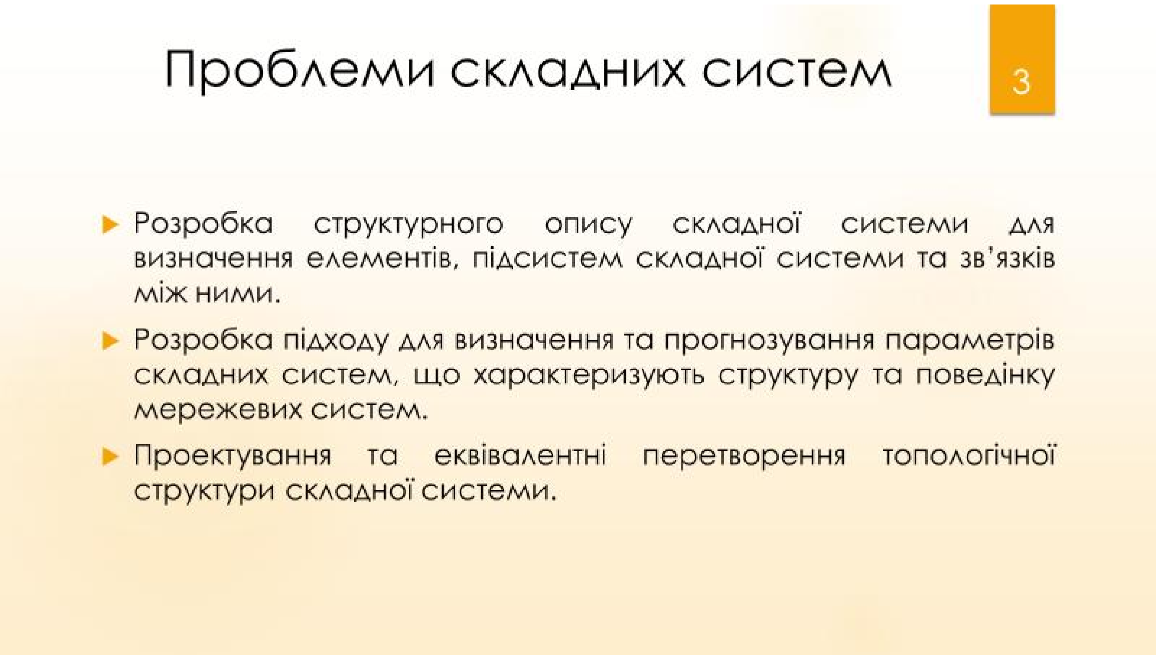
ДОДАТОК А

Графічний матеріал атестаційної роботи

Складні системи

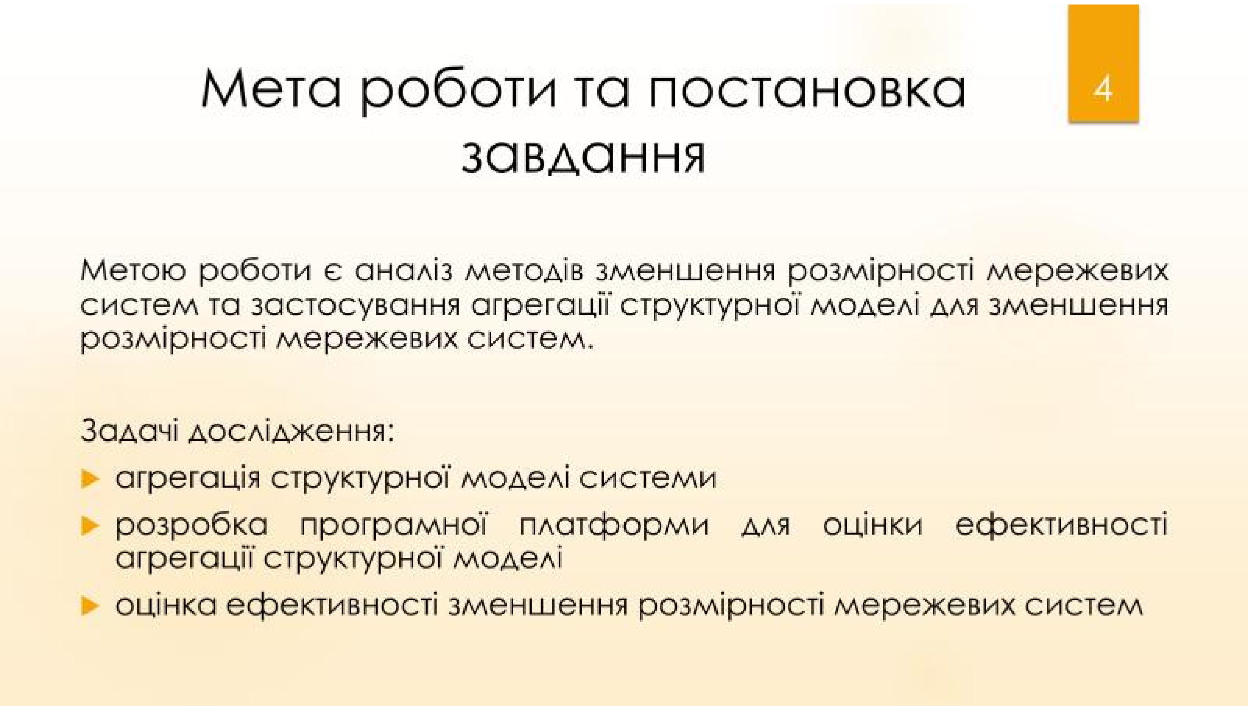
Рисунок А.1 – Слайд №2 презентації



Проблеми складних систем

- Розробка структурного опису складної системи для визначення елементів, підсистем складної системи та зв'язків між ними.
- Розробка підходу для визначення та прогнозування параметрів складних систем, що характеризують структуру та поведінку мережевих систем.
- Проектування та еквівалентні перетворення топологічної структури складної системи.

Рисунок А.2 – Слайд №3 презентації



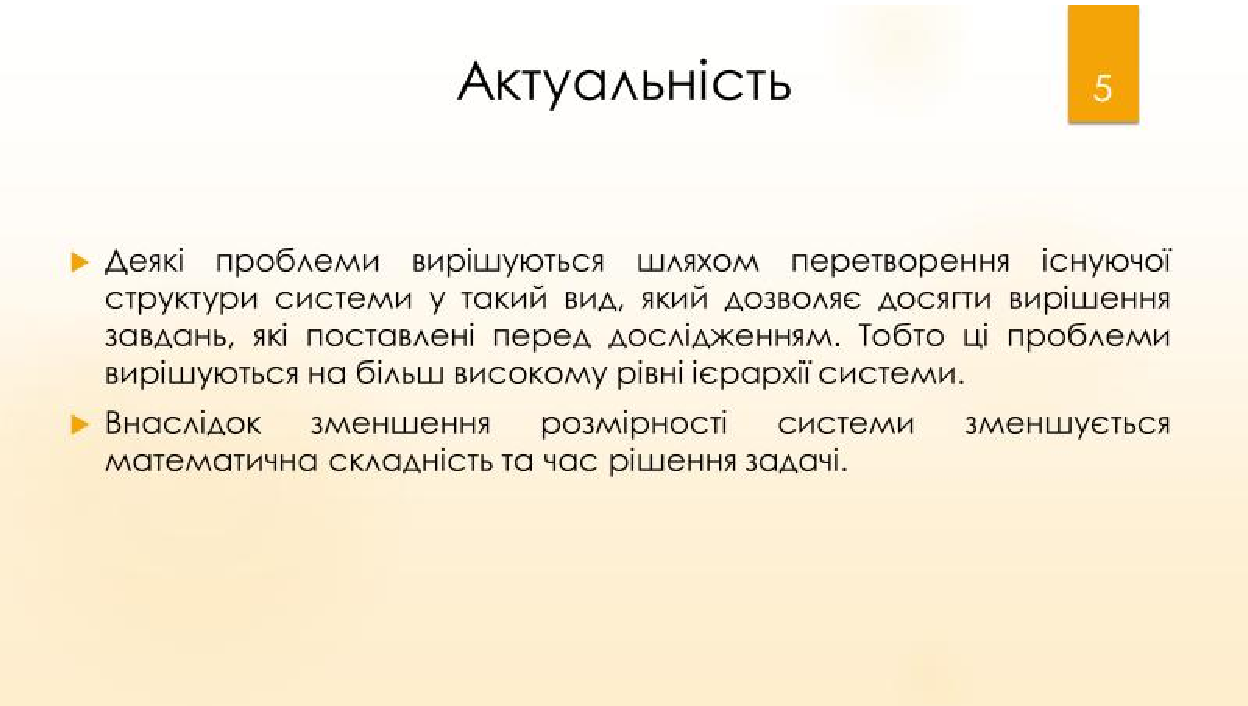
Мета роботи та постановка завдання

Метою роботи є аналіз методів зменшення розмірності мережевих систем та застосування агрегації структурної моделі для зменшення розмірності мережевих систем.

Задачі дослідження:

- агрегація структурної моделі системи
- розробка програмної платформи для оцінки ефективності агрегації структурної моделі
- оцінка ефективності зменшення розмірності мережевих систем

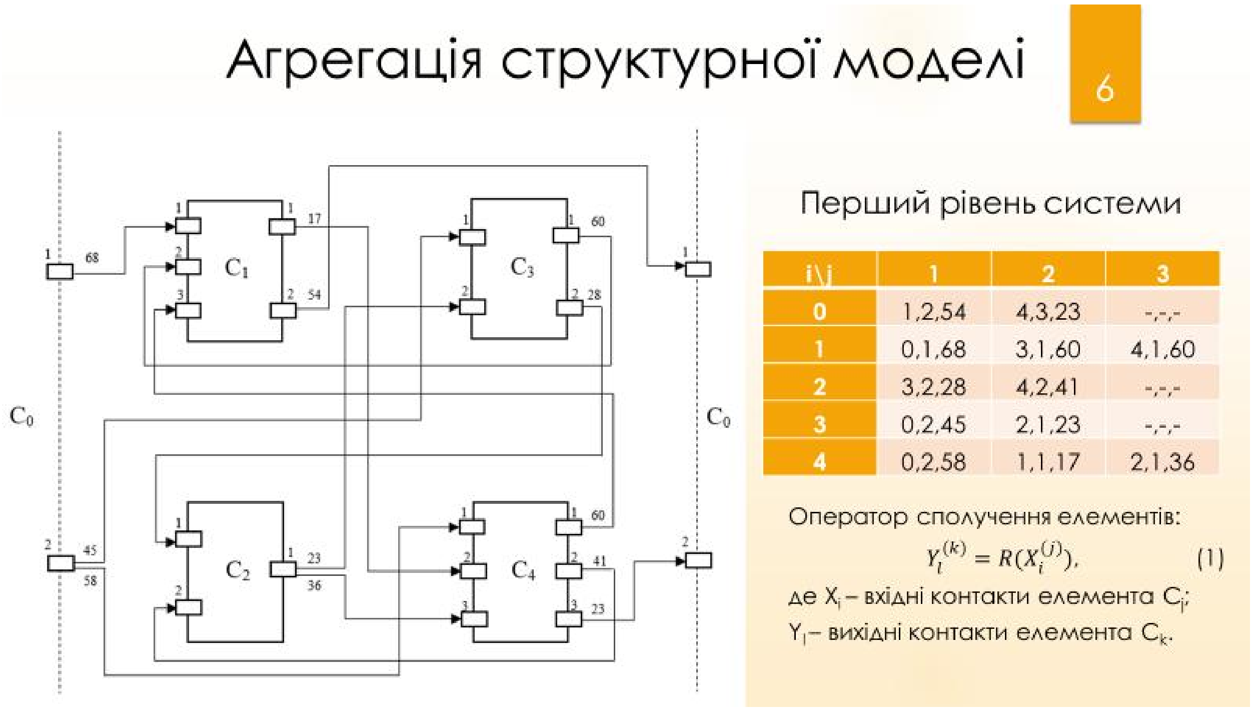
Рисунок А.3 – Слайд №4 презентації



Актуальність

- Деякі проблеми вирішуються шляхом перетворення існуючої структури системи у такий вид, який дозволяє досягти вирішення завдань, які поставлені перед дослідженням. Тобто ці проблеми вирішуються на більш високому рівні ієрархії системи.
- Внаслідок зменшення розмірності системи зменшується математична складність та час рішення задачі.

Рисунок А.4 – Слайд №5 презентації



Агрегація структурної моделі

Перший рівень системи

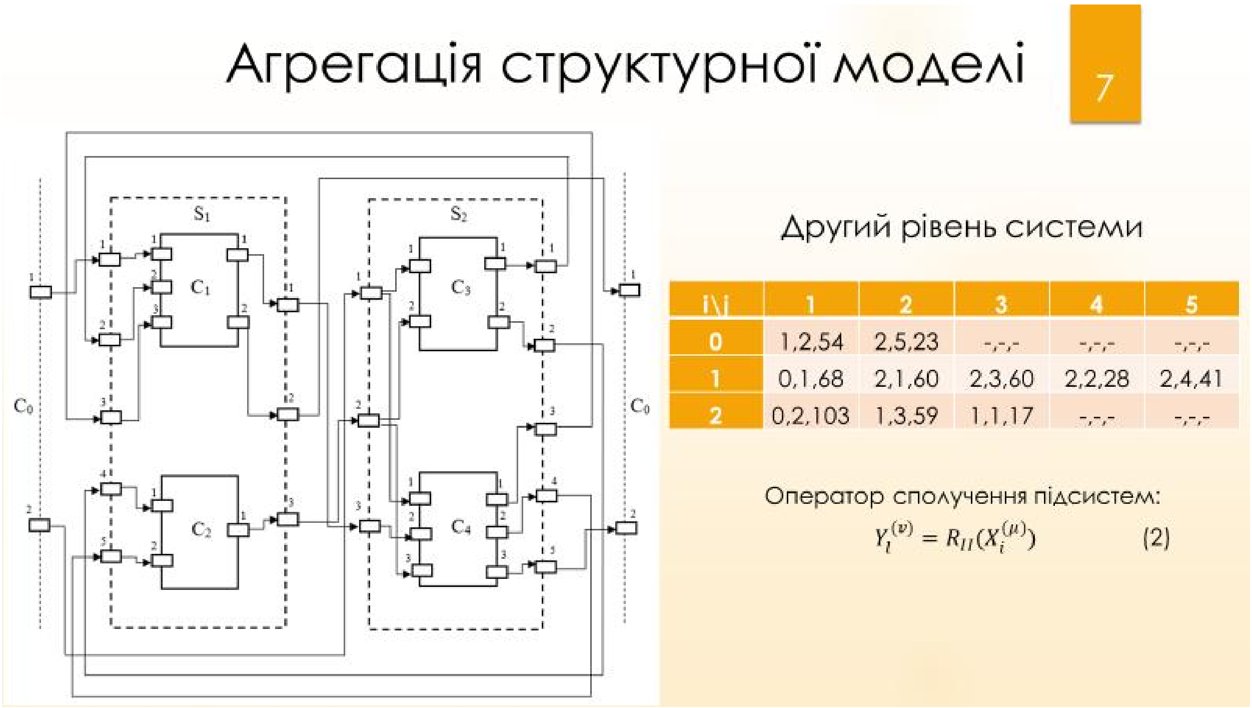
i\j	1	2	3
0	1,2,54	4,3,23	-,-,-
1	0,1,68	3,1,60	4,1,60
2	3,2,28	4,2,41	-,-,-
3	0,2,45	2,1,23	-,-,-
4	0,2,58	1,1,17	2,1,36

Оператор сполучення елементів:

$$Y_l^{(k)} = R(X_i^{(j)}), \qquad (1)$$

де X_i – вхідні контакти елемента C_j;
Y_l – вихідні контакти елемента C_k.

Рисунок А.5 – Слайд №6 презентації



Агрегація структурної моделі

Другий рівень системи

i\j	1	2	3	4	5
0	1,2,54	2,5,23	-,-,-	-,-,-	-,-,-
1	0,1,68	2,1,60	2,3,60	2,2,28	2,4,41
2	0,2,103	1,3,59	1,1,17	-,-,-	-,-,-

Оператор сполучення підсистем:

