the decomposition problem of the architecture of a monolithic software system into separate services.

It has been proposed to use the description of the architecture of the "Formation and maintenance of an individual plan of a scientific and pedagogical employee of the department" functional task as the initial data in the study. This description is quite suitable for presenting the description of the system architecture at the level of individual functional CIs. In this case, descriptions of individual functions of this task act as such CIs. These descriptions, in turn, consist of descriptions of the corresponding functions, as well as the input and output data streams of these functions. The use of the proposed description of the architecture makes it possible to establish differences in the course and results of the application of the researched clustering methods to solve the problem of identifying functional CIs.

The process of solving the problem of identifying functional CIs using the selected clustering methods has been considered. Based on the results of this review, it can be concluded that the simplest of the selected methods is the AGNES agglomerative clustering method (using the nearest neighbor algorithm). However, this statement requires further research, since it is not known how the procedure for solving the above-mentioned problem will change in the case of using other

methods and algorithms (for example, Ward's method). In general, it should be noted that with a small amount of initial data, the computational complexity of the considered clustering methods is approximately the same and does not allow to reasonably choose the best of these methods.

A comparison of the results of solving the task of identifying functional CIs using the selected clustering methods allows to establish that the results of the application of the DIANA method should be considered the most accurate. Using the method of divisive clustering makes it possible to single out in a separate cluster those functional Cis, which descriptions completely match each other. This selection allows to avoid the mistake of assigning the most similar functional CIs to different teams of IT project executors in the future. A modification of the AGNES method has been developed to provide a similar possibility of presenting the results of solving the clustering problem.

It should be noted that the use of hierarchical clustering methods to solve the problem should be considered a better alternative. Such a point of view makes it possible to further consider the task of assigning a list of functional CIs to individual teams of IT project executors that require implementation as a sequence of individual single-criteria optimization tasks. An attempt to establish such lists of functional CIs based on the a priori known number of executive teams (using the k-means algorithm) leads to the formation of individual lists, the similarity of items of which will be sufficiently artificial. However, to verify this conclusion, it is necessary to conduct further research using more complex and modern clustering algorithms.

REFERENCES

1. Bourque, P., Fairley, R. E. (Eds.) (2014). Guide to the Software Engineering Body of Knowledge. Version 3.0. IEEE Computer Society.
2. ISO/IEC/IEEE International Standard – Systems and software engineering – System life cycle processes: ISO/IEC/IEEE 15288:2015 (2015). IEEE. https://doi.org/10.1109/ieeestd.2015.7106435
3. Levykin, V. M., Ievlanov, M. V., Kernosov, M. A. (2014). Pattern planning of requirements to the informative systems: design and application: monograph. Kharkiv: The "Kompaniya "Smit LTD".
4. 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
5. Fritzsch, J., Bogner, J., Zimmermann, A., Wagner, S. (2019). From monolith to microservices: A classification of refactoring approaches. 1st International Workshop on Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment, DEVOPS 2018, 128–141. https://doi.org/10.1007/978-3-030-06019-0_10
6. Sellami, Kh., Saied, M. A., Ouni, A. (2022). A Hierarchical DBSCAN Method for Extracting Microservices from Monolithic Applications. 2022 ACM International Conference on Eva-

luation and Assessment in Software Engineering, EASE 2022, 201–210. https://doi.org/10.1145/3530019.3530040
7. 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 2020, 9–16. https://doi.org/10.1109/icsa-c50368.2020.00011
8. Reiff-Marganiec, S., Tilly, M. (Eds.) (2012). Handbook of Research on Service-Oriented Systems and Non-Functional Properties: Future Directions. IGI Global. https://doi.org/10.4018/978-1-61350-432-1
9. Shahin, R. (2021). Towards Assurance-Driven Architectural Decomposition of Software Systems. 40th International Conference on Computer Safety, Reliability and Security, SAFECOMP 2021 held in conjunction with Workshops on DECSoS, MAPSOD, DepDevOps, USDAI and WAISE 2021, 187–196. https://doi.org/10.1007/978-3-030-83906-2_15
10. Faitelson, D., Heinrich, R., Tyszberowicz, Sh. (2017). From monolith to microservices: Supporting software architecture evolution by functional decomposition. 5th International Conference on Model-Driven Engineering and Software Development, MODELSWARD 2017, 435–442. https://doi.org/10.5220/0006206204350442
11. 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
12. Han, J., Kamber, M., Pei, J. (2012). Data Mining. Concepts and Techniques. Waltham: Morgan Kaufmann Publishers. https://doi.org/10.1016/c2009-0-61819-5
13. Barseghyan, A. A., Kupriyanov, M. S. Kholod, I. I., Tess, M. D., Elizarov, S. I. (2009). Analiz dannykh i protcessov. Saint-Petersburg: BHV-Petersburg, 512.
14. Kaufman, L., Rousseeuw, P. J. (2005). Finding Groups in Data. Introduction to Cluster Analysis. John Wiley & Sons, Inc.
15. Wierzchoń, S., Kłopotek, M. (2018). Modern Algorithms of Cluster Analysis. Cham: Springer. https://doi.org/10.1007/978-3-319-69308-8
16. 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
17. Vasiltcova, N., Panforova, I. (2022). Research on the use of hierarchical clustering methods when solving the task of IT product configuration analysis. Management Information Systems and Devices, 178, 37–49. Available at: https://www.ewdtest.com/asu/wp-content/uploads/2024/01/ASUiPA_178_37_49.pdf