UDC 004.728:519.87 **doi: 10.20998/2522-9052.2018.2.20**

V. Donets, N. Kuchuk, S. Shmatkov

V. N. Karazin Kharkiv National University, Kharkiv, Ukraine

DEVELOPMENT OF SOFTWARE OF E-LEARNING INFORMATION SYSTEM SYNTHESIS MODELING PROCESS

The article reflects the results of the study of the created model of the information structure of the system of support. The model allowed to carry out the process of modeling the synthesis of the information system e-learning. A program with a graphical user interface for the synthesis model of the e-learning information system was used for modeling. The program uses the parameters of the hyperconvergent base network and the e-learning system as the input. The result of the synthesis is the optimal placement of users, applications and data blocks by the nodes of the base network. This takes into account the capacity of the system's transactions. This optimizes the capacity of nodes. As a result, the effiency support of the e-learning system is increasing. The structure of the hyperconvergent base network e-learning support network is considered as the main factor that affects the quality of the system's requests. Therefore, it is important to analyze the structure when choosing options for building a hyperconvergent base network and its management. The main purpose of the structure analysis is to determine the parameters of the data streams in the network communication channels. The obtained results are necessary for an adequate estimation of network channels and nodes capacity. Data streams form e-learning tasks, which use applications that launch on network nodes and generate network traffic.

Keywords: e-learning, modeling; hyperconverting network; genetic algorithm; software.

1. Setting objectives

In accordance with the developed model of the information structure of the system e-learning [1], deployed on a hyperconverting server the criterion for evaluating this structure is the balance of the load on the nodes. Indicator function is the sum of deviations of nodes loading from the average load. The lower the download variance index, the higher the quality performance of the e-learning system. With the aim of obtaining the most effective structure of the basic support network e-learning and adding to this knowledge about the genetic algorithm we can say about indicator of the efficiency of the structure of the basic network is the value of the fitness function in the genetic algorithm.

From here we have input data. This is information about the number of applications (A), users (U), transactions (E), nodes (N) and data stores (D) functioning in the e-learning system. We also need information about transactions. There are some applications ($\overline{a_e} = (a_{e1},...,a_{eA})$) and data stores ($\overline{d_e} = (d_{e1},...,d_{eD})$)are used. Also the order of application launch ($\overline{W_e} = (w_{ij})$). But this is not enough. We need to have an idea of the functioning of each application in a particular transaction. This requires knowledge of the volumes of data exchange ($A_{ae} = \left\langle \overline{\lambda_{ae}}, \overline{\beta_{ae}} \right\rangle$, де $\overline{\lambda}_{ae} = (\lambda_{ae1},...,\lambda_{aeD})$ і $\beta_{ae} = (\beta_{ae1,...,}\beta_{aeA})$). In order to simulate the real functioning of the system. With its user interaction, we need the user-initiated transaction intensities.

After working the genetic algorithm with the original data which will characterize the structure of the basic support e-learning network will accommodate users (H), applications (G), data stores (S) by the nodes of the network. All this together will form a tuple model of the information system e-learning:

$$\Im = \left\langle M_U, M_N, M_E, M_D, \{E_e\}, \{A_{ae}\}, G, H, S \right\rangle \text{ [1].}$$

2. The algorithm of the synthesis of the e-learning system

The genetic algorithm is based on the algorithm for the synthesis of the model of the information structure of the e-learning system. A stop for the algorithm will be the condition of the number of generations. Or approximation to the objective value of a fitness function.

It remains to look at the algorithm for calculating the value of the fitness function. Having all the necessary data, we will do the following

1. By the intensity matrix of the start of transactions by users

$$\left(\Omega = \left(\omega_{i,j}\right); \omega_{i,j} \ge 0; i = 1,...,U; j = 1,...E\right)$$

total intensity of requests to start the j-th transaction:

$$\overline{\omega} = \left(\omega_{i,j}\right), \omega_{i,j} = \sum\nolimits_{i}^{U} = 1\omega_{i,j} \qquad (1)$$