$$Y_l^{(v)} = R_{II}(X_i^{(\mu)}) \qquad (2)$$

Рисунок А.6– Слайд №7 презентації

Програмна платформа

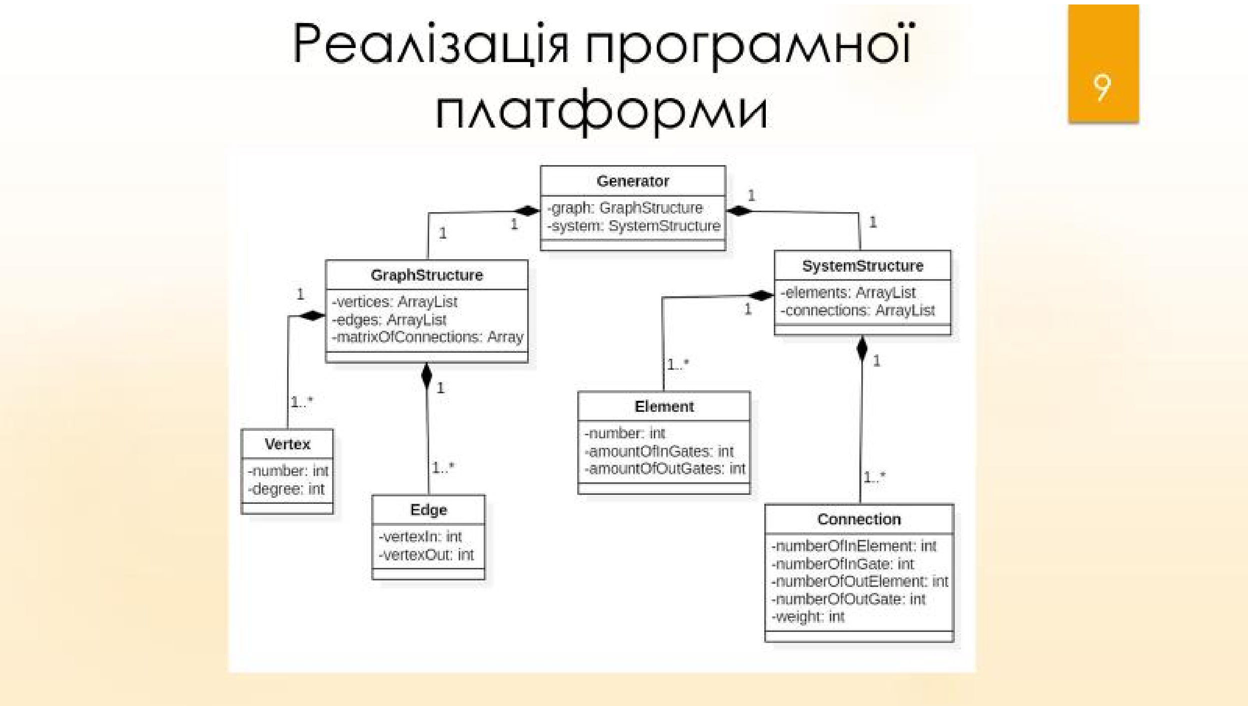
Модуль генератора мережевих структур

Модуль агрегації структурної моделі

Модуль аналізу характеристик реальних систем

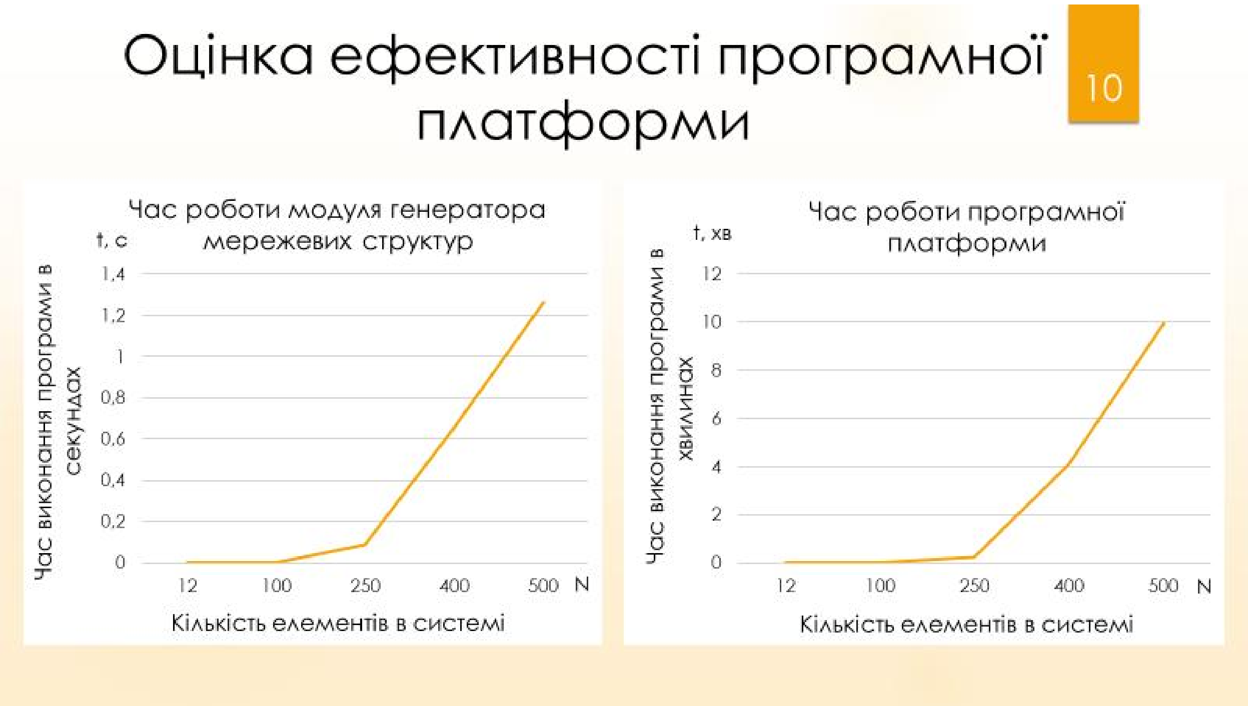
Модуль пошуку максимального потоку

Рисунок А.7 – Слайд №8 презентації



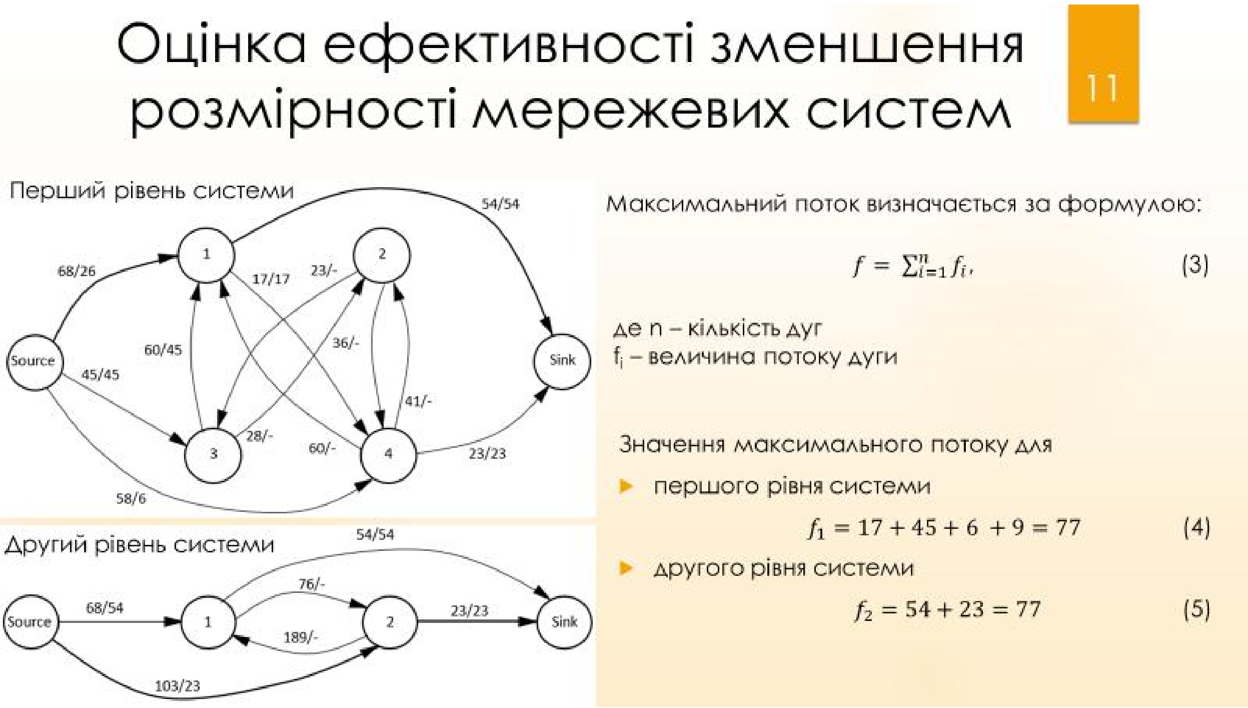
Реалізація програмної платформи

Рисунок А.8 – Слайд №9 презентації



Оцінка ефективності програмної платформи

Рисунок А.9 – Слайд №10 презентації



Оцінка ефективності зменшення розмірності мережевих систем

Максимальний поток визначається за формулою:

$$f = \sum_{i=1}^{n} f_i, \quad (3)$$

де n – кількість дуг
f_i – величина потоку дуги

Значення максимального потоку для

- першого рівня системи

$$f_1 = 17 + 45 + 6 + 9 = 77 \quad (4)$$

- другого рівня системи

$$f_2 = 54 + 23 = 77 \quad (5)$$

Рисунок А.10 – Слайд №11 презентації

Проблема втрати інформації

Втрата інформації може розглядатися як вартість агрегації J :

$$J = M(N) - M(\bar{N}), \qquad (6)$$

де M – це інформація, що міститься у структурі мережі.

В результаті пошуку максимального потоку для першого (4) та другого (5) рівня системи отримуємо вираз:

$$f_1 = f_2 \qquad (7)$$

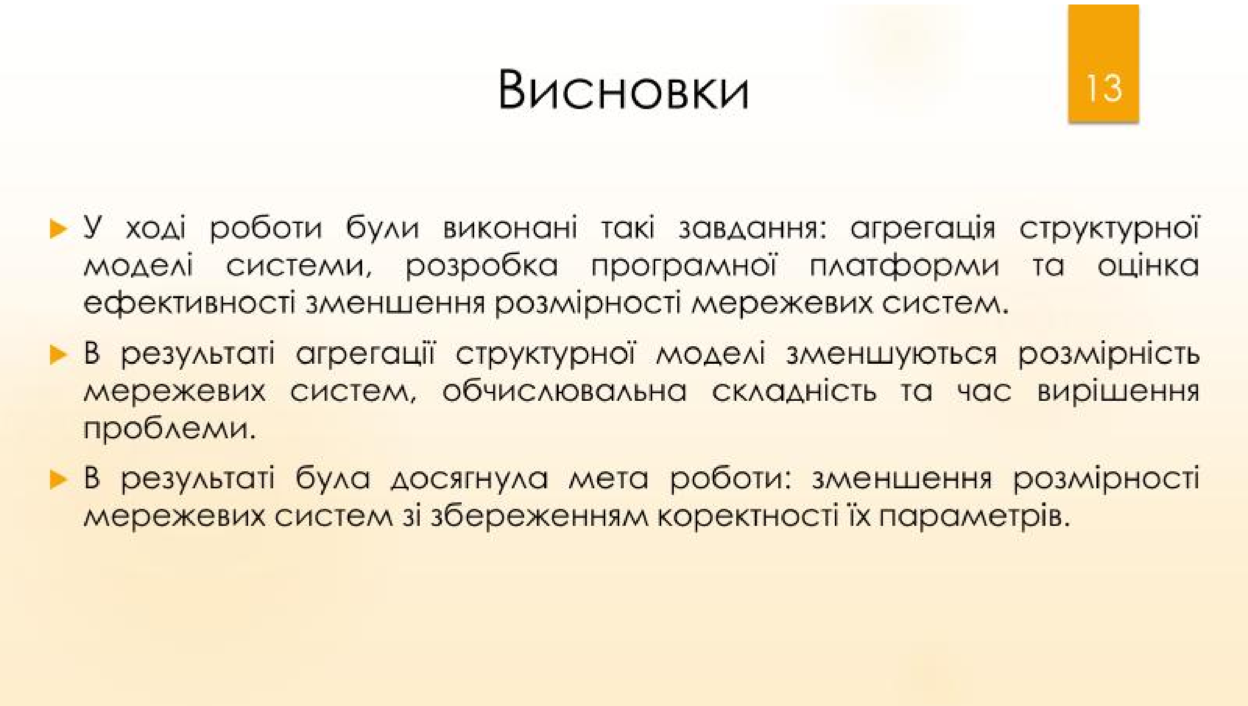
Приймемо значення максимального потоку f як інформацію M.

По формулі (6) знайдемо вартість агрегації J:

$$J = M(N) - M(\bar{N}) = 0 \qquad (8)$$

Тому у цьому випадку в результаті агрегації структурної моделі немає втрати інформації.

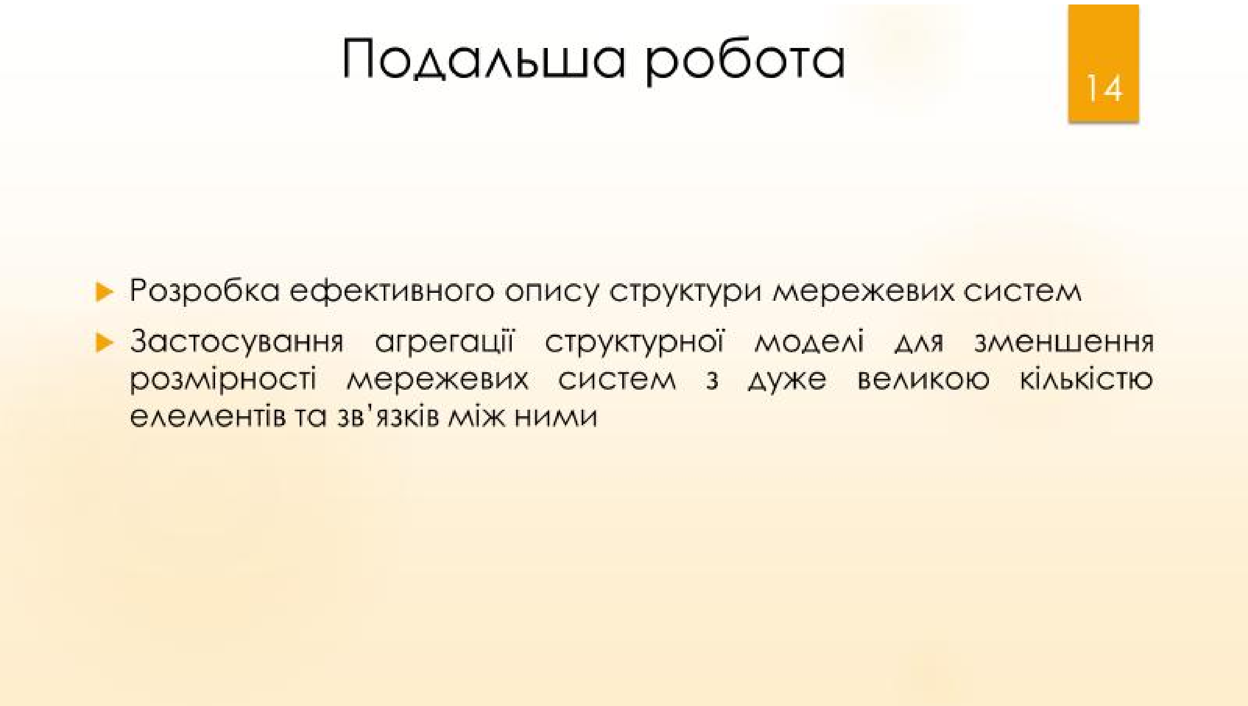
Рисунок А.11 – Слайд №12 презентації



Висновки

- У ході роботи були виконані такі завдання: агрегація структурної моделі системи, розробка програмної платформи та оцінка ефективності зменшення розмірності мережевих систем.
- В результаті агрегації структурної моделі зменшуються розмірність мережевих систем, обчислювальна складність та час вирішення проблеми.
- В результаті була досягнула мета роботи: зменшення розмірності мережевих систем зі збереженням коректності їх параметрів.

Рисунок А.12 – Слайд №13 презентації



Подальша робота

- Розробка ефективного опису структури мережевих систем
- Застосування агрегації структурної моделі для зменшення розмірності мережевих систем з дуже великою кількістю елементів та зв'язків між ними

Рисунок А.13 – Слайд №14 презентації

ДОДАТОК Б

Публікації по темі атестаційної роботи

IEEE

PROCEEDINGS of
2018 IEEE 9th International Conference on
Dependable Systems, Services and Technologies

DESSERT'2018

INTERNATIONAL SCIENTIFIC TECHNICAL CONFERENCE
IEEE DeSSerT 2018 UKRAINE KYIV MAY 24-27
DEPENDABLE SYSTEMS, SERVICES & TECHNOLOGIES

Ukraine, Kyiv
May 24-27, 2018

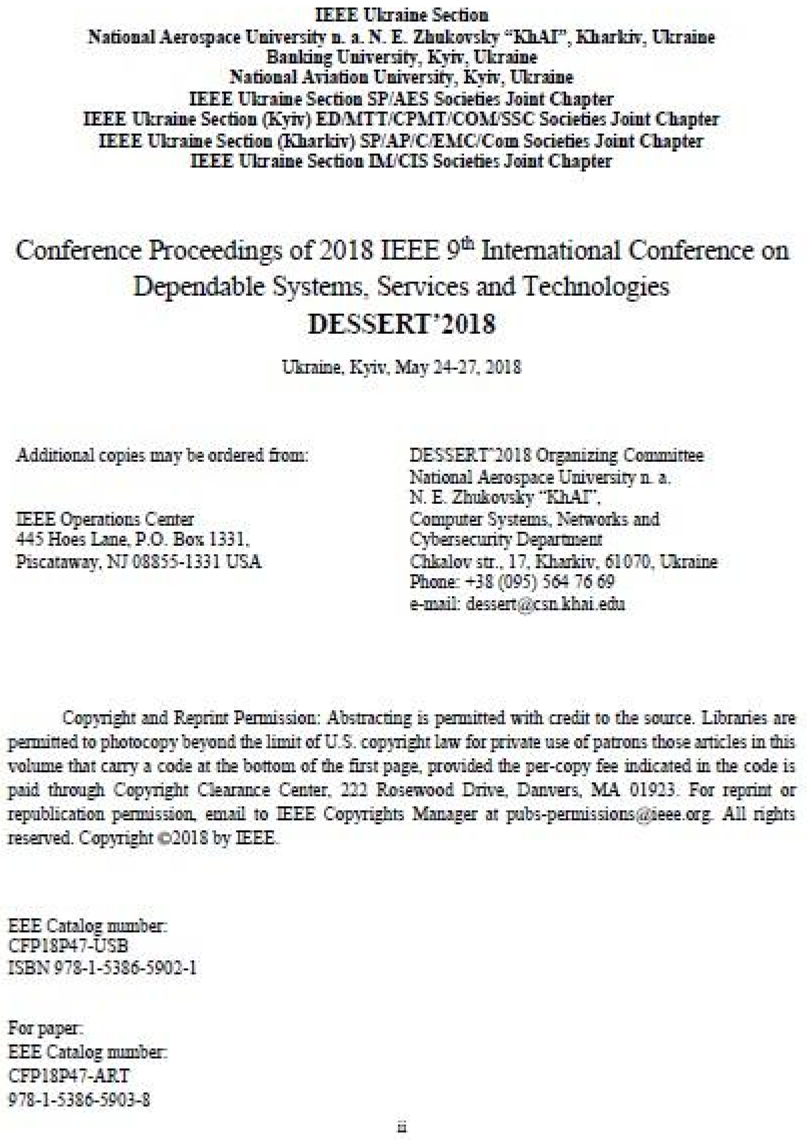
ORGANIZED BY

IEEE Ukraine Section
IEEE Ukraine Section SP/AES Societies Joint Chapter
IEEE Ukraine Section (Kyiv) ED/MTT/CPMT/COM/SSC Societies Joint Chapter
IEEE Ukraine Section (Kharkiv) SP/AP/C/EMC/Com Societies Joint Chapter
IEEE Ukraine Section IM/CIS Societies Joint Chapter
National Aerospace University n. a. N. E. Zhukovsky "KhAI", Kharkiv, Ukraine
Banking University, Kyiv, Ukraine
National Aviation University, Kyiv, Ukraine

EXCLUSIVE PARTNERS

Research and Production Corporation Radiy, Ukraine
National Bank of Ukraine

Рисунок Б.1 – Тези доповіді конференції « Formal transformations of structural models of complex network systems»



IEEE Ukraine Section
National Aerospace University n. a. N. E. Zhukovsky "KhAI", Kharkiv, Ukraine
Banking University, Kyiv, Ukraine
National Aviation University, Kyiv, Ukraine
IEEE Ukraine Section SP/AES Societies Joint Chapter
IEEE Ukraine Section (Kyiv) ED/MTT/CPMT/COM/SSC Societies Joint Chapter
IEEE Ukraine Section (Kharkiv) SP/AP/C/EMC/Com Societies Joint Chapter
IEEE Ukraine Section IM/CIS Societies Joint Chapter

Conference Proceedings of 2018 IEEE 9th International Conference on
Dependable Systems, Services and Technologies
DESSERT'2018

Ukraine, Kyiv, May 24-27, 2018

Additional copies may be ordered from:

IEEE Operations Center
445 Hoes Lane, P.O. Box 1331,
Piscataway, NJ 08855-1331 USA