2. Find the total intensity of k-th application start-up (k = 1, …, A) all system transactions, here $a_{j,k}$ – вoolean vector of required applications for j-th transaction:

$$\theta = \left(\theta_k\right), \theta_k = \sum\nolimits_{j=1}^{E} \omega_j \alpha_{j,k} \qquad (2)$$

3. functioning of j-th transaction on information nodes n and m (n, m = 1, …, N):

$$. \quad Z_{jnm1} = \sum\nolimits_{\xi=1}^{A} \alpha_{j,\xi} g_{\xi,n} \left(\sum\nolimits_{\eta=1}^{D} \alpha_{j,\eta} g_{\eta,m} \beta_{\xi,\eta} \right) \qquad (3)$$

4. Define for (j, n, m) the volumetric data that are transmitted from the data store:

$$Z_{jnm2} = \sum\nolimits_{\xi=1}^{A} \alpha_{j,\xi} g_{\xi,n} \left(\sum\nolimits_{\eta=1}^{D} d_{j,\eta} s_{\eta,m} \lambda_{\xi,\eta} \right) \qquad (4)$$

5. Then the volumetric data in the network can be determined by the formula:

$$Z = \left(Z_{n,m}\right), Z_{n,m} = \sum\nolimits_{j=1}^{E}\left(Z_{jnm1} + Z_{jnm2}\right) \ . \qquad (5)$$

6. We construct the intensity matrix of the exchange between the information network nodes runtime the j-th transaction:

$$C_j = \left(C_{j,n,m}\right), C_{j,n,m} = \omega_j Z_{j,n,m}. \qquad (6)$$

7. The loading matrix for information nodes is determined by the formula:

$$C = \left(C_{n,m}\right), C_{n,m} = \sum\nolimits_{j=1}^{A} C_{j,n,m} \qquad (7)$$

8. Determine the average load of one network node:

$$C_{cp} = \left(\sum\nolimits_{n=1}^{N}\sum\nolimits_{m=1}^{N} C_{n,m}\right) / N . \qquad (8)$$

9. Define the characteristic of the system. The criterion is the load balancing of the nodes. The indicator is the sum of the deviations of the load from the mean:

$$\Psi = \sum\nolimits_{n=1}^{N} \ \sum\nolimits_{m=1}^{N} \left|C_{n,m} - C_{cp.}\right| . \qquad (9)$$

10. So, we have a fitness function for the genetic algorithm:

$$\Psi \to \min \ . \qquad (10)$$

11. Also, we calculate the load of nodes in the network. The intensity of the request flow for running the i application installed on the node n:

$$B = \left(b_{i,n}\right) = \theta_{\text{діаг.}} G, b_{i,n} = \theta_i g_{i,n} \ . \qquad (11)$$

12. Loading nodes with data repository elements: j-transaction, n- node, d- data stores:

$$\phi_{j,n,d} = \theta_j S_{n,d} d_{j,n} \ . \qquad (12)$$

13. Then, to the data block d on node n, the intensity of requests to the data store is equal to:

$$\varphi = \left(\phi_{n,d}\right), \phi_{nd} = \sum\nolimits_{j=1}^{E} \phi_{j,n,d} \ . \qquad (13)$$

3. Diagram of options for using a computer program

Let's consider the usage diagrams. They describe the relationships and dependencies between the groups of use options and the participants. Which are involved in the process. Usage diagrams are not intended to display the project and can not describe the internal device system. Usage diagrams are designed to simplify interaction with users and clients. Especially useful for determining the necessary characteristics of the system. Usage diagrams indicate what the system should do without specifying the methods themselves [2, 3].

The use case describes a group of actions in the system. Which lead to a concrete result. Usage options are descriptions of typical interactions between system users and the system itself. They reflect the external interface of the system.

Specify the form of what the system should do. Rules at work:

- each use case refers to at least one user
- each use case has an initiator
- each use case leads to the corresponding result (the result with a "business value").

Use cases can also interact with other uses. The most common types of interaction include:

1. Enable - indicates that the use case is embedded in another use case.

2. Adding - indicates that in certain situations or at a specific point (called an extension point) the use case will be extended by another.

3. Generalization - indicates that the use case inherits the characteristics of the "parent" use case and can override some of them or add new ones, similar to inheritance in classes.