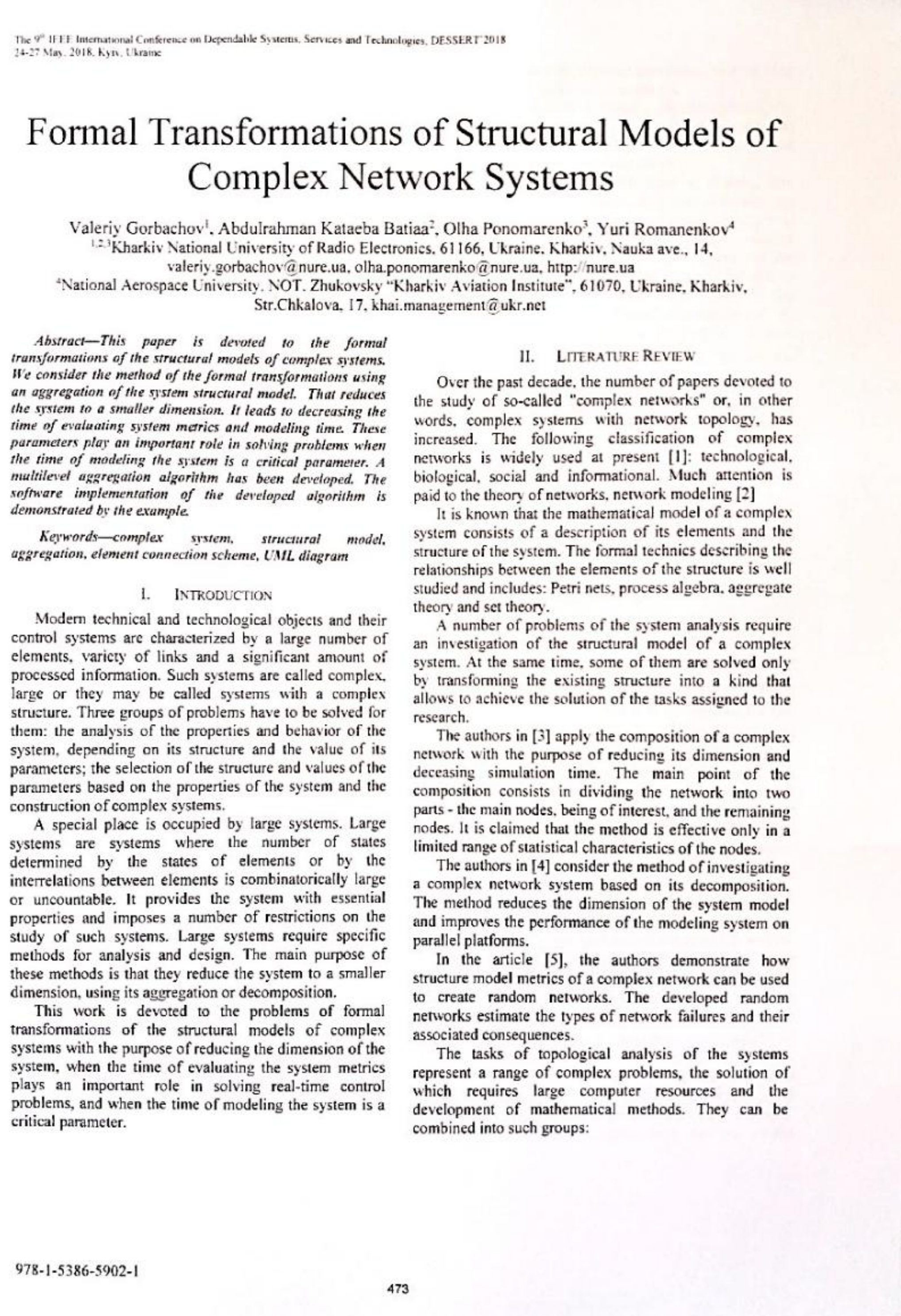
DESSERT'2018 Organizing Committee
National Aerospace University n. a.
N. E. Zhukovsky "KhAI",
Computer Systems, Networks and
Cybersecurity Department
Chkalov str., 17, Kharkiv, 61070, Ukraine
Phone: +38 (095) 564 76 69
e-mail: dessert@csn.khai.edu

EEE Catalog number:
CFP18P47-USB
ISBN 978-1-5386-5902-1

For paper:
EEE Catalog number:
CFP18P47-ART
978-1-5386-5903-8

ii

Рисунок Б.2 – Тези доповіді конференції « Formal transformations of structural models of complex network systems»



The 9th IEEE International Conference on Dependable Systems, Services and Technologies, DESSERT'2018
24-27 May, 2018, Kyiv, Ukraine

Formal Transformations of Structural Models of Complex Network Systems

Valeriy Gorbachov[1], Abdulrahman Kataeba Batiaa[2], Olha Ponomarenko[3], Yuri Romanenkov[4]
[1,2,3]Kharkiv National University of Radio Electronics, 61166, Ukraine, Kharkiv, Nauka ave., 14,
valeriy.gorbachov@nure.ua, olha.ponomarenko@nure.ua, http://nure.ua
[4]National Aerospace University, NOT. Zhukovsky "Kharkiv Aviation Institute", 61070, Ukraine, Kharkiv,
Str.Chkalova, 17, khai.management@ukr.net

Abstract*—This paper is devoted to the formal transformations of the structural models of complex systems. We consider the method of the formal transformations using an aggregation of the system structural model. That reduces the system to a smaller dimension. It leads to decreasing the time of evaluating system metrics and modeling time. These parameters play an important role in solving problems when the time of modeling the system is a critical parameter. A multilevel aggregation algorithm has been developed. The software implementation of the developed algorithm is demonstrated by the example.

***Keywords*—complex system, structural model, aggregation, element connection scheme, UML diagram**

I. INTRODUCTION

Modern technical and technological objects and their control systems are characterized by a large number of elements, variety of links and a significant amount of processed information. Such systems are called complex, large or they may be called systems with a complex structure. Three groups of problems have to be solved for them: the analysis of the properties and behavior of the system, depending on its structure and the value of its parameters; the selection of the structure and values of the parameters based on the properties of the system and the construction of complex systems.

A special place is occupied by large systems. Large systems are systems where the number of states determined by the states of elements or by the interrelations between elements is combinatorically large or uncountable. It provides the system with essential properties and imposes a number of restrictions on the study of such systems. Large systems require specific methods for analysis and design. The main purpose of these methods is that they reduce the system to a smaller dimension, using its aggregation or decomposition.

This work is devoted to the problems of formal transformations of the structural models of complex systems with the purpose of reducing the dimension of the system, when the time of evaluating the system metrics plays an important role in solving real-time control problems, and when the time of modeling the system is a critical parameter.

II. LITERATURE REVIEW

Over the past decade, the number of papers devoted to the study of so-called "complex networks" or, in other words, complex systems with network topology, has increased. The following classification of complex networks is widely used at present [1]: technological, biological, social and informational. Much attention is paid to the theory of networks, network modeling [2]

It is known that the mathematical model of a complex system consists of a description of its elements and the structure of the system. The formal technics describing the relationships between the elements of the structure is well studied and includes: Petri nets, process algebra, aggregate theory and set theory.

A number of problems of the system analysis require an investigation of the structural model of a complex system. At the same time, some of them are solved only by transforming the existing structure into a kind that allows to achieve the solution of the tasks assigned to the research.

The authors in [3] apply the composition of a complex network with the purpose of reducing its dimension and deceasing simulation time. The main point of the composition consists in dividing the network into two parts - the main nodes, being of interest, and the remaining nodes. It is claimed that the method is effective only in a limited range of statistical characteristics of the nodes.

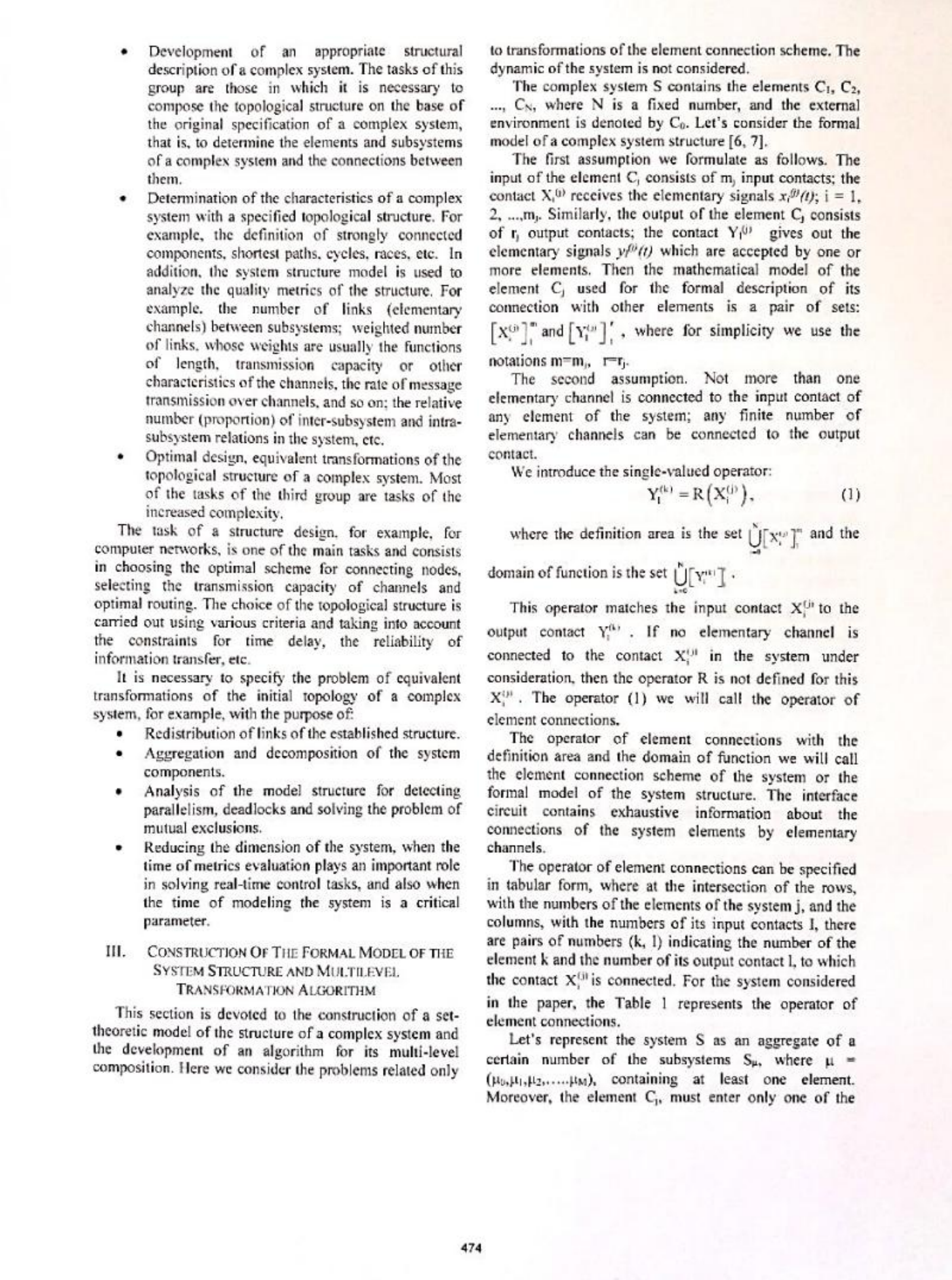
The authors in [4] consider the method of investigating a complex network system based on its decomposition. The method reduces the dimension of the system model and improves the performance of the modeling system on parallel platforms.

In the article [5], the authors demonstrate how structure model metrics of a complex network can be used to create random networks. The developed random networks estimate the types of network failures and their associated consequences.

The tasks of topological analysis of the systems represent a range of complex problems, the solution of which requires large computer resources and the development of mathematical methods. They can be combined into such groups:

978-1-5386-5902-1

Рисунок Б.3 – Тези доповіді конференції « Formal transformations of structural models of complex network systems»



- Development of an appropriate structural description of a complex system. The tasks of this group are those in which it is necessary to compose the topological structure on the base of the original specification of a complex system, that is, to determine the elements and subsystems of a complex system and the connections between them.
- Determination of the characteristics of a complex system with a specified topological structure. For example, the definition of strongly connected components, shortest paths, cycles, races, etc. In addition, the system structure model is used to analyze the quality metrics of the structure. For example, the number of links (elementary channels) between subsystems; weighted number of links, whose weights are usually the functions of length, transmission capacity or other characteristics of the channels, the rate of message transmission over channels, and so on; the relative number (proportion) of inter-subsystem and intra-subsystem relations in the system, etc.
- Optimal design, equivalent transformations of the topological structure of a complex system. Most of the tasks of the third group are tasks of the increased complexity.

The task of a structure design, for example, for computer networks, is one of the main tasks and consists in choosing the optimal scheme for connecting nodes, selecting the transmission capacity of channels and optimal routing. The choice of the topological structure is carried out using various criteria and taking into account the constraints for time delay, the reliability of information transfer, etc.

It is necessary to specify the problem of equivalent transformations of the initial topology of a complex system, for example, with the purpose of:

- Redistribution of links of the established structure.
- Aggregation and decomposition of the system components.
- Analysis of the model structure for detecting parallelism, deadlocks and solving the problem of mutual exclusions.
- Reducing the dimension of the system, when the time of metrics evaluation plays an important role in solving real-time control tasks, and also when the time of modeling the system is a critical parameter.

III. CONSTRUCTION OF THE FORMAL MODEL OF THE SYSTEM STRUCTURE AND MULTILEVEL TRANSFORMATION ALGORITHM

This section is devoted to the construction of a set-theoretic model of the structure of a complex system and the development of an algorithm for its multi-level composition. Here we consider the problems related only to transformations of the element connection scheme. The dynamic of the system is not considered.

The complex system S contains the elements C_1, C_2, ..., C_N, where N is a fixed number, and the external environment is denoted by C_0. Let's consider the formal model of a complex system structure [6, 7].

The first assumption we formulate as follows. The input of the element C_j consists of m_j input contacts; the contact $X_i^{(j)}$ receives the elementary signals $x_i^{(j)}(t)$; i = 1, 2, ...,m_j. Similarly, the output of the element C_j consists of r_j output contacts; the contact $Y_l^{(j)}$ gives out the elementary signals $y_l^{(j)}(t)$ which are accepted by one or more elements. Then the mathematical model of the element C_j used for the formal description of its connection with other elements is a pair of sets: $\left[X_i^{(j)}\right]_1^m$ and $\left[Y_l^{(j)}\right]_1^r$, where for simplicity we use the notations $m=m_j$, $r=r_j$.

The second assumption. Not more than one elementary channel is connected to the input contact of any element of the system; any finite number of elementary channels can be connected to the output contact.

We introduce the single-valued operator:

$$Y_l^{(k)} = R\left(X_i^{(j)}\right), \qquad (1)$$

where the definition area is the set $\bigcup_{j=1}^{N}\left[X_i^{(j)}\right]_1^m$ and the domain of function is the set $\bigcup_{k=0}^{N}\left[Y_l^{(k)}\right]$.

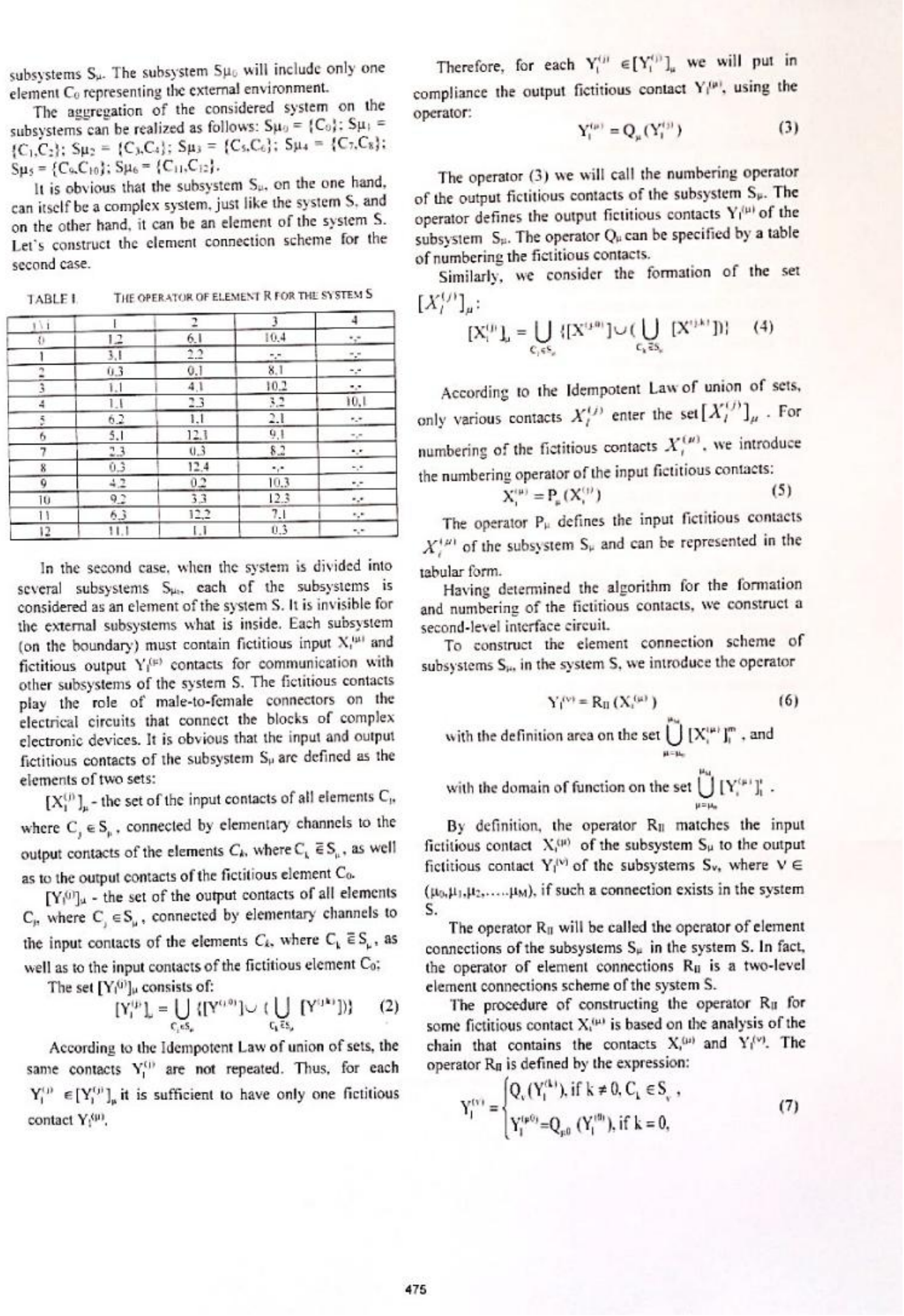
This operator matches the input contact $X_i^{(j)}$ to the output contact $Y_l^{(k)}$. If no elementary channel is connected to the contact $X_i^{(j)}$ in the system under consideration, then the operator R is not defined for this $X_i^{(j)}$. The operator (1) we will call the operator of element connections.

The operator of element connections with the definition area and the domain of function we will call the element connection scheme of the system or the formal model of the system structure. The interface circuit contains exhaustive information about the connections of the system elements by elementary channels.

The operator of element connections can be specified in tabular form, where at the intersection of the rows, with the numbers of the elements of the system j, and the columns, with the numbers of its input contacts I, there are pairs of numbers (k, l) indicating the number of the element k and the number of its output contact l, to which the contact $X_i^{(j)}$ is connected. For the system considered in the paper, the Table 1 represents the operator of element connections.

Let's represent the system S as an aggregate of a certain number of the subsystems S_μ, where $\mu = (\mu_0, \mu_1, \mu_2, \ldots, \mu_M)$, containing at least one element. Moreover, the element C_j, must enter only one of the

Рисунок Б.4 – Тези доповіді конференції « Formal transformations of structural models of complex network systems»



subsystems S_μ. The subsystem $S\mu_0$ will include only one element C_0 representing the external environment.

The aggregation of the considered system on the subsystems can be realized as follows: $S\mu_0 = \{C_0\}$; $S\mu_1 = \{C_1,C_2\}$; $S\mu_2 = \{C_3,C_4\}$; $S\mu_3 = \{C_5,C_6\}$; $S\mu_4 = \{C_7,C_8\}$; $S\mu_5 = \{C_9,C_{10}\}$; $S\mu_6 = \{C_{11},C_{12}\}$.

It is obvious that the subsystem S_μ, on the one hand, can itself be a complex system, just like the system S, and on the other hand, it can be an element of the system S. Let's construct the element connection scheme for the second case.

TABLE I. THE OPERATOR OF ELEMENT R FOR THE SYSTEM S

j\i	1	2	3	4
0	1.2	6.1	10.4	-.-
1	3.1	2.2	-.-	-.-
2	0.3	0.1	8.1	-.-
3	1.1	4.1	10.2	-.-
4	1.1	2.3	3.2	10.1
5	6.2	1.1	2.1	-.-
6	5.1	12.1	9.1	-.-
7	2.3	0.3	8.2	-.-
8	0.3	12.4	-.-	-.-
9	4.2	0.2	10.3	-.-
10	9.2	3.3	12.3	-.-
11	6.3	12.2	7.1	-.-
12	11.1	1.1	0.3	-.-

In the second case, when the system is divided into several subsystems $S_{\mu i}$, each of the subsystems is considered as an element of the system S. It is invisible for the external subsystems what is inside. Each subsystem (on the boundary) must contain fictitious input $X_l^{(\mu)}$ and fictitious output $Y_l^{(\mu)}$ contacts for communication with other subsystems of the system S. The fictitious contacts play the role of male-to-female connectors on the electrical circuits that connect the blocks of complex electronic devices. It is obvious that the input and output fictitious contacts of the subsystem S_μ are defined as the elements of two sets:

$[X_l^{(j)}]_\mu$ - the set of the input contacts of all elements C_j, where $C_j \in S_\mu$, connected by elementary channels to the output contacts of the elements C_k, where $C_k \overline{\in} S_\mu$, as well as to the output contacts of the fictitious element C_0.

$[Y_l^{(j)}]_\mu$ - the set of the output contacts of all elements C_j, where $C_j \in S_\mu$, connected by elementary channels to the input contacts of the elements C_k, where $C_k \overline{\in} S_\mu$, as well as to the input contacts of the fictitious element C_0;

The set $[Y_l^{(j)}]_\mu$ consists of:

$$[Y_l^{(j)}]_\mu = \bigcup_{C_j \in S_\mu} \{[Y^{(j,0)}] \cup (\bigcup_{C_k \overline{\in} S_\mu} [Y^{(j,k)}])\} \qquad (2)$$

According to the Idempotent Law of union of sets, the same contacts $Y_l^{(j)}$ are not repeated. Thus, for each $Y_l^{(j)} \in [Y_l^{(j)}]_\mu$ it is sufficient to have only one fictitious contact $Y_l^{(\mu)}$.

Therefore, for each $Y_l^{(j)} \in [Y_l^{(j)}]_\mu$ we will put in compliance the output fictitious contact $Y_l^{(\mu)}$, using the operator:

$$Y_l^{(\mu)} = Q_\mu(Y_l^{(j)}) \qquad (3)$$

The operator (3) we will call the numbering operator of the output fictitious contacts of the subsystem S_μ. The operator defines the output fictitious contacts $Y_l^{(\mu)}$ of the subsystem S_μ. The operator Q_μ can be specified by a table of numbering the fictitious contacts.

Similarly, we consider the formation of the set $[X_l^{(j)}]_\mu$:

$$[X_l^{(j)}]_\mu = \bigcup_{C_j \in S_\mu} \{[X^{(j,0)}] \cup (\bigcup_{C_k \overline{\in} S_\mu} [X^{(j,k)}])\} \qquad (4)$$

According to the Idempotent Law of union of sets, only various contacts $X_l^{(j)}$ enter the set $[X_l^{(j)}]_\mu$. For numbering of the fictitious contacts $X_l^{(\mu)}$, we introduce the numbering operator of the input fictitious contacts:

$$X_l^{(\mu)} = P_\mu(X_l^{(j)}) \qquad (5)$$

The operator P_μ defines the input fictitious contacts $X_l^{(\mu)}$ of the subsystem S_μ and can be represented in the tabular form.

Having determined the algorithm for the formation and numbering of the fictitious contacts, we construct a second-level interface circuit.

To construct the element connection scheme of subsystems $S_{\mu i}$ in the system S, we introduce the operator

$$Y_l^{(\nu)} = R_{II}(X_l^{(\mu)}) \qquad (6)$$

with the definition area on the set $\bigcup_{\mu=\mu_0}^{\mu_M} [X_l^{(\mu)}]_l^m$, and

with the domain of function on the set $\bigcup_{\mu=\mu_0}^{\mu_M} [Y_l^{(\mu)}]_l^t$.

By definition, the operator R_{II} matches the input fictitious contact $X_l^{(\mu)}$ of the subsystem S_μ to the output fictitious contact $Y_l^{(\nu)}$ of the subsystems S_ν, where $\nu \in (\mu_0,\mu_1,\mu_2,\ldots,\mu_M)$, if such a connection exists in the system S.

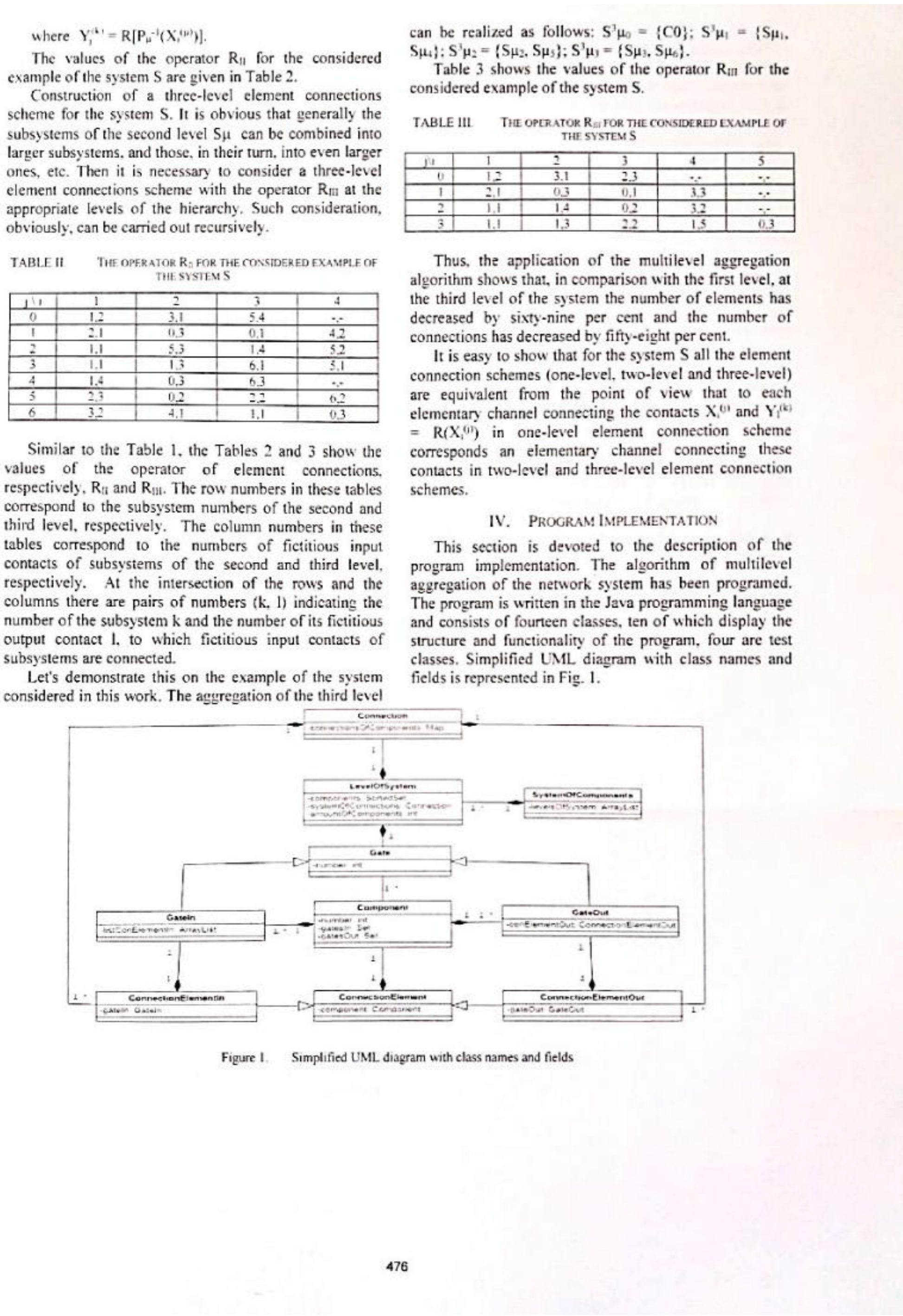
The operator R_{II} will be called the operator of element connections of the subsystems S_μ in the system S. In fact, the operator of element connections R_{II} is a two-level element connections scheme of the system S.

The procedure of constructing the operator R_{II} for some fictitious contact $X_l^{(\mu)}$ is based on the analysis of the chain that contains the contacts $X_l^{(\mu)}$ and $Y_l^{(\nu)}$. The operator R_{II} is defined by the expression:

$$Y_l^{(\nu)} = \begin{cases} Q_\nu(Y_l^{(k)}), \text{ if } k \neq 0, C_k \in S_\nu, \\ Y_l^{(\mu 0)} = Q_{\mu 0}(Y_l^{(0)}), \text{ if } k = 0, \end{cases} \qquad (7)$$

475

Рисунок Б.5 – Тези доповіді конференції « Formal transformations of structural models of complex network systems»



where $Y_j^{(k)} = R[P_\mu^{-1}(X_i^{(\mu)})]$.

The values of the operator R_{II} for the considered example of the system S are given in Table 2.

Construction of a three-level element connections scheme for the system S. It is obvious that generally the subsystems of the second level $S\mu$ can be combined into larger subsystems, and those, in their turn, into even larger ones, etc. Then it is necessary to consider a three-level element connections scheme with the operator R_{III} at the appropriate levels of the hierarchy. Such consideration, obviously, can be carried out recursively.

TABLE II THE OPERATOR R_{II} FOR THE CONSIDERED EXAMPLE OF THE SYSTEM S

j \ i	1	2	3	4
0	1,2	3,1	5,4	-,-
1	2,1	0,3	0,1	4,2
2	1,1	5,3	1,4	5,2
3	1,1	1,3	6,1	5,1
4	1,4	0,3	6,3	-,-
5	2,3	0,2	2,2	6,2
6	3,2	4,1	1,1	0,3

Similar to the Table 1, the Tables 2 and 3 show the values of the operator of element connections, respectively, R_{II} and R_{III}. The row numbers in these tables correspond to the subsystem numbers of the second and third level, respectively. The column numbers in these tables correspond to the numbers of fictitious input contacts of subsystems of the second and third level, respectively. At the intersection of the rows and the columns there are pairs of numbers (k, l) indicating the number of the subsystem k and the number of its fictitious output contact l, to which fictitious input contacts of subsystems are connected.

Let's demonstrate this on the example of the system considered in this work. The aggregation of the third level can be realized as follows: $S^3\mu_0 = \{C0\}$; $S^3\mu_1 = \{S\mu_1, S\mu_4\}$; $S^3\mu_2 = \{S\mu_2, S\mu_5\}$; $S^3\mu_3 = \{S\mu_3, S\mu_6\}$.

Table 3 shows the values of the operator R_{III} for the considered example of the system S.

TABLE III THE OPERATOR R_{III} FOR THE CONSIDERED EXAMPLE OF THE SYSTEM S

j \ i	1	2	3	4	5
0	1,2	3,1	2,3	-,-	-,-
1	2,1	0,3	0,1	3,3	-,-
2	1,1	1,4	0,2	3,2	-,-
3	1,1	1,3	2,2	1,5	0,3

Thus, the application of the multilevel aggregation algorithm shows that, in comparison with the first level, at the third level of the system the number of elements has decreased by sixty-nine per cent and the number of connections has decreased by fifty-eight per cent.

It is easy to show that for the system S all the element connection schemes (one-level, two-level and three-level) are equivalent from the point of view that to each elementary channel connecting the contacts $X_i^{(i)}$ and $Y_j^{(k)} = R(X_i^{(i)})$ in one-level element connection scheme corresponds an elementary channel connecting these contacts in two-level and three-level element connection schemes.

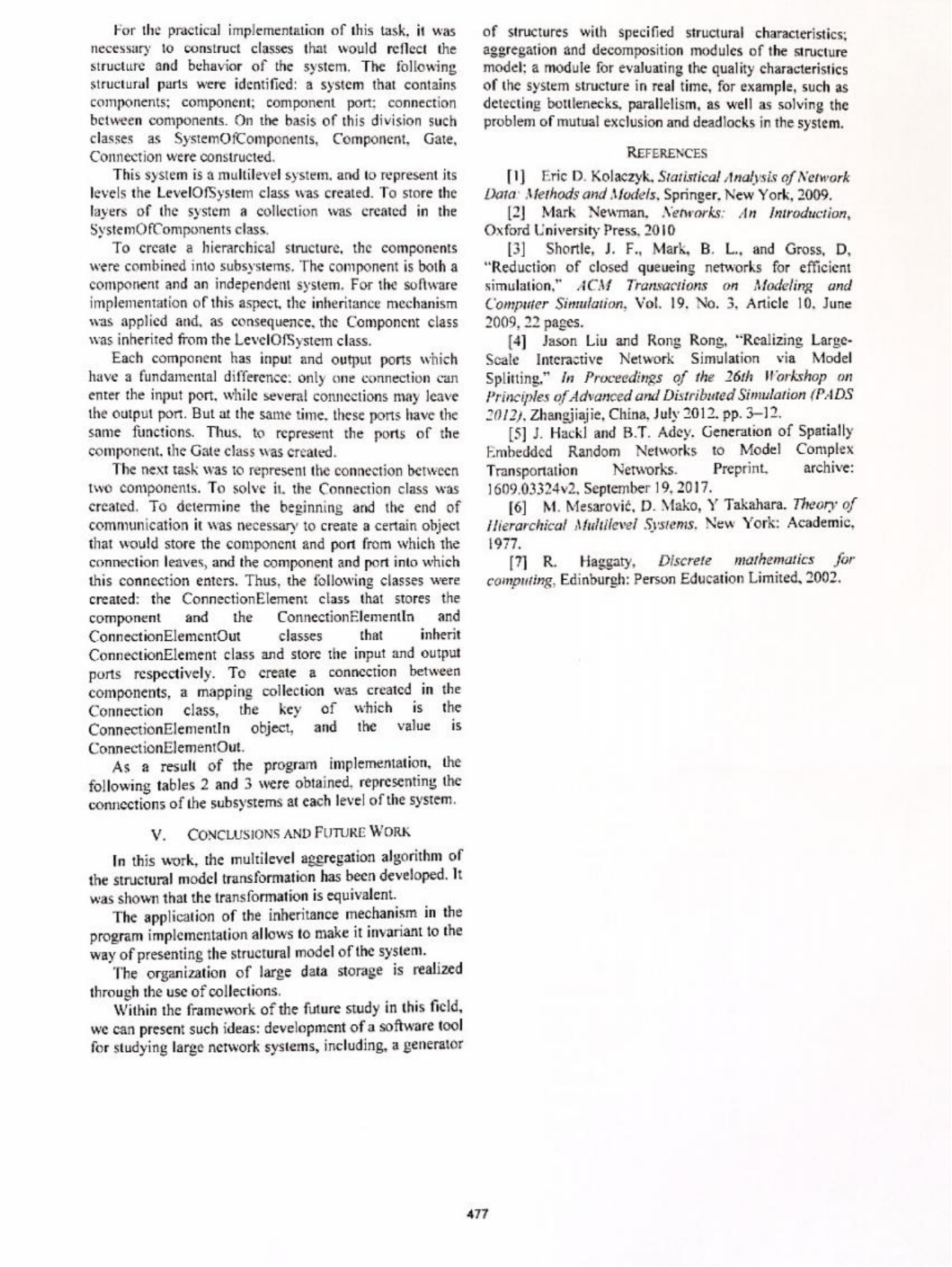
IV. PROGRAM IMPLEMENTATION

This section is devoted to the description of the program implementation. The algorithm of multilevel aggregation of the network system has been programed. The program is written in the Java programming language and consists of fourteen classes, ten of which display the structure and functionality of the program, four are test classes. Simplified UML diagram with class names and fields is represented in Fig. 1.

Figure 1. Simplified UML diagram with class names and fields

476

Рисунок Б.6 – Тези доповіді конференції « Formal transformations of structural models of complex network systems»



For the practical implementation of this task, it was necessary to construct classes that would reflect the structure and behavior of the system. The following structural parts were identified: a system that contains components; component; component port; connection between components. On the basis of this division such classes as SystemOfComponents, Component, Gate, Connection were constructed.

This system is a multilevel system, and to represent its levels the LevelOfSystem class was created. To store the layers of the system a collection was created in the SystemOfComponents class.

To create a hierarchical structure, the components were combined into subsystems. The component is both a component and an independent system. For the software implementation of this aspect, the inheritance mechanism was applied and, as consequence, the Component class was inherited from the LevelOfSystem class.

Each component has input and output ports which have a fundamental difference: only one connection can enter the input port, while several connections may leave the output port. But at the same time, these ports have the same functions. Thus, to represent the ports of the component, the Gate class was created.

The next task was to represent the connection between two components. To solve it, the Connection class was created. To determine the beginning and the end of communication it was necessary to create a certain object that would store the component and port from which the connection leaves, and the component and port into which this connection enters. Thus, the following classes were created: the ConnectionElement class that stores the component and the ConnectionElementIn and ConnectionElementOut classes that inherit ConnectionElement class and store the input and output ports respectively. To create a connection between components, a mapping collection was created in the Connection class, the key of which is the ConnectionElementIn object, and the value is ConnectionElementOut.

As a result of the program implementation, the following tables 2 and 3 were obtained, representing the connections of the subsystems at each level of the system.

V. CONCLUSIONS AND FUTURE WORK

In this work, the multilevel aggregation algorithm of the structural model transformation has been developed. It was shown that the transformation is equivalent.

The application of the inheritance mechanism in the program implementation allows to make it invariant to the way of presenting the structural model of the system.

The organization of large data storage is realized through the use of collections.

Within the framework of the future study in this field, we can present such ideas: development of a software tool for studying large network systems, including, a generator of structures with specified structural characteristics; aggregation and decomposition modules of the structure model; a module for evaluating the quality characteristics of the system structure in real time, for example, such as detecting bottlenecks, parallelism, as well as solving the problem of mutual exclusion and deadlocks in the system.

REFERENCES

[1] Eric D. Kolaczyk, *Statistical Analysis of Network Data: Methods and Models*, Springer, New York, 2009.

[2] Mark Newman, *Networks: An Introduction*, Oxford University Press, 2010

[3] Shortle, J. F., Mark, B. L., and Gross, D, "Reduction of closed queueing networks for efficient simulation," *ACM Transactions on Modeling and Computer Simulation*, Vol. 19, No. 3, Article 10, June 2009, 22 pages.