The person acting is an external source (not an element of the system) that interacts with the system through the use case. Actors can be either real people (for example, users of the system), or other computer systems or external events [4, 5].

Actors do not represent individuals or systems, but their roles. When a person interacts with the system in various ways. It is displayed by several characters. For example, a person who works in the customer service and receives orders from customers. It will be displayed in the system as a "member of the support department" and "member of the sales department" [6].

Considering the knowledge of the diagram of usage options and information about the obtained model of synthesis of the information system of e-learning. We will get a diagram of the use cases (Fig. 1). This will help in the future to develop a synthesis program.

4. A class diagram that was used in design

Class diagram - displays system classes. Their attributes, methods and interrelationships between them. There are different points of view on the construction of class diagrams. It depends on the purposes of their application [5, 7]:

- Conceptual point of view. The class diagram describes the model of the subject domain. Only application application classes are present in it.

- View point of the specification. A class diagram is used in the design of information systems.

- The point of view of implementation. The class diagram contains classes. They are used directly in the program code).

In the future, a viewpoint for implementation will be used to construct a class diagram. Which will describe the necessary classes and interrelationships between them.

To specify the visibility of members of a class, these notation must be placed before the participant's name [4, 7]:

«+» –public;
«-» –private;
«#» –protected;
«/» –derived;
«~» –package.