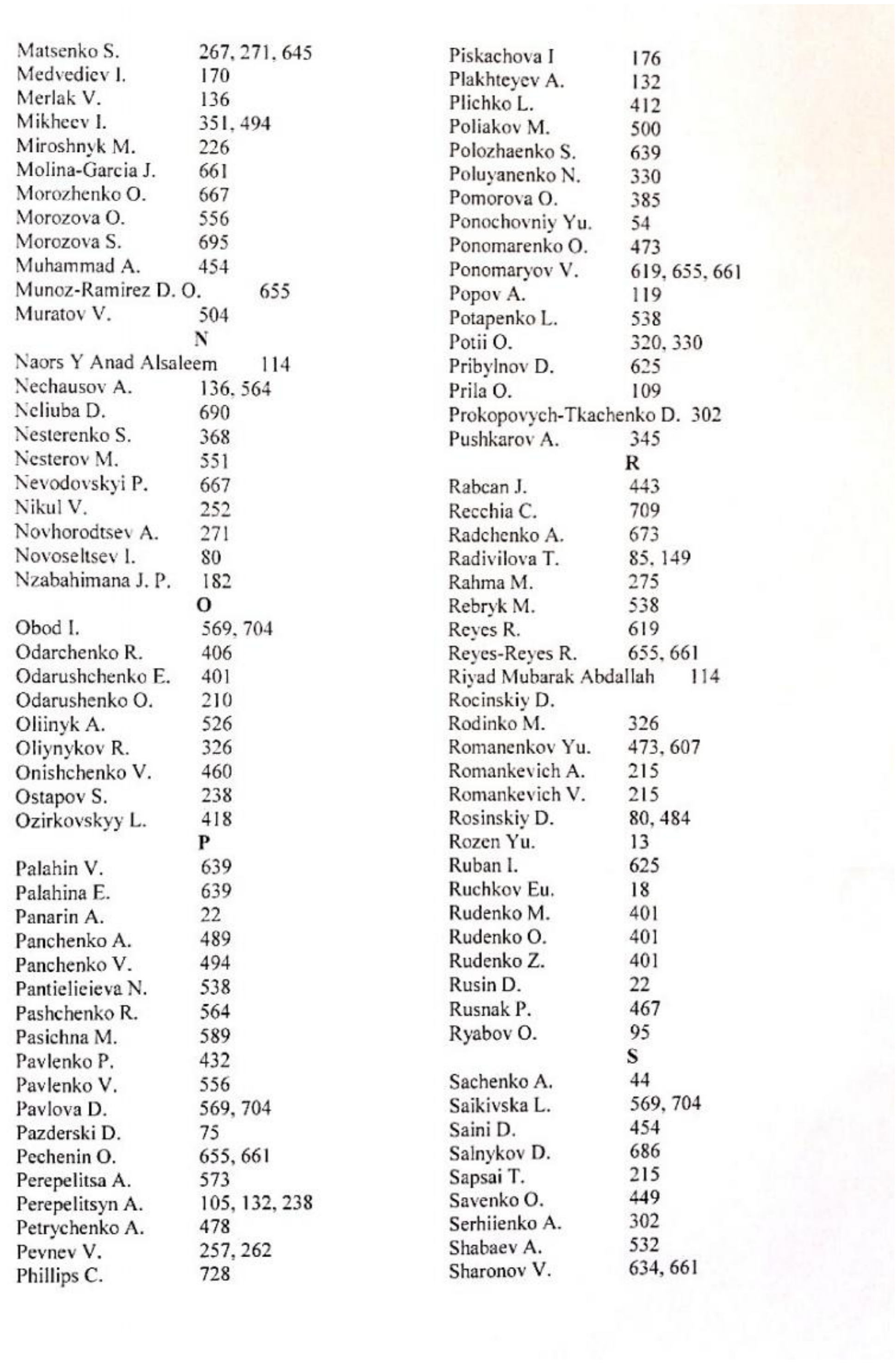
[4] Jason Liu and Rong Rong, "Realizing Large-Scale Interactive Network Simulation via Model Splitting," *In Proceedings of the 26th Workshop on Principles of Advanced and Distributed Simulation (PADS 2012)*, Zhangjiajie, China, July 2012, pp. 3–12.

[5] J. Hackl and B.T. Adey. Generation of Spatially Embedded Random Networks to Model Complex Transportation Networks. Preprint, archive: 1609.03324v2, September 19, 2017.

[6] M. Mesarović, D. Mako, Y Takahara. *Theory of Hierarchical Multilevel Systems*. New York: Academic, 1977.

[7] R. Haggaty, *Discrete mathematics for computing*, Edinburgh: Person Education Limited, 2002.

Рисунок Б.7 – Тези доповіді конференції « Formal transformations of structural models of complex network systems»



Matsenko S. 267, 271, 645
Medvediev I. 170
Merlak V. 136
Mikheev I. 351, 494
Miroshnyk M. 226
Molina-Garcia J. 661
Morozhenko O. 667
Morozova O. 556
Morozova S. 695
Muhammad A. 454
Munoz-Ramirez D. O. 655
Muratov V. 504

N

Naors Y Anad Alsaleem 114
Nechausov A. 136, 564
Neliuba D. 690
Nesterenko S. 368
Nesterov M. 551
Nevodovskyi P. 667
Nikul V. 252
Novhorodtsev A. 271
Novoseltsev I. 80
Nzabahimana J. P. 182

O

Obod I. 569, 704
Odarchenko R. 406
Odarushchenko E. 401
Odarushenko O. 210
Oliinyk A. 526
Oliynykov R. 326
Onishchenko V. 460
Ostapov S. 238
Ozirkovskyy L. 418

P

Palahin V. 639
Palahina E. 639
Panarin A. 22
Panchenko A. 489
Panchenko V. 494
Pantielieieva N. 538
Pashchenko R. 564
Pasichna M. 589
Pavlenko P. 432
Pavlenko V. 556
Pavlova D. 569, 704
Pazderski D. 75
Pechenin O. 655, 661
Perepelitsa A. 573
Perepelitsyn A. 105, 132, 238
Petrychenko A. 478
Pevnev V. 257, 262
Phillips C. 728
Piskachova I 176
Plakhteyev A. 132
Plichko L. 412
Poliakov M. 500
Polozhaenko S. 639
Poluyanenko N. 330
Pomorova O. 385
Ponochovniy Yu. 54
Ponomarenko O. 473
Ponomaryov V. 619, 655, 661
Popov A. 119
Potapenko L. 538
Potii O. 320, 330
Pribylnov D. 625
Prila O. 109
Prokopovych-Tkachenko D. 302
Pushkarov A. 345

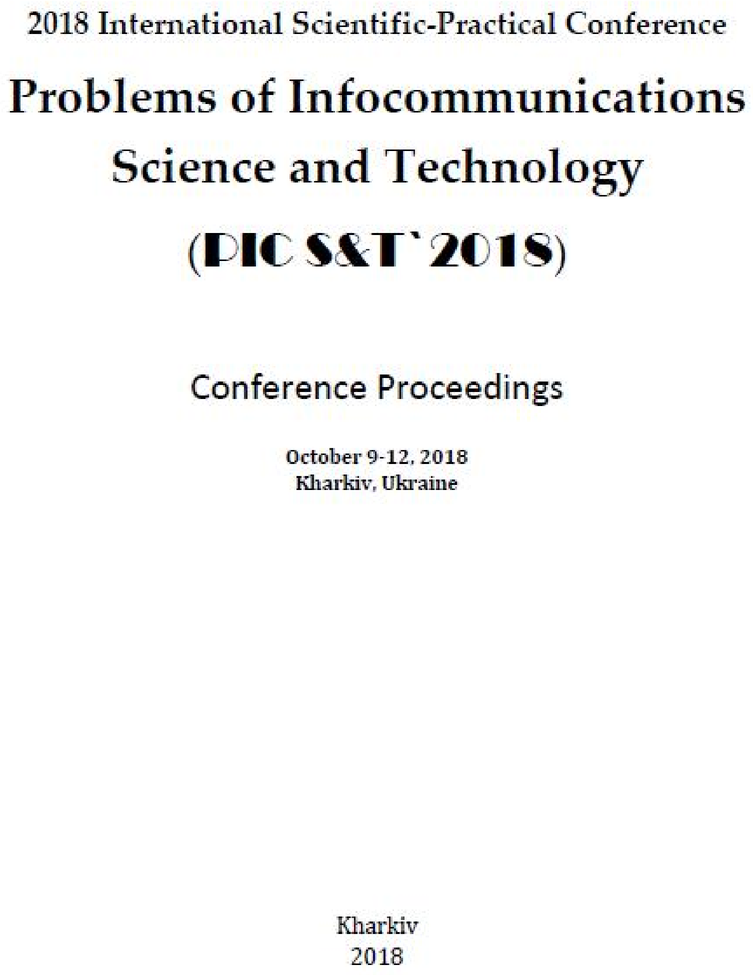
R

Rabcan J. 443
Recchia C. 709
Radchenko A. 673
Radivilova T. 85, 149
Rahma M. 275
Rebryk M. 538
Reyes R. 619
Reyes-Reyes R. 655, 661
Riyad Mubarak Abdallah 114
Rocinskiy D.
Rodinko M. 326
Romanenkov Yu. 473, 607
Romankevich A. 215
Romankevich V. 215
Rosinskiy D. 80, 484
Rozen Yu. 13
Ruban I. 625
Ruchkov Eu. 18
Rudenko M. 401
Rudenko O. 401
Rudenko Z. 401
Rusin D. 22
Rusnak P. 467
Ryabov O. 95

S

Sachenko A. 44
Saikivska L. 569, 704
Saini D. 454
Salnykov D. 686
Sapsai T. 215
Savenko O. 449
Serhiienko A. 302
Shabaev A. 532
Sharonov V. 634, 661

Рисунок Б.8 – Тези доповіді конференції « Formal transformations of structural models of complex network systems»



2018 International Scientific-Practical Conference

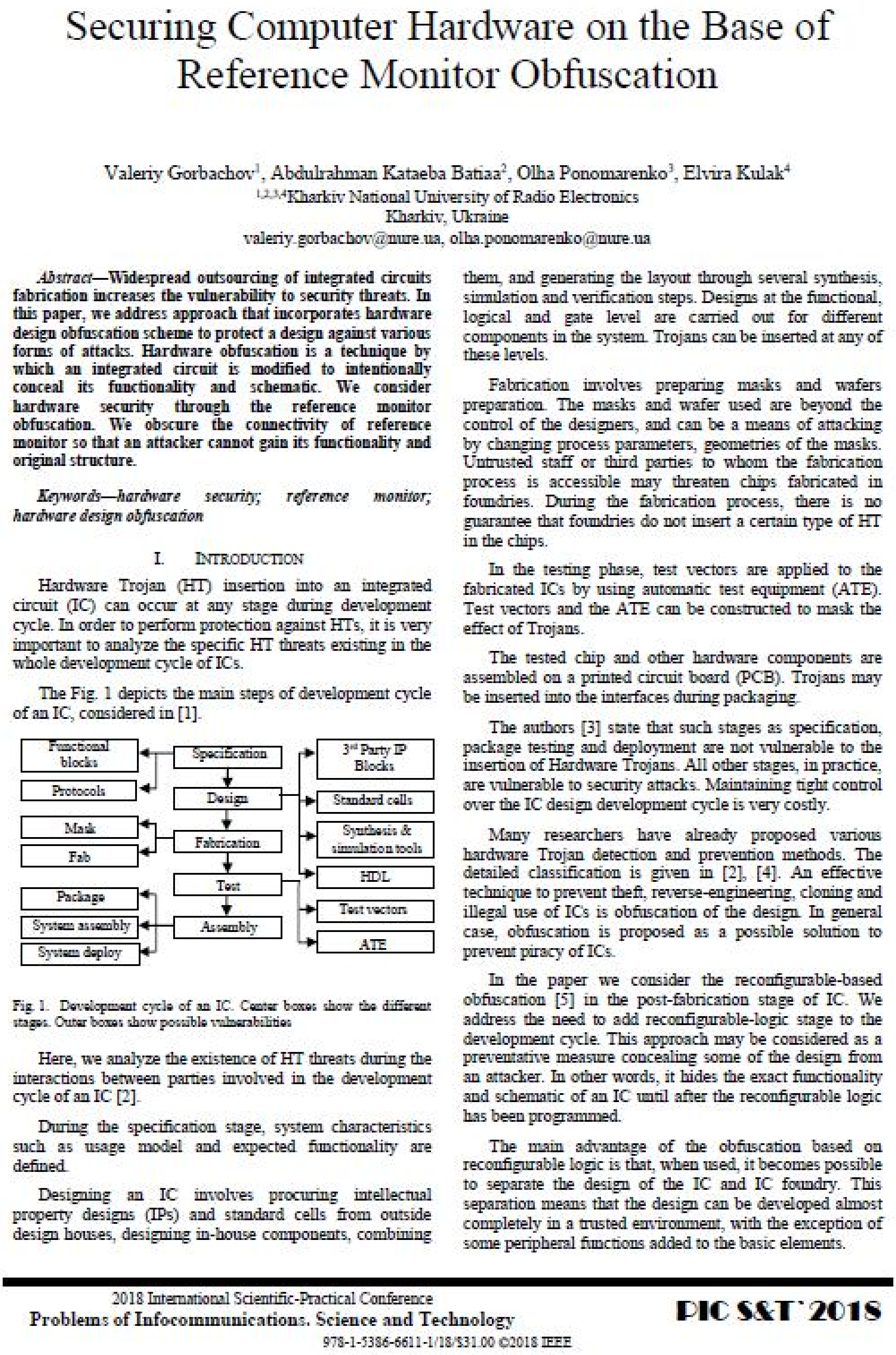
Problems of Infocommunications
Science and Technology
(PIC S&T`2018)

Conference Proceedings

October 9-12, 2018
Kharkiv, Ukraine

Kharkiv
2018

Рисунок Б.9 – Тези доповіді конференції «Securing Computer Hardware on the Base of Reference Monitor Obfuscation»



Securing Computer Hardware on the Base of Reference Monitor Obfuscation

Valeriy Gorbachov[1], Abdulrahman Kataeba Batiaa[2], Olha Ponomarenko[3], Elvira Kulak[4]
[1,2,3,4]Kharkiv National University of Radio Electronics
Kharkiv, Ukraine
valeriy.gorbachov@nure.ua, olha.ponomarenko@nure.ua

***Abstract*—Widespread outsourcing of integrated circuits fabrication increases the vulnerability to security threats. In this paper, we address approach that incorporates hardware design obfuscation scheme to protect a design against various forms of attacks. Hardware obfuscation is a technique by which an integrated circuit is modified to intentionally conceal its functionality and schematic. We consider hardware security through the reference monitor obfuscation. We obscure the connectivity of reference monitor so that an attacker cannot gain its functionality and original structure.**

***Keywords*—hardware security; reference monitor; hardware design obfuscation**

I. INTRODUCTION

Hardware Trojan (HT) insertion into an integrated circuit (IC) can occur at any stage during development cycle. In order to perform protection against HTs, it is very important to analyze the specific HT threats existing in the whole development cycle of ICs.

The Fig. 1 depicts the main steps of development cycle of an IC, considered in [1].

Fig. 1. Development cycle of an IC. Center boxes show the different stages. Outer boxes show possible vulnerabilities

Here, we analyze the existence of HT threats during the interactions between parties involved in the development cycle of an IC [2].

During the specification stage, system characteristics such as usage model and expected functionality are defined.

Designing an IC involves procuring intellectual property designs (IPs) and standard cells from outside design houses, designing in-house components, combining them, and generating the layout through several synthesis, simulation and verification steps. Designs at the functional, logical and gate level are carried out for different components in the system. Trojans can be inserted at any of these levels.

Fabrication involves preparing masks and wafers preparation. The masks and wafer used are beyond the control of the designers, and can be a means of attacking by changing process parameters, geometries of the masks. Untrusted staff or third parties to whom the fabrication process is accessible may threaten chips fabricated in foundries. During the fabrication process, there is no guarantee that foundries do not insert a certain type of HT in the chips.

In the testing phase, test vectors are applied to the fabricated ICs by using automatic test equipment (ATE). Test vectors and the ATE can be constructed to mask the effect of Trojans.

The tested chip and other hardware components are assembled on a printed circuit board (PCB). Trojans may be inserted into the interfaces during packaging.

The authors [3] state that such stages as specification, package testing and deployment are not vulnerable to the insertion of Hardware Trojans. All other stages, in practice, are vulnerable to security attacks. Maintaining tight control over the IC design development cycle is very costly.

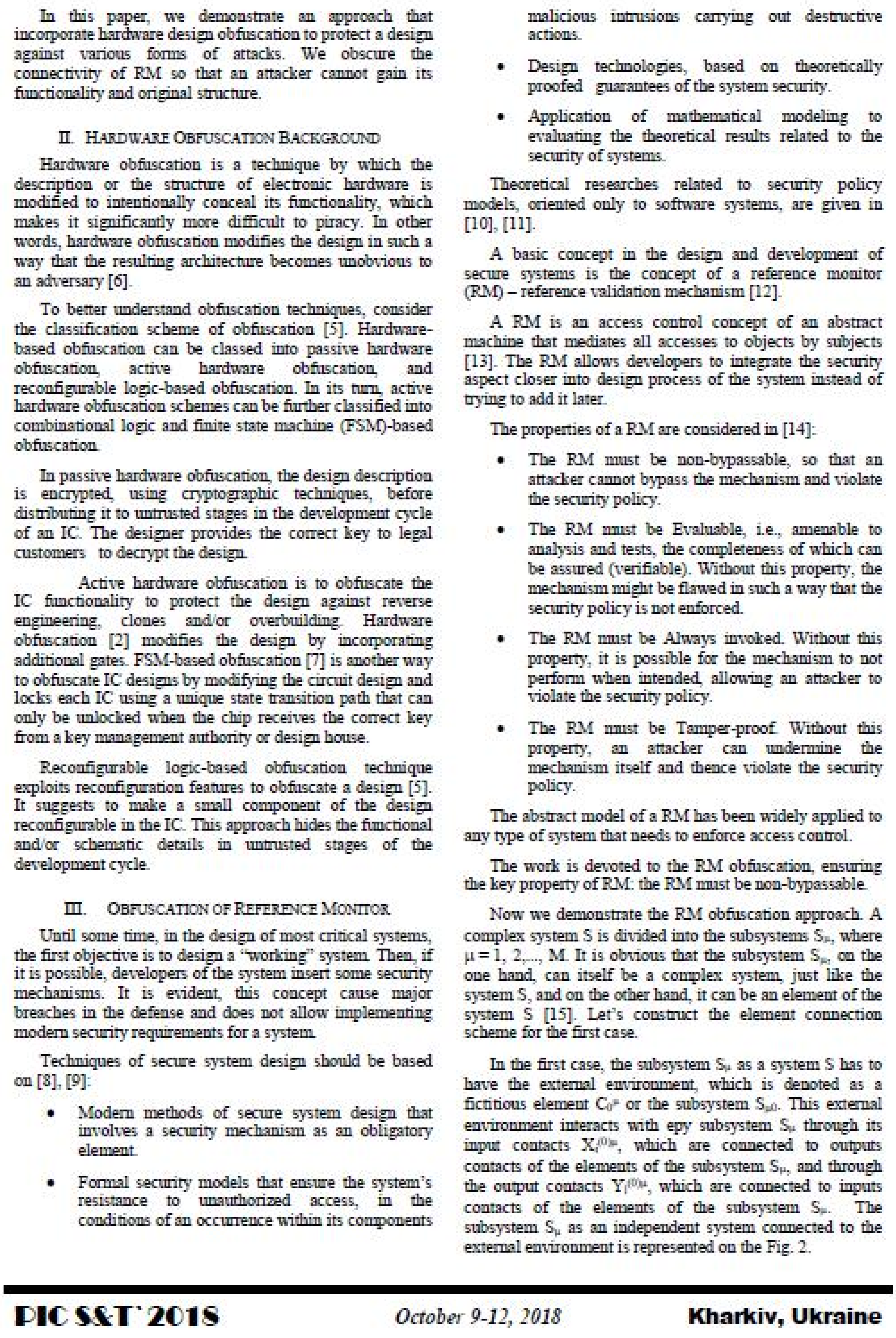
Many researchers have already proposed various hardware Trojan detection and prevention methods. The detailed classification is given in [2], [4]. An effective technique to prevent theft, reverse-engineering, cloning and illegal use of ICs is obfuscation of the design. In general case, obfuscation is proposed as a possible solution to prevent piracy of ICs.

In the paper we consider the reconfigurable-based obfuscation [5] in the post-fabrication stage of IC. We address the need to add reconfigurable-logic stage to the development cycle. This approach may be considered as a preventative measure concealing some of the design from an attacker. In other words, it hides the exact functionality and schematic of an IC until after the reconfigurable logic has been programmed.

The main advantage of the obfuscation based on reconfigurable logic is that, when used, it becomes possible to separate the design of the IC and IC foundry. This separation means that the design can be developed almost completely in a trusted environment, with the exception of some peripheral functions added to the basic elements.

2018 International Scientific-Practical Conference
Problems of Infocommunications. Science and Technology
PIC S&T`2018
978-1-5386-6611-1/18/$31.00 ©2018 IEEE

Рисунок Б.10 – Тези доповіді конференції «Securing Computer Hardware on the Base of Reference Monitor Obfuscation»



In this paper, we demonstrate an approach that incorporate hardware design obfuscation to protect a design against various forms of attacks. We obscure the connectivity of RM so that an attacker cannot gain its functionality and original structure.

II. HARDWARE OBFUSCATION BACKGROUND

Hardware obfuscation is a technique by which the description or the structure of electronic hardware is modified to intentionally conceal its functionality, which makes it significantly more difficult to piracy. In other words, hardware obfuscation modifies the design in such a way that the resulting architecture becomes unobvious to an adversary [6].

To better understand obfuscation techniques, consider the classification scheme of obfuscation [5]. Hardware-based obfuscation can be classed into passive hardware obfuscation, active hardware obfuscation, and reconfigurable logic-based obfuscation. In its turn, active hardware obfuscation schemes can be further classified into combinational logic and finite state machine (FSM)-based obfuscation.

In passive hardware obfuscation, the design description is encrypted, using cryptographic techniques, before distributing it to untrusted stages in the development cycle of an IC. The designer provides the correct key to legal customers to decrypt the design.

Active hardware obfuscation is to obfuscate the IC functionality to protect the design against reverse engineering, clones and/or overbuilding. Hardware obfuscation [2] modifies the design by incorporating additional gates. FSM-based obfuscation [7] is another way to obfuscate IC designs by modifying the circuit design and locks each IC using a unique state transition path that can only be unlocked when the chip receives the correct key from a key management authority or design house.

Reconfigurable logic-based obfuscation technique exploits reconfiguration features to obfuscate a design [5]. It suggests to make a small component of the design reconfigurable in the IC. This approach hides the functional and/or schematic details in untrusted stages of the development cycle.

III. OBFUSCATION OF REFERENCE MONITOR

Until some time, in the design of most critical systems, the first objective is to design a "working" system. Then, if it is possible, developers of the system insert some security mechanisms. It is evident, this concept cause major breaches in the defense and does not allow implementing modern security requirements for a system.

Techniques of secure system design should be based on [8], [9]:

- Modern methods of secure system design that involves a security mechanism as an obligatory element.
- Formal security models that ensure the system's resistance to unauthorized access, in the conditions of an occurrence within its components malicious intrusions carrying out destructive actions.
- Design technologies, based on theoretically proofed guarantees of the system security.
- Application of mathematical modeling to evaluating the theoretical results related to the security of systems.

Theoretical researches related to security policy models, oriented only to software systems, are given in [10], [11].

A basic concept in the design and development of secure systems is the concept of a reference monitor (RM) – reference validation mechanism [12].

A RM is an access control concept of an abstract machine that mediates all accesses to objects by subjects [13]. The RM allows developers to integrate the security aspect closer into design process of the system instead of trying to add it later.

The properties of a RM are considered in [14]:

- The RM must be non-bypassable, so that an attacker cannot bypass the mechanism and violate the security policy.
- The RM must be Evaluable, i.e., amenable to analysis and tests, the completeness of which can be assured (verifiable). Without this property, the mechanism might be flawed in such a way that the security policy is not enforced.
- The RM must be Always invoked. Without this property, it is possible for the mechanism to not perform when intended, allowing an attacker to violate the security policy.
- The RM must be Tamper-proof. Without this property, an attacker can undermine the mechanism itself and thence violate the security policy.

The abstract model of a RM has been widely applied to any type of system that needs to enforce access control.

The work is devoted to the RM obfuscation, ensuring the key property of RM: the RM must be non-bypassable.

Now we demonstrate the RM obfuscation approach. A complex system S is divided into the subsystems S_μ, where $\mu = 1, 2, \ldots, M$. It is obvious that the subsystem S_μ, on the one hand, can itself be a complex system, just like the system S, and on the other hand, it can be an element of the system S [15]. Let's construct the element connection scheme for the first case.

In the first case, the subsystem S_μ as a system S has to have the external environment, which is denoted as a fictitious element C_0^μ or the subsystem $S_{\mu 0}$. This external environment interacts with epy subsystem S_μ through its input contacts $X_i^{(0)\mu}$, which are connected to outputs contacts of the elements of the subsystem S_μ, and through the output contacts $Y_i^{(0)\mu}$, which are connected to inputs contacts of the elements of the subsystem S_μ. The subsystem S_μ as an independent system connected to the external environment is represented on the Fig. 2.

Рисунок Б.11 – Тези доповіді конференції «Securing Computer Hardware on the Base of Reference Monitor Obfuscation»

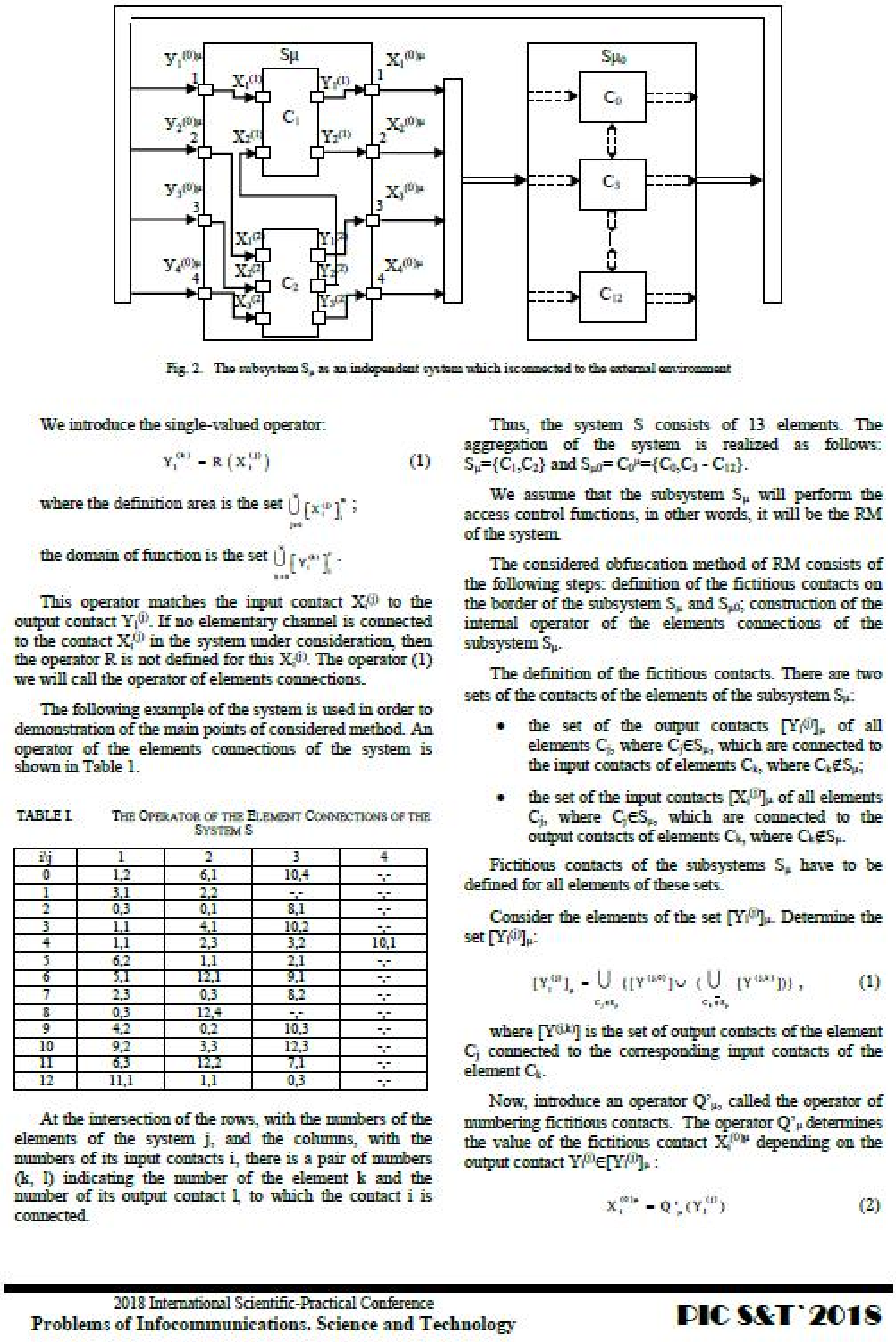


Fig. 2. The subsystem S_μ as an independent system which isconnected to the external environment

We introduce the single-valued operator:

$$Y_i^{(k)} = R\left(X_i^{(j)}\right) \qquad (1)$$

where the definition area is the set $\bigcup_{j=0}^{n}[X_i^{(j)}]_i^{n}$;

the domain of function is the set $\bigcup_{k=0}^{n}[Y_i^{(k)}]_i^{n}$.

This operator matches the input contact $X_i^{(j)}$ to the output contact $Y_l^{(j)}$. If no elementary channel is connected to the contact $X_i^{(j)}$ in the system under consideration, then the operator R is not defined for this $X_i^{(j)}$. The operator (1) we will call the operator of elements connections.

The following example of the system is used in order to demonstration of the main points of considered method. An operator of the elements connections of the system is shown in Table 1.

TABLE I. THE OPERATOR OF THE ELEMENT CONNECTIONS OF THE SYSTEM S

i\j	1	2	3	4
0	1,2	6,1	10,4	-,-
1	3,1	2,2	-,-	-,-
2	0,3	0,1	8,1	-,-
3	1,1	4,1	10,2	-,-
4	1,1	2,3	3,2	10,1
5	6,2	1,1	2,1	-,-
6	5,1	12,1	9,1	-,-
7	2,3	0,3	8,2	-,-
8	0,3	12,4	-,-	-,-
9	4,2	0,2	10,3	-,-
10	9,2	3,3	12,3	-,-
11	6,3	12,2	7,1	-,-
12	11,1	1,1	0,3	-,-

At the intersection of the rows, with the numbers of the elements of the system j, and the columns, with the numbers of its input contacts i, there is a pair of numbers (k, l) indicating the number of the element k and the number of its output contact l, to which the contact i is connected.

Thus, the system S consists of 13 elements. The aggregation of the system is realized as follows: $S_\mu=\{C_1,C_2\}$ and $S_{\mu 0}= C_0^{\mu}=\{C_0,C_3 - C_{12}\}$.

We assume that the subsystem S_μ will perform the access control functions, in other words, it will be the RM of the system.

The considered obfuscation method of RM consists of the following steps: definition of the fictitious contacts on the border of the subsystem S_μ and $S_{\mu 0}$; construction of the internal operator of the elements connections of the subsystem S_μ.

The definition of the fictitious contacts. There are two sets of the contacts of the elements of the subsystem S_μ:

- the set of the output contacts $[Y_l^{(j)}]_\mu$ of all elements C_j, where $C_j \in S_\mu$, which are connected to the input contacts of elements C_k, where $C_k \notin S_\mu$;
- the set of the input contacts $[X_i^{(j)}]_\mu$ of all elements C_j, where $C_j \in S_\mu$, which are connected to the output contacts of elements C_k, where $C_k \notin S_\mu$.

Fictitious contacts of the subsystems S_μ have to be defined for all elements of these sets.

Consider the elements of the set $[Y_l^{(j)}]_\mu$. Determine the set $[Y_l^{(j)}]_\mu$:

$$[Y_i^{(j)}]_\mu = \bigcup_{C_j \in S_\mu} ([Y^{(j,0)}] \cup (\bigcup_{C_k \bar{\in} S_\mu} [Y^{(j,k)}])) , \qquad (1)$$

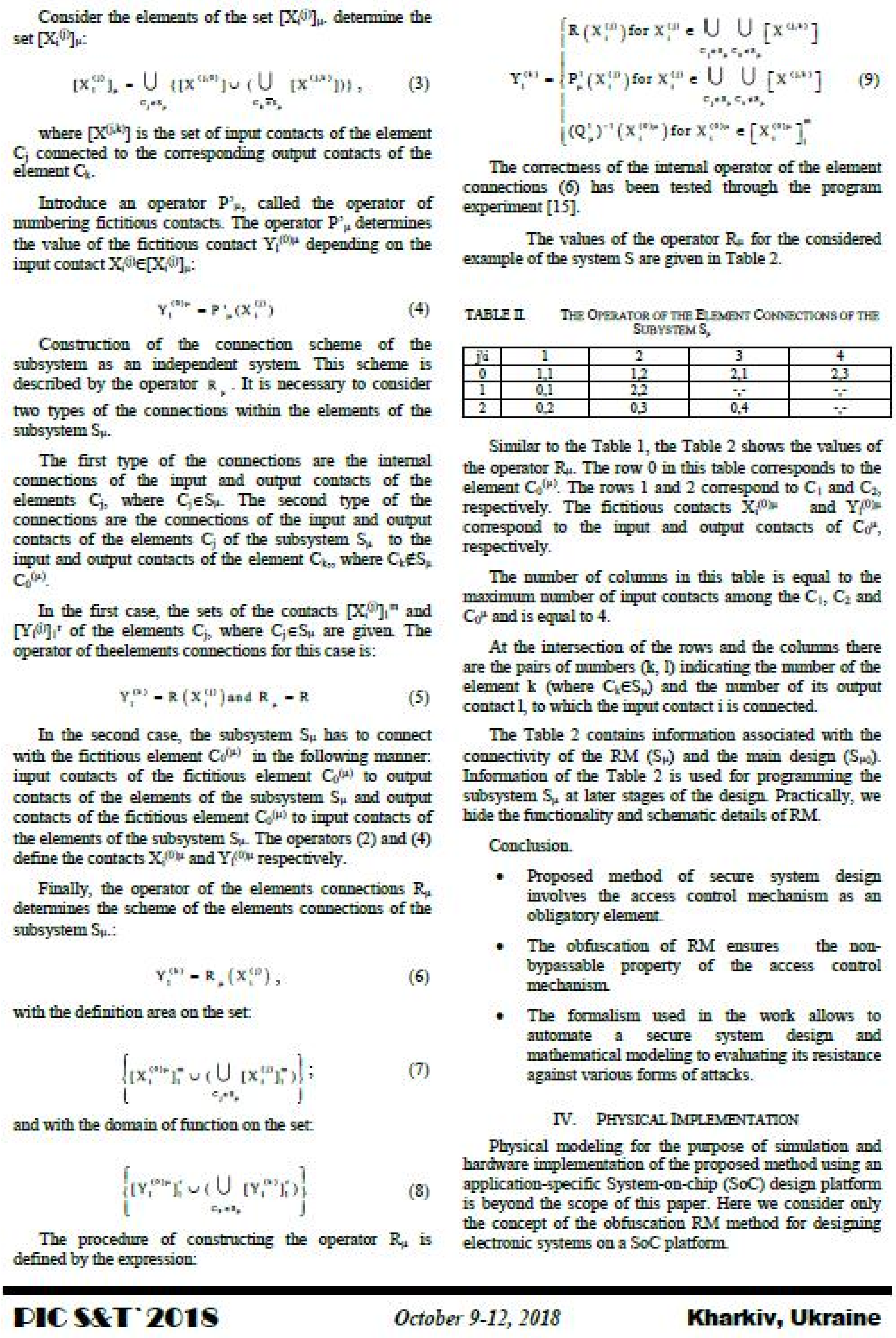
where $[Y^{(j,k)}]$ is the set of output contacts of the element C_j connected to the corresponding input contacts of the element C_k.

Now, introduce an operator Q'_μ, called the operator of numbering fictitious contacts. The operator Q'_μ determines the value of the fictitious contact $X_i^{(0)\mu}$ depending on the output contact $Y_l^{(j)} \in [Y_l^{(j)}]_\mu$:

$$X_i^{(0)\mu} = Q'_\mu(Y_i^{(j)}) \qquad (2)$$

2018 International Scientific-Practical Conference
Problems of Infocommunications. Science and Technology **PIC S&T`2018**

Рисунок Б.12 – Тези доповіді конференції «Securing Computer Hardware on the Base of Reference Monitor Obfuscation»



Consider the elements of the set $[X_i^{(j)}]_\mu$. determine the set $[X_i^{(j)}]_\mu$:

$$[X_i^{(j)}]_\mu = \bigcup_{C_j \in S_\mu} \{[X^{(j,k)}] \cup (\bigcup_{C_k \notin S_\mu} [X^{(j,k)}])\}, \qquad (3)$$

where $[X^{(j,k)}]$ is the set of input contacts of the element C_j connected to the corresponding output contacts of the element C_k.

Introduce an operator P'_μ, called the operator of numbering fictitious contacts. The operator P'_μ determines the value of the fictitious contact $Y_l^{(0)\mu}$ depending on the input contact $X_i^{(j)} \in [X_i^{(j)}]_\mu$:

$$Y_l^{(0)\mu} = P'_\mu(X_i^{(j)}) \qquad (4)$$

Construction of the connection scheme of the subsystem as an independent system. This scheme is described by the operator R_μ. It is necessary to consider two types of the connections within the elements of the subsystem S_μ.

The first type of the connections are the internal connections of the input and output contacts of the elements C_j, where $C_j \in S_\mu$. The second type of the connections are the connections of the input and output contacts of the elements C_j of the subsystem S_μ to the input and output contacts of the element C_k,, where $C_k \notin S_\mu$ $C_0^{(\mu)}$.

In the first case, the sets of the contacts $[X_i^{(j)}]_i^m$ and $[Y_l^{(j)}]_l^r$ of the elements C_j, where $C_j \in S_\mu$ are given. The operator of theelements connections for this case is:

$$Y_l^{(k)} = R(X_i^{(j)}) \text{ and } R_\mu = R \qquad (5)$$

In the second case, the subsystem S_μ has to connect with the fictitious element $C_0^{(\mu)}$ in the following manner: input contacts of the fictitious element $C_0^{(\mu)}$ to output contacts of the elements of the subsystem S_μ and output contacts of the fictitious element $C_0^{(\mu)}$ to input contacts of the elements of the subsystem S_μ. The operators (2) and (4) define the contacts $X_i^{(0)\mu}$ and $Y_l^{(0)\mu}$ respectively.