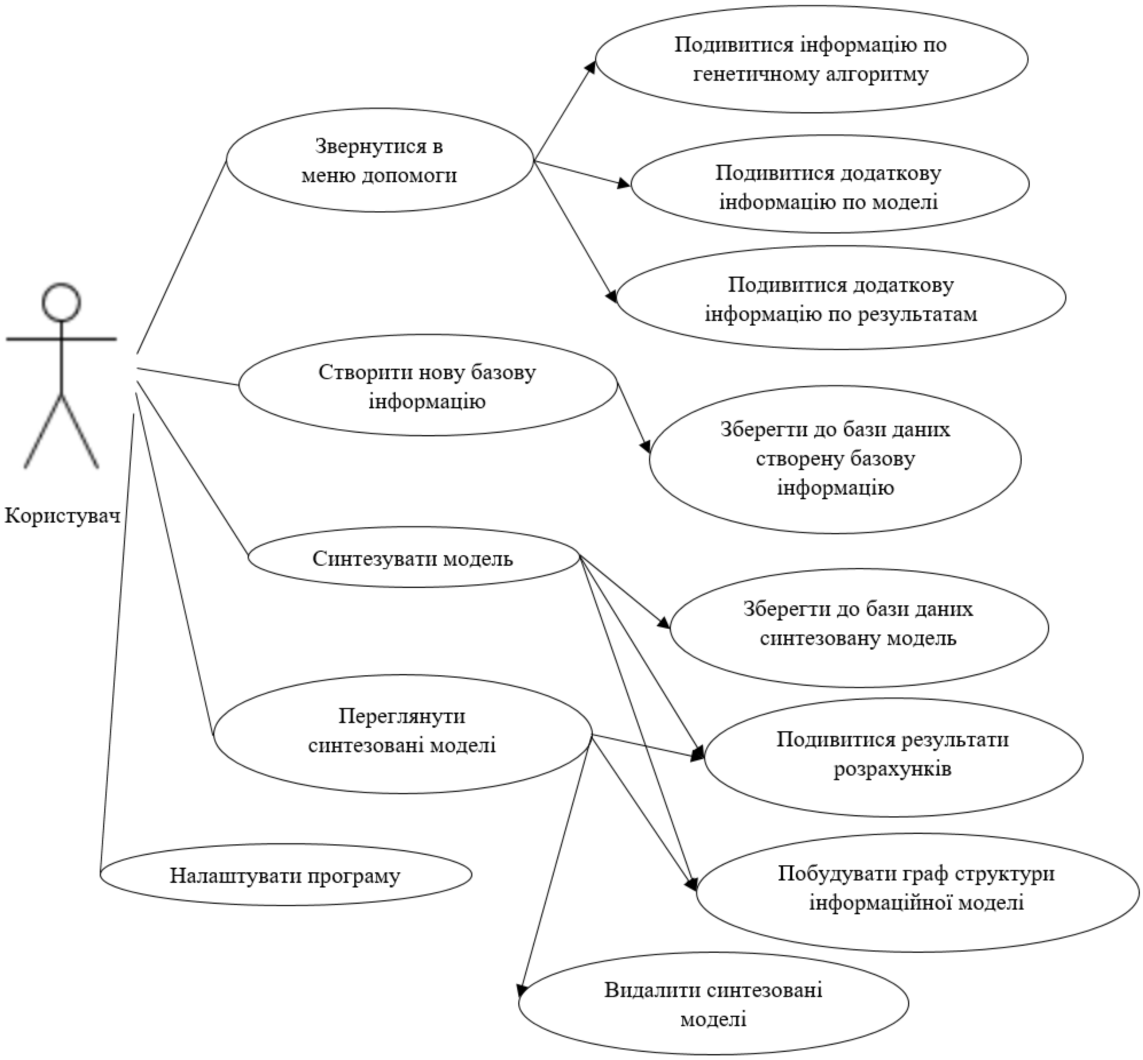

Користувач
Звернутися в меню допомоги
Подивитися інформацію по генетичному алгоритму
Подивитися додаткову інформацію по моделі
Подивитися додаткову інформацію по результатам
Створити нову базову інформацію
Зберегти до бази даних створену базову інформацію
Синтезувати модель
Зберегти до бази даних синтезовану модель
Переглянути синтезовані моделі
Подивитися результати розрахунків
Побудувати граф структури інформаційної моделі
Налаштувати програму
Видалити синтезовані моделі

Fig. 1. The developed diagram of the use of the program synthesis of the information system e-learning

Interconnection is a special type of logical relationship between entities, shown on the charts of classes and objects. Relationships in UML are depicted in fig. 2.

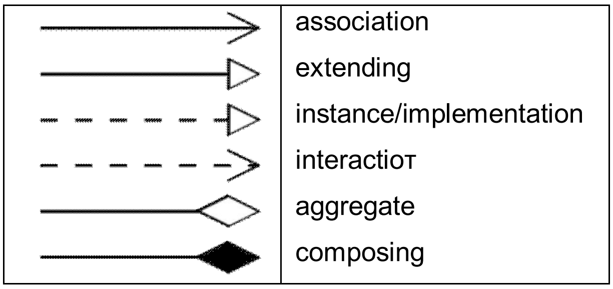


Fig. 2. Types of relations in UML

Having an idea of the class diagram and the structure of the model. With a genetic algorithm, we have a class diagram (Fig. 3). It depicts the classes necessary for the functioning of the genetic algorithm. We also need to store the basic information and results of the synthesis of the information structure of the e-learning system in the database. To do this, we have a diagram of the classes in Fig. 4. It is also necessary to arrange the graphical interface of the program. For user-friendly operation. In this case, the MVC programming pattern was used. This will allow you to make convenient transfers and extensions of the program. The corresponding diagram of classes is presented in Fig. 5

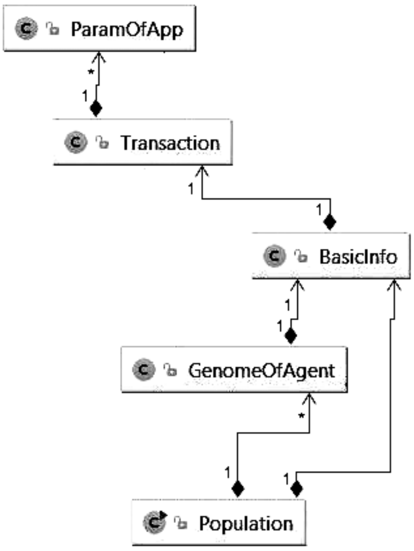


Fig. 3. Diagram of the classes necessary for the functioning of the genetic algorithm