Finally, the operator of the elements connections R_μ determines the scheme of the elements connections of the subsystem S_μ.:

$$Y_l^{(k)} = R_\mu(X_i^{(j)}), \qquad (6)$$

with the definition area on the set:

$$\left\{[X_i^{(0)\mu}]_i^m \cup (\bigcup_{C_j \in S_\mu} [X_i^{(j)}]_i^m)\right\}; \qquad (7)$$

and with the domain of function on the set:

$$\left\{[Y_l^{(0)\mu}]_l^r \cup (\bigcup_{C_k \in S_\mu} [Y_l^{(k)}]_l^r)\right\} \qquad (8)$$

The procedure of constructing the operator R_μ is defined by the expression:

$$Y_l^{(k)} = \begin{cases} R(X_i^{(j)}) \text{ for } X_i^{(j)} \in \bigcup_{C_j \in S_\mu} \bigcup_{C_k \in S_\mu} [X^{(j,k)}] \\ P_\mu^1(X_i^{(j)}) \text{ for } X_i^{(j)} \in \bigcup_{C_j \in S_\mu} \bigcup_{C_k \notin S_\mu} [X^{(j,k)}] \\ (Q_\mu^1)^{-1}(X_i^{(0)\mu}) \text{ for } X_i^{(0)\mu} \in [X_i^{(0)\mu}]_i^m \end{cases} \qquad (9)$$

The correctness of the internal operator of the element connections (6) has been tested through the program experiment [15].

The values of the operator R_μ for the considered example of the system S are given in Table 2.

TABLE II. THE OPERATOR OF THE ELEMENT CONNECTIONS OF THE SUBYSTEM S_μ

j\i	1	2	3	4
0	1,1	1,2	2,1	2,3
1	0,1	2,2	-,-	-,-
2	0,2	0,3	0,4	-,-

Similar to the Table 1, the Table 2 shows the values of the operator R_μ. The row 0 in this table corresponds to the element $C_0^{(\mu)}$. The rows 1 and 2 correspond to C_1 and C_2, respectively. The fictitious contacts $X_i^{(0)\mu}$ and $Y_l^{(0)\mu}$ correspond to the input and output contacts of C_0^μ, respectively.

The number of columns in this table is equal to the maximum number of input contacts among the C_1, C_2 and C_0^μ and is equal to 4.

At the intersection of the rows and the columns there are the pairs of numbers (k, l) indicating the number of the element k (where $C_k \in S_\mu$) and the number of its output contact l, to which the input contact i is connected.

The Table 2 contains information associated with the connectivity of the RM (S_μ) and the main design ($S_{\mu 0}$). Information of the Table 2 is used for programming the subsystem S_μ at later stages of the design. Practically, we hide the functionality and schematic details of RM.

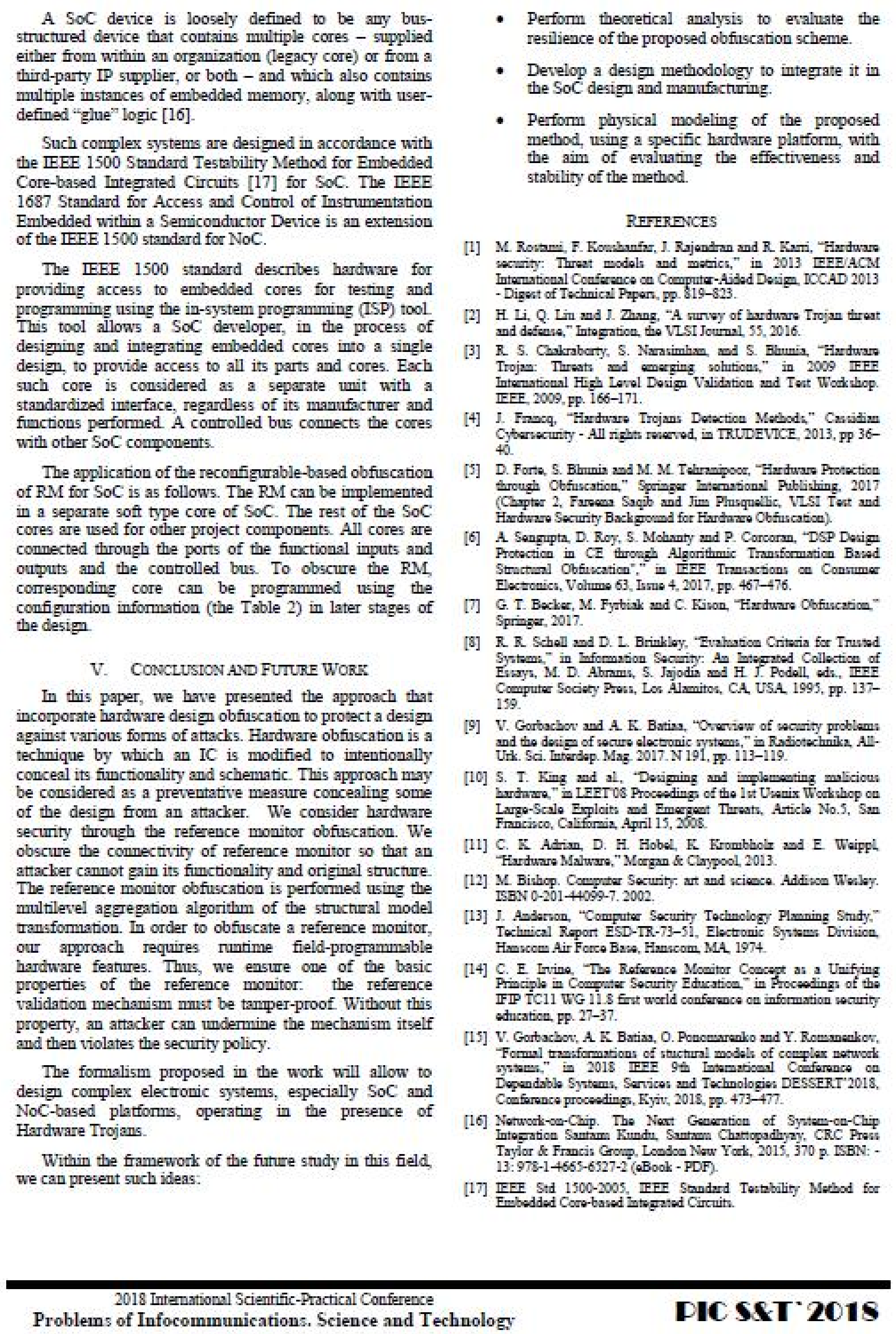
Conclusion.

- Proposed method of secure system design involves the access control mechanism as an obligatory element.
- The obfuscation of RM ensures the non-bypassable property of the access control mechanism.
- The formalism used in the work allows to automate a secure system design and mathematical modeling to evaluating its resistance against various forms of attacks.

IV. PHYSICAL IMPLEMENTATION

Physical modeling for the purpose of simulation and hardware implementation of the proposed method using an application-specific System-on-chip (SoC) design platform is beyond the scope of this paper. Here we consider only the concept of the obfuscation RM method for designing electronic systems on a SoC platform.

Рисунок Б.13 – Тези доповіді конференції «Securing Computer Hardware on the Base of Reference Monitor Obfuscation»



A SoC device is loosely defined to be any bus-structured device that contains multiple cores – supplied either from within an organization (legacy core) or from a third-party IP supplier, or both – and which also contains multiple instances of embedded memory, along with user-defined "glue" logic [16].

Such complex systems are designed in accordance with the IEEE 1500 Standard Testability Method for Embedded Core-based Integrated Circuits [17] for SoC. The IEEE 1687 Standard for Access and Control of Instrumentation Embedded within a Semiconductor Device is an extension of the IEEE 1500 standard for NoC.

The IEEE 1500 standard describes hardware for providing access to embedded cores for testing and programming using the in-system programming (ISP) tool. This tool allows a SoC developer, in the process of designing and integrating embedded cores into a single design, to provide access to all its parts and cores. Each such core is considered as a separate unit with a standardized interface, regardless of its manufacturer and functions performed. A controlled bus connects the cores with other SoC components.

The application of the reconfigurable-based obfuscation of RM for SoC is as follows. The RM can be implemented in a separate soft type core of SoC. The rest of the SoC cores are used for other project components. All cores are connected through the ports of the functional inputs and outputs and the controlled bus. To obscure the RM, corresponding core can be programmed using the configuration information (the Table 2) in later stages of the design.

V. CONCLUSION AND FUTURE WORK

In this paper, we have presented the approach that incorporate hardware design obfuscation to protect a design against various forms of attacks. Hardware obfuscation is a technique by which an IC is modified to intentionally conceal its functionality and schematic. This approach may be considered as a preventative measure concealing some of the design from an attacker. We consider hardware security through the reference monitor obfuscation. We obscure the connectivity of reference monitor so that an attacker cannot gain its functionality and original structure. The reference monitor obfuscation is performed using the multilevel aggregation algorithm of the structural model transformation. In order to obfuscate a reference monitor, our approach requires runtime field-programmable hardware features. Thus, we ensure one of the basic properties of the reference monitor: the reference validation mechanism must be tamper-proof. Without this property, an attacker can undermine the mechanism itself and then violates the security policy.

The formalism proposed in the work will allow to design complex electronic systems, especially SoC and NoC-based platforms, operating in the presence of Hardware Trojans.

Within the framework of the future study in this field, we can present such ideas:

- Perform theoretical analysis to evaluate the resilience of the proposed obfuscation scheme.
- Develop a design methodology to integrate it in the SoC design and manufacturing.
- Perform physical modeling of the proposed method, using a specific hardware platform, with the aim of evaluating the effectiveness and stability of the method.

REFERENCES

[1] M. Rostami, F. Koushanfar, J. Rajendran and R. Karri, "Hardware security: Threat models and metrics," in 2013 IEEE/ACM International Conference on Computer-Aided Design, ICCAD 2013 - Digest of Technical Papers, pp. 819–823.

[2] H. Li, Q. Liu and J. Zhang, "A survey of hardware Trojan threat and defense," Integration, the VLSI Journal, 55, 2016.

[3] R. S. Chakraborty, S. Narasimhan, and S. Bhunia, "Hardware Trojan: Threats and emerging solutions," in 2009 IEEE International High Level Design Validation and Test Workshop. IEEE, 2009, pp. 166–171.

[4] J. Francq, "Hardware Trojans Detection Methods," Cassidian Cybersecurity - All rights reserved, in TRUDEVICE, 2013, pp 36–40.

[5] D. Forte, S. Bhunia and M. M. Tehranipoor, "Hardware Protection through Obfuscation," Springer International Publishing, 2017 (Chapter 2, Fareena Saqib and Jim Plusquellic, VLSI Test and Hardware Security Background for Hardware Obfuscation).

[6] A. Sengupta, D. Roy, S. Mohanty and P. Corcoran, "DSP Design Protection in CE through Algorithmic Transformation Based Structural Obfuscation"," in IEEE Transactions on Consumer Electronics, Volume 63, Issue 4, 2017, pp. 467–476.

[7] G. T. Becker, M. Fyrbiak and C. Kison, "Hardware Obfuscation," Springer, 2017.

[8] R. R. Schell and D. L. Brinkley, "Evaluation Criteria for Trusted Systems," in Information Security: An Integrated Collection of Essays, M. D. Abrams, S. Jajodia and H. J. Podell, eds., IEEE Computer Society Press, Los Alamitos, CA, USA, 1995, pp. 137–159.

[9] V. Gorbachov and A. K. Batiaa, "Overview of security problems and the design of secure electronic systems," in Radiotechnika, All-Urk. Sci. Interdep. Mag. 2017. N 191, pp. 113–119.

[10] S. T. King and al., "Designing and implementing malicious hardware," in LEET'08 Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats, Article No.5, San Francisco, California, April 15, 2008.

[11] C. K. Adrian, D. H. Hobel, K. Krombholz and E. Weippl, "Hardware Malware," Morgan & Claypool, 2013.

[12] M. Bishop. Computer Security: art and science. Addison Wesley. ISBN 0-201-44099-7. 2002.

[13] J. Anderson, "Computer Security Technology Planning Study," Technical Report ESD-TR-73–51, Electronic Systems Division, Hanscom Air Force Base, Hanscom, MA, 1974.

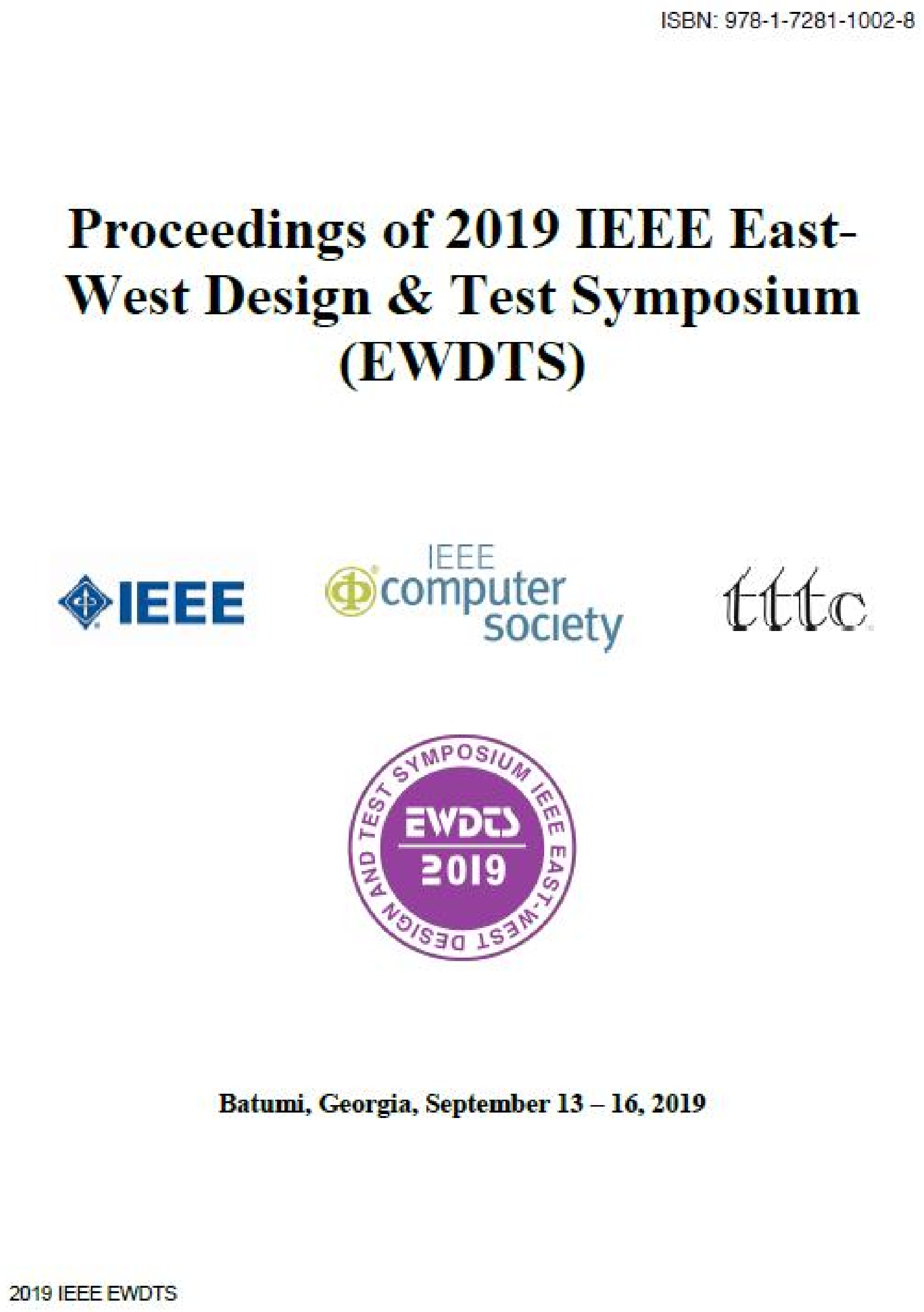
[14] C. E. Irvine, "The Reference Monitor Concept as a Unifying Principle in Computer Security Education," in Proceedings of the IFIP TC11 WG 11.8 first world conference on information security education, pp. 27–37.

[15] V. Gorbachov, A. K. Batiaa, O. Ponomarenko and Y. Romanenkov, "Formal transformations of stuctural models of complex network systems," in 2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies DESSERT'2018, Conference proceedings, Kyiv, 2018, pp. 473–477.

[16] Network-on-Chip. The Next Generation of System-on-Chip Integration Santanu Kundu, Santanu Chattopadhyay, CRC Press Taylor & Francis Group, London New York, 2015, 370 p. ISBN: -13: 978-1-4665-6527-2 (eBook - PDF).

[17] IEEE Std 1500-2005, IEEE Standard Testability Method for Embedded Core-based Integrated Circuits.