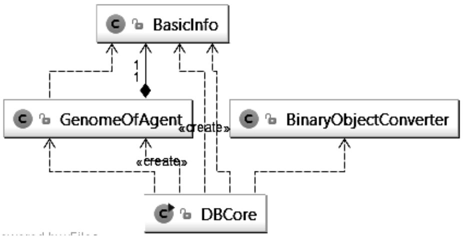


Fig. 4. The diagram of the classes required to work with the database

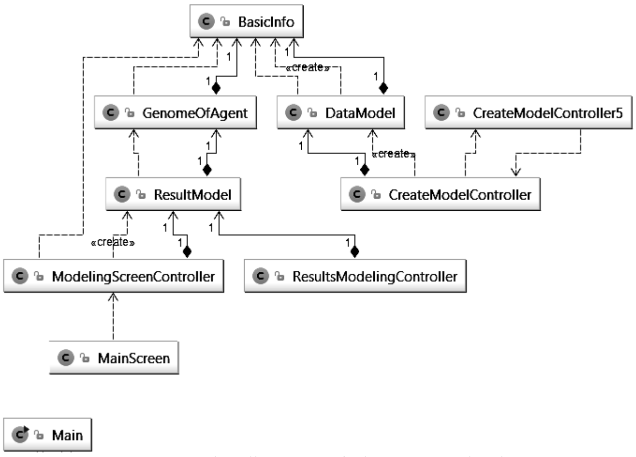
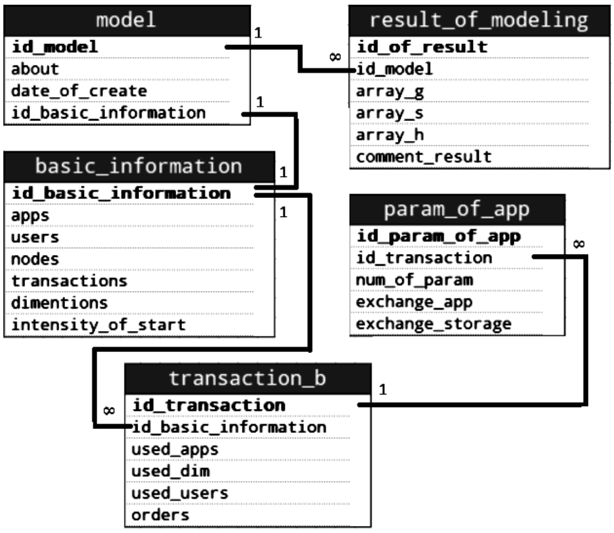


Fig. 5. The diagram of classes required for graphical user interface work.

The database schema is its structure. The scheme is described in a formal language that is supported by the engineering data-base management system. In relational databases, the schema defines tables. Fields in each table and integrity constraints. Schemes are generally stored in a data dictionary. Although the schema is defined in the database language as text. The term is often used to denote a graphical representation of the structure of a database. The main objects of the graphical representation of the schema are tables and links. They are defined by foreign keys. Having about the input and output information of the model is not difficult to draw a conclusion. About the fact that in order to save it to a database, you need a relational database whose schema is shown in Fig. 6.

5. Explanation of the choice of the software programming language

The program will have a large user interface. At the same time, the scheme of the genetic algorithm will have a high cost of resources. Only in the case of a large dimension of the task. Therefore, it is advisable to use the Java programming language. It will be able to ensure the portability of the program with minor losers compared to C / C ++. To create a graphical interface, we will use a platform based on Java - JavaFX. Tables do not have very complex interconnections. Small tables are small. So to use the built-in database. Do not overload an excessive functionality to preserve the benefits of relational databases. Namely SQLite database. Which will also allow you to use the SQL query language.


model
id_model
about
date_of_create
id_basic_information

result_of_modeling
id_of_result
id_model
array_g
array_s
array_h
comment_result

basic_information
id_basic_information
apps
users
nodes
transactions
dimentions
intensity_of_start

param_of_app
id_param_of_app
id_transaction
num_of_param
exchange_app
exchange_storage

transaction_b
id_transaction
id_basic_information
used_apps
used_dim
used_users
orders

Fig. 6. Database connection that stores input and output data

Conclusions

The developed algorithm for synthesizing the information structure model is presented. The basis of which lies the genetic algorithm. The algorithm with formulas for calculating fitness functions is presented below. He characterizes the fitness of the agent. The formulas for calculating the load on the nodes and network equipment of the system are developed. Also the information on the diagram of variants of use of computer programs is given. According to this information and knowledge of the tuple of the model, a diagram of the program usage options is given. That in the future will help in the development of the program. The chosen point of view of implementation takes into account the genetic algorithm. To create a visual interface, a class diagram has been designed. They are necessary for creating a program. The presented diagram of the developed database with several tables. It is able to store incoming and outgoing data**.**

СПИСОК ЛІТЕРАТУРИ

1. Шматков С. І. Модель інформаційної структури гіперконвергентної системи підтримки електронних обчислювальних ресурсів університетської e-learning / С. І. Шматков, Н. Г. Кучук, В. В. Донець// Системи управління, навігації та зв'язку : науковий журнал. – Полтава : ПНТУ, 2018. – Вип. 2 (48). – С. 97-100.
2. Semenov S. Development of graphic-analytical models for the software security testing algorithm / S. Semenov, O. Sira, N. Kuchuk // Eastern-European journal of enterprise technologies. – 2018. – № 2/4(92). – P. 39-46.
3. Kuchuk G. Approaches to selection of combinatorial algorithm for optimization in network traffic control of safety-critical systems / G. Kuchuk, V. Kharchenko, A. Kovalenko, E.Ruchkov // East-West Design & Test Symposium (EWDTS). – 2016. – P. 1-6. doi : https://doi.org/10.1109/EWDTS.2016.7807655.
4. Kuchuk G. A. An Approach To Development Of Complex Metric For Multiservice Network Security Assessment / G. A. Kuchuk, A. A. Kovalenko, A. A. Mozhaev // Statistical Methods Of Signal and Data Processing (SMSDP – 2010): Proc. Int. Conf., October 13-14, 2010.– Kyiv: NAU, RED, IEEE Ukraine section joint SP, 2010. – P. 158-160.

5. Кучук Г. А. Метод синтезу інформаційної структури зв'язного фрагменту корпоративної мультисервісної мережі / Г. А. Кучук // Збірник наукових праць Харківського університету Повітряних сил. – 2013. – Вип. 2(35). – С. 97-102.
6. Диаграмма классов [Електронний ресурс]. – Режим доступу: https://ru.wikipedia.org/wiki/Диаграмма_классов.
7. Mozhaev O. Multiservise network security metric / O.Mozhaev, H. Kuchuk, N. Kuchuk, M. Mozhaev, M. Lohvynenco // IEEE Advanced information and communication technologies-2017. Proc. of the 2th Int. Conf. – Lviv, 2017. – P. 133-136.

REFERENCES

1. Shmatkov, S.I., Kuchuk, N.G. and Donets VV (2018), "Model of information structure of the hyperconvergent system of support of electronic computing resources of university e-learning", *Control systems, navigation and communication*, Poltava, No. 2 (48), pp. 97-100.
2. Semenov, S., Sira, O. and Kuchuk N. (2018), "Development of graphic-analytical models for the software security testing algorithm", *Eastern-European journal of enterprise technologies*, No. 2/4(92), pp. 39-46.
3. Kuchuk, G., Kharchenko, V., Kovalenko, A. and Ruchkov, E. (2016), "Approaches to selection of combinatorial algorithm for optimization in network traffic control of safety-critical systems", *East-West Design & Test Symposium (EWDTS)*, pp. 1-6, available at : https://doi.org/10.1109/EWDTS.2016.7807655.
4. Kuchuk, G.A., Kovalenko, A.A. and Mozhaev A.A. (2010), "An Approach To Development Of Complex Metric For Multiservice Network Security Assessment", *Statistical Methods Of Signal and Data Processing (SMSDP – 2010)*: Proc. Int. Conf., October 13-14, 2010, NAU, RED, IEEE Ukraine section joint SP, Kyiv, pp. 158-160.
5. Kuchuk, G.A. (2013), "Method of synthesis of the information structure of the coherent fragment of the corporate multiservice network", *Collection of scientific works of the Kharkiv University of Air Forces*, No. 2 (35), pp. 97-102.
6. *Class diagram* (2018), available at : https://ru.wikipedia.org/wiki/Диаграмма_классов (last accessed March 21, 2018).
7. Mozhaev, O., Kuchuk H., Kuchuk, N., Mozhaev, M. and Lohvynenko M. (2017), "Multiservise network security metric", *IEEE Advanced information and communication technologies-2017.* Proc. of the 2th Int. Conf. – Lviv, 2017. – pp. 133-136.

Received (Надійшла) 16.04.2018
Accepted for publication (Прийнята до друку) 30.05.2018

Розробка програмного забезпечення процесу моделювання синтезу інформаційної системи e-learning

В. В. Донець, Н. Г. Кучук, С. І. Шматков

В статті відображено результати створеної моделі інформаційної структури системи підтримки електронних освітніх ресурсів e-learning. Відповідно до цієї моделі, була створена модель синтезу інформаційної системи e-learning. Результатом моделювання є програма із графічним інтерфейсом користувача моделі синтезу інформаційної системи e-learning, що приймає на вхід параметри гіперконвергентної базової мережі та системи електронного навчання, а результатом синтезу є оптимальне розташування користувачів, додатків та блоків даних по вузлам базової мережі з урахуванням завантаженості транзакцій системи, що оптимізує завантаження вузлів. Цим самим покращуючи ефективність функціонування системи e-learning. Структура гіперконвергентної базової мережі підтримки e-learning розглядається як основний фактор, що впливає на якість запитів системи. Тому важливо проаналізувати структуру при виборі варіантів побудови гіперконвергентної базової мережі та її управління. Основна мета аналізу структури – визначення параметрів потоків даних в каналах зв'язку мережі. Отримані результати необхідні для адекватної оцінки навантаження каналів та вузлів мережі. Але потоки даних формуються завданнями e-learning, які використовують застосунки, що запускаються на вузлах мережі та генерують ще трафік між собою. Також наведена інформацію про діаграми варіантів використання комп'ютерних програм. Відповідно до цієї інформації та знань про кортеж моделі наведена діаграма варіантів використання програми, що надалі допоможе в розробці програми. **Висновок.** Обрана точка зору реалізації та ураховуючи генетичний алгоритм, необхідність в збереженні результатів та створення візуального інтерфейсу спроектована діаграма класів необхідних для створення програми. Представлена схема розробленої бази даних із декількома таблицями, здатна зберігати вхідні та вихідні дані, та не перевантажена зв'язками.

Ключові слова: e-learning; моделювання; гіперконвергентна мережа; генетичний алгоритм; програмне забезпечення.

Разработка программного обеспечения процесса моделирования синтеза информационной системы e-learning

В. В. Донец, Н. Г. Кучук, С. И. Шматков

В статье отражены результаты исследования созданной модели информационной структуры системы поддержки электронных образовательных ресурсов e-learning. Данная модель позволила осуществить процесс моделирования синтеза информационной системы e-learning. Для моделирования использовалась программа с графическим интерфейсом пользователя модели синтеза информационной системы e-learning. Программа в качестве входных данных использует параметры гиперконвергентной базовой сети и системы электронного обучения. Результатом синтеза является оптимальное размещение пользователей, приложений и блоков данных по узлам базовой сети. При этом учитывается загруженность транзакций системы. Это оптимизирует загрузку узлов. В результате увеличивается эффективность функционирования системы e-learning. Структура гиперконвергентной базовой сети поддержки e-learning рассматривается как основной фактор, который влияет на качество запросов системы. Поэтому важно проанализировать структуру при выборе вариантов построения гиперконвергентнои базовой сети и ее управления. Основной целью анализа структуры является определение параметров потоков данных в каналах связи сети. Полученные результаты необходимы для адекватной оценки нагрузки каналов и узлов сети. Потоки данных формируют задачи e-learning. Задачи используют приложения, которые запускаются на узлах сети и генерируют сетевой трафик.

Ключевые слова: e-learning; моделирование; гиперконвергентная сеть; генетический алгоритм; программное обеспечение.