Рисунок Б.14 – Тези доповіді конференції «Securing Computer Hardware on the Base of Reference Monitor Obfuscation»



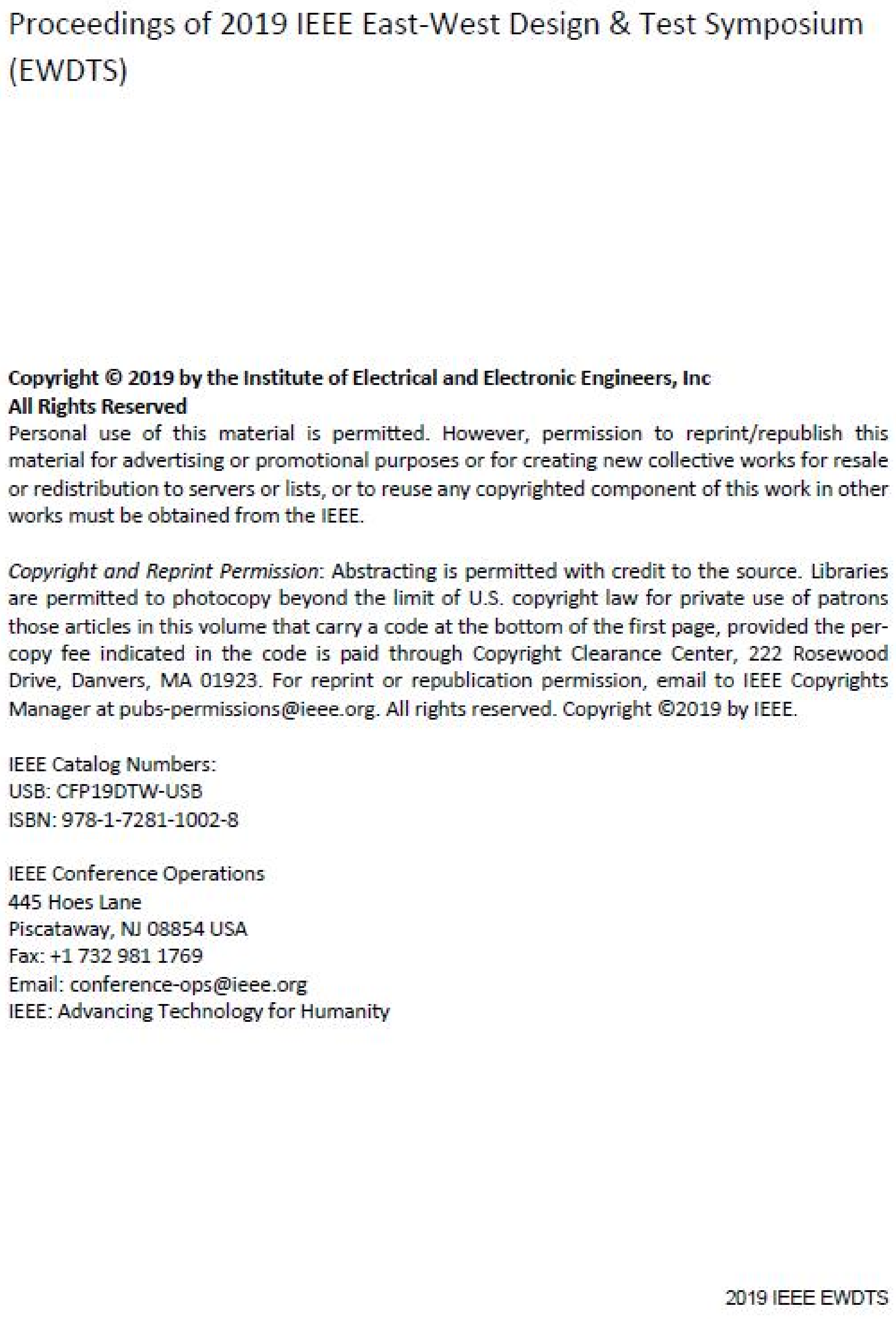
ISBN: 978-1-7281-1002-8

Proceedings of 2019 IEEE East-West Design & Test Symposium (EWDTS)

Batumi, Georgia, September 13 – 16, 2019

2019 IEEE EWDTS

Рисунок Б.15 – Тези доповіді конференції «The Software Platform for Evaluation of Effectiveness of Network Systems Analysis Technologies»



Proceedings of 2019 IEEE East-West Design & Test Symposium (EWDTS)

IEEE Catalog Numbers:
USB: CFP19DTW-USB
ISBN: 978-1-7281-1002-8

IEEE Conference Operations
445 Hoes Lane
Piscataway, NJ 08854 USA
Fax: +1 732 981 1769
Email: conference-ops@ieee.org
IEEE: Advancing Technology for Humanity

2019 IEEE EWDTS

Рисунок Б.16 – Тези доповіді конференції «The Software Platform for Evaluation of Effectiveness of Network Systems Analysis Technologies»

Рисунок Б.17 – Тези доповіді конференції «The Software Platform for Evaluation of Effectiveness of Network Systems Analysis Technologies»

The Software Platform for Evaluation of Effectiveness of Network Systems Analysis Technologies

Olha Ponomarenko
Department of Electronic Computers
Kharkiv National University of Radio Electronics
Kharkiv, Ukraine
olha.ponomarenko@nure.ua

Valeriy Gorbachov
Department of Electronic Computers
Kharkiv National University of Radio Electronics
Kharkiv, Ukraine
valeriy.gorbachov@nure.ua

Abdulrahman Kataeba Batiaa
Department of Electronic Computers
Kharkiv National University of Radio Electronics
Kharkiv, Ukraine
kotaeba04@gmail.com

Oksana Kotkova
Department of Electronic Computers
Kharkiv National University of Radio Electronics
Kharkiv, Ukraine
oksana.kotkova@nure.ua

Abstract—**Networks are used as a common model of a wide variety of complex systems, including social, biological, information, and technological domains. Nodes represent components of the system and links indicate interactions between them. Network monitoring, fast decision making and modeling techniques are fundamental to topology research of network systems. Topological analysis of networks is needed to develop network planning and network management, bottleneck and failure detection algorithms, system performance evaluation. The main objective of this paper is to research on network topology and use a software platform to evaluate the effectiveness of topology formal transformations for reducing the system dimension. The platform includes the set of modules: evaluation of topological network parameters, equivalent topological transformations, maximum flow searching, and topology generator. The experiment allowed to evaluate the effectiveness of both the network topology formal transformations and the effectiveness of the platform itself.**

Keywords—large scale system, topological transformations, software platform, topology generator, maximum flow problem, UML diagram

I. INTRODUCTION

Over the past decade, representations and studies of complex systems have been associated with so-called complex networks, which are network-based representations of complex systems. Complex systems are systems in which the pattern of interactions between a system's constituent parts is itself complex and is evolving together with the system's dynamics. In the context of network theory, a complex network is a network (graph) with non-trivial topological features and with a multitude of non-trivial statistical challenges [1], [2].

It is not uncommon now to see networks with millions or even billions of vertices. An increase in the size of networks leads to the development of new analytical approaches for their presentation and performance evaluation. When developing such approaches, researchers face various kinds of problems.

For networks consisting of even several dozens of vertices, it is quite simple to draw a picture of the network and answer specific questions about the structure of the network by studying this picture. This has been one of the primary ways to gain an understanding of network structure. Nowadays, a variety of great visualization tools are available, which helps to structure and to visualize the networks [3]. However, they are useless for an analysis of networks consisting of a million or a billion vertices.

In recent years, complex network theory becomes more and more popular. This theory is based on a solid mathematical framework that aims to solve a range of complex problems.

1. Development of an appropriate structural description of a complex system to determine the elements, subsystems and connections among them.

2. Development of approaches of determination and prediction of statistical properties, that characterize the structure and behavior of networked systems. For example, the definition of strongly connected components, shortest paths, cycles, races, etc. In addition, the system structure model is used to analyze the quality metrics of the structure. For example, clustering problems, network correlations.

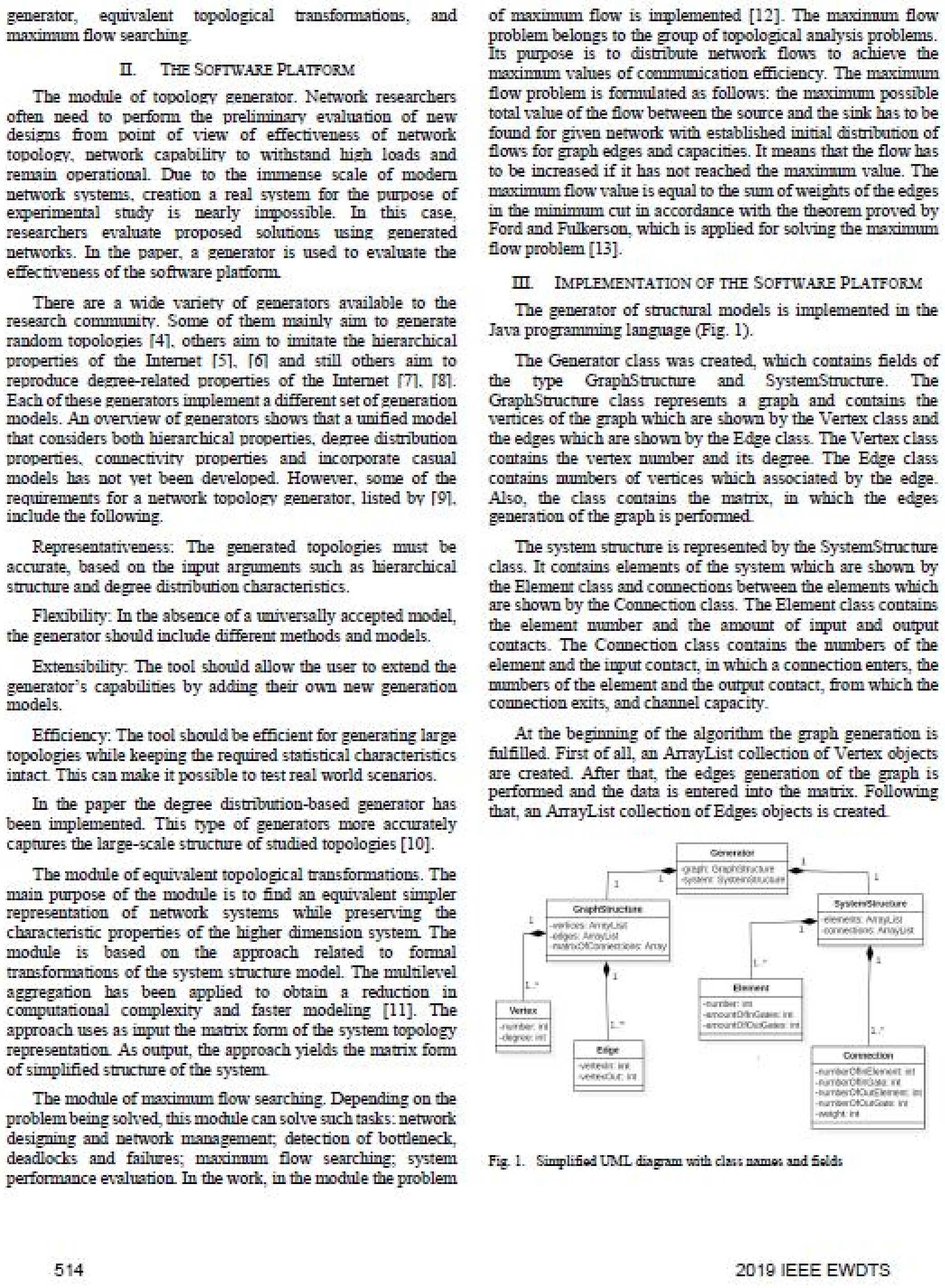
3. Implementation of aggregation and decomposition technologies to reduce the dimension of the system, when the time of performance evaluation plays an important role.

4. Optimal structural design. Most of problems of this group are the problems of increased complexity, such as, evaluation of the effects of structure on system behavior, equivalent transformations of the topological structure reducing the dimension of system, redistribution of links of the established structure, bottlenecks detecting.

Network systems require specific methods for analysis and design. The main objective of this paper is to research on network topology and use a software platform to evaluate the effectiveness of various systems analysis technologies including network dimensions reduction technology. The software platform, considered in the paper, includes the set of modules: evaluation of topological network parameters, topology

978-1-7281-1003-5/19/$31.00 ©2019 IEEE

Рисунок Б.18 – Тези доповіді конференції «The Software Platform for Evaluation of Effectiveness of Network Systems Analysis Technologies»



generator, equivalent topological transformations, and maximum flow searching.

II. THE SOFTWARE PLATFORM

The module of topology generator. Network researchers often need to perform the preliminary evaluation of new designs from point of view of effectiveness of network topology, network capability to withstand high loads and remain operational. Due to the immense scale of modern network systems, creation a real system for the purpose of experimental study is nearly impossible. In this case, researchers evaluate proposed solutions using generated networks. In the paper, a generator is used to evaluate the effectiveness of the software platform.

There are a wide variety of generators available to the research community. Some of them mainly aim to generate random topologies [4], others aim to imitate the hierarchical properties of the Internet [5], [6] and still others aim to reproduce degree-related properties of the Internet [7], [8]. Each of these generators implement a different set of generation models. An overview of generators shows that a unified model that considers both hierarchical properties, degree distribution properties, connectivity properties and incorporate casual models has not yet been developed. However, some of the requirements for a network topology generator, listed by [9], include the following.

Representativeness: The generated topologies must be accurate, based on the input arguments such as hierarchical structure and degree distribution characteristics.

Flexibility: In the absence of a universally accepted model, the generator should include different methods and models.

Extensibility: The tool should allow the user to extend the generator's capabilities by adding their own new generation models.

Efficiency: The tool should be efficient for generating large topologies while keeping the required statistical characteristics intact. This can make it possible to test real world scenarios.

In the paper the degree distribution-based generator has been implemented. This type of generators more accurately captures the large-scale structure of studied topologies [10].

The module of equivalent topological transformations. The main purpose of the module is to find an equivalent simpler representation of network systems while preserving the characteristic properties of the higher dimension system. The module is based on the approach related to formal transformations of the system structure model. The multilevel aggregation has been applied to obtain a reduction in computational complexity and faster modeling [11]. The approach uses as input the matrix form of the system topology representation. As output, the approach yields the matrix form of simplified structure of the system.

The module of maximum flow searching. Depending on the problem being solved, this module can solve such tasks: network designing and network management; detection of bottleneck, deadlocks and failures; maximum flow searching; system performance evaluation. In the work, in the module the problem of maximum flow is implemented [12]. The maximum flow problem belongs to the group of topological analysis problems. Its purpose is to distribute network flows to achieve the maximum values of communication efficiency. The maximum flow problem is formulated as follows: the maximum possible total value of the flow between the source and the sink has to be found for given network with established initial distribution of flows for graph edges and capacities. It means that the flow has to be increased if it has not reached the maximum value. The maximum flow value is equal to the sum of weights of the edges in the minimum cut in accordance with the theorem proved by Ford and Fulkerson, which is applied for solving the maximum flow problem [13].

III. IMPLEMENTATION OF THE SOFTWARE PLATFORM

The generator of structural models is implemented in the Java programming language (Fig. 1).

The Generator class was created, which contains fields of the type GraphStructure and SystemStructure. The GraphStructure class represents a graph and contains the vertices of the graph which are shown by the Vertex class and the edges which are shown by the Edge class. The Vertex class contains the vertex number and its degree. The Edge class contains numbers of vertices which associated by the edge. Also, the class contains the matrix, in which the edges generation of the graph is performed.

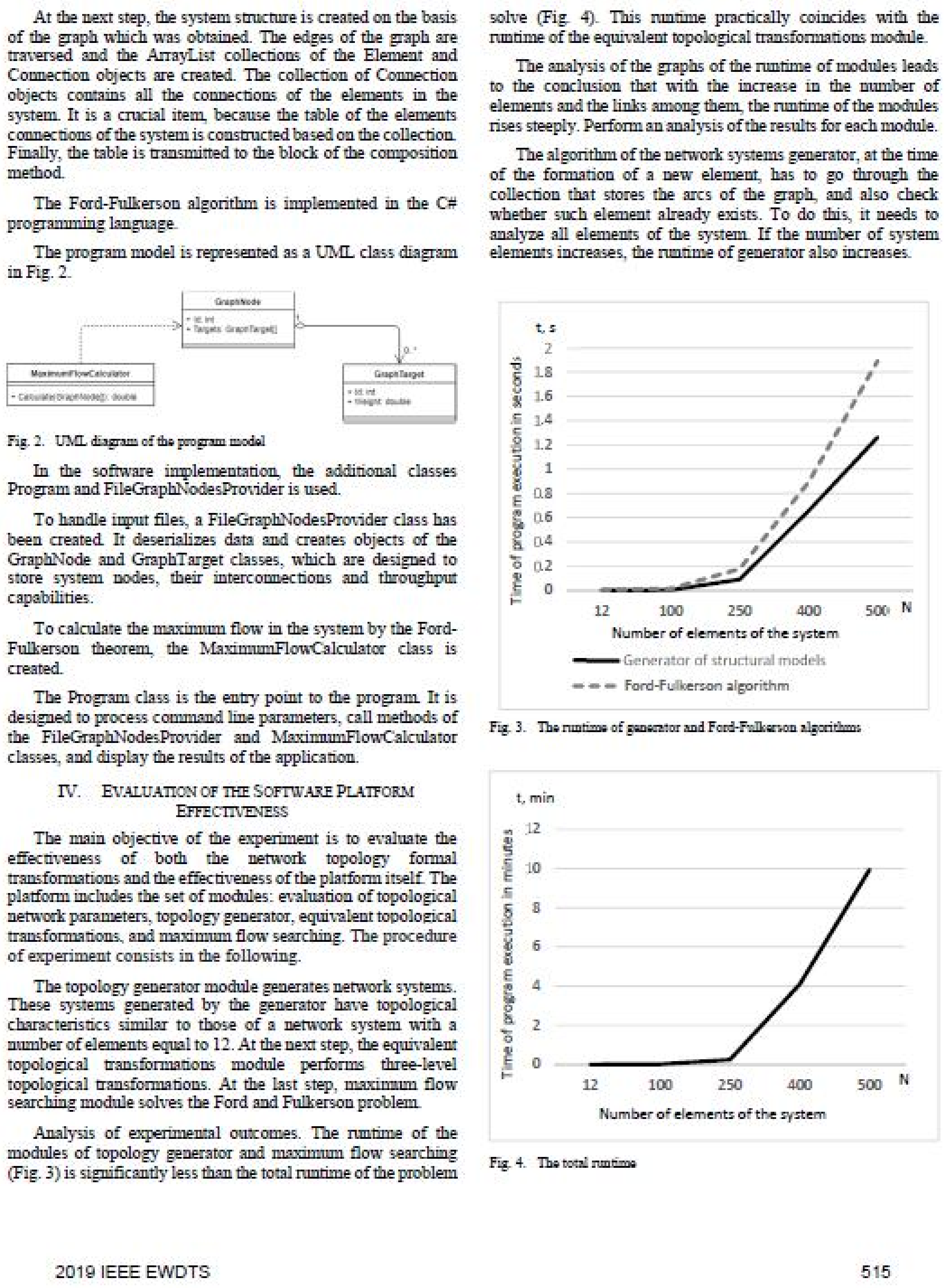
The system structure is represented by the SystemStructure class. It contains elements of the system which are shown by the Element class and connections between the elements which are shown by the Connection class. The Element class contains the element number and the amount of input and output contacts. The Connection class contains the numbers of the element and the input contact, in which a connection enters, the numbers of the element and the output contact, from which the connection exits, and channel capacity.

At the beginning of the algorithm the graph generation is fulfilled. First of all, an ArrayList collection of Vertex objects are created. After that, the edges generation of the graph is performed and the data is entered into the matrix. Following that, an ArrayList collection of Edges objects is created.

Fig. 1. Simplified UML diagram with class names and fields

514 2019 IEEE EWDTS

Рисунок Б.19 – Тези доповіді конференції «The Software Platform for Evaluation of Effectiveness of Network Systems Analysis Technologies»



At the next step, the system structure is created on the basis of the graph which was obtained. The edges of the graph are traversed and the ArrayList collections of the Element and Connection objects are created. The collection of Connection objects contains all the connections of the elements in the system. It is a crucial item, because the table of the elements connections of the system is constructed based on the collection. Finally, the table is transmitted to the block of the composition method.

The Ford-Fulkerson algorithm is implemented in the C# programming language.

The program model is represented as a UML class diagram in Fig. 2.

Fig. 2. UML diagram of the program model

In the software implementation, the additional classes Program and FileGraphNodesProvider is used.

To handle input files, a FileGraphNodesProvider class has been created. It deserializes data and creates objects of the GraphNode and GraphTarget classes, which are designed to store system nodes, their interconnections and throughput capabilities.

To calculate the maximum flow in the system by the Ford-Fulkerson theorem, the MaximumFlowCalculator class is created.

The Program class is the entry point to the program. It is designed to process command line parameters, call methods of the FileGraphNodesProvider and MaximumFlowCalculator classes, and display the results of the application.

IV. EVALUATION OF THE SOFTWARE PLATFORM EFFECTIVENESS

The main objective of the experiment is to evaluate the effectiveness of both the network topology formal transformations and the effectiveness of the platform itself. The platform includes the set of modules: evaluation of topological network parameters, topology generator, equivalent topological transformations, and maximum flow searching. The procedure of experiment consists in the following.

The topology generator module generates network systems. These systems generated by the generator have topological characteristics similar to those of a network system with a number of elements equal to 12. At the next step, the equivalent topological transformations module performs three-level topological transformations. At the last step, maximum flow searching module solves the Ford and Fulkerson problem.

Analysis of experimental outcomes. The runtime of the modules of topology generator and maximum flow searching (Fig. 3) is significantly less than the total runtime of the problem solve (Fig. 4). This runtime practically coincides with the runtime of the equivalent topological transformations module.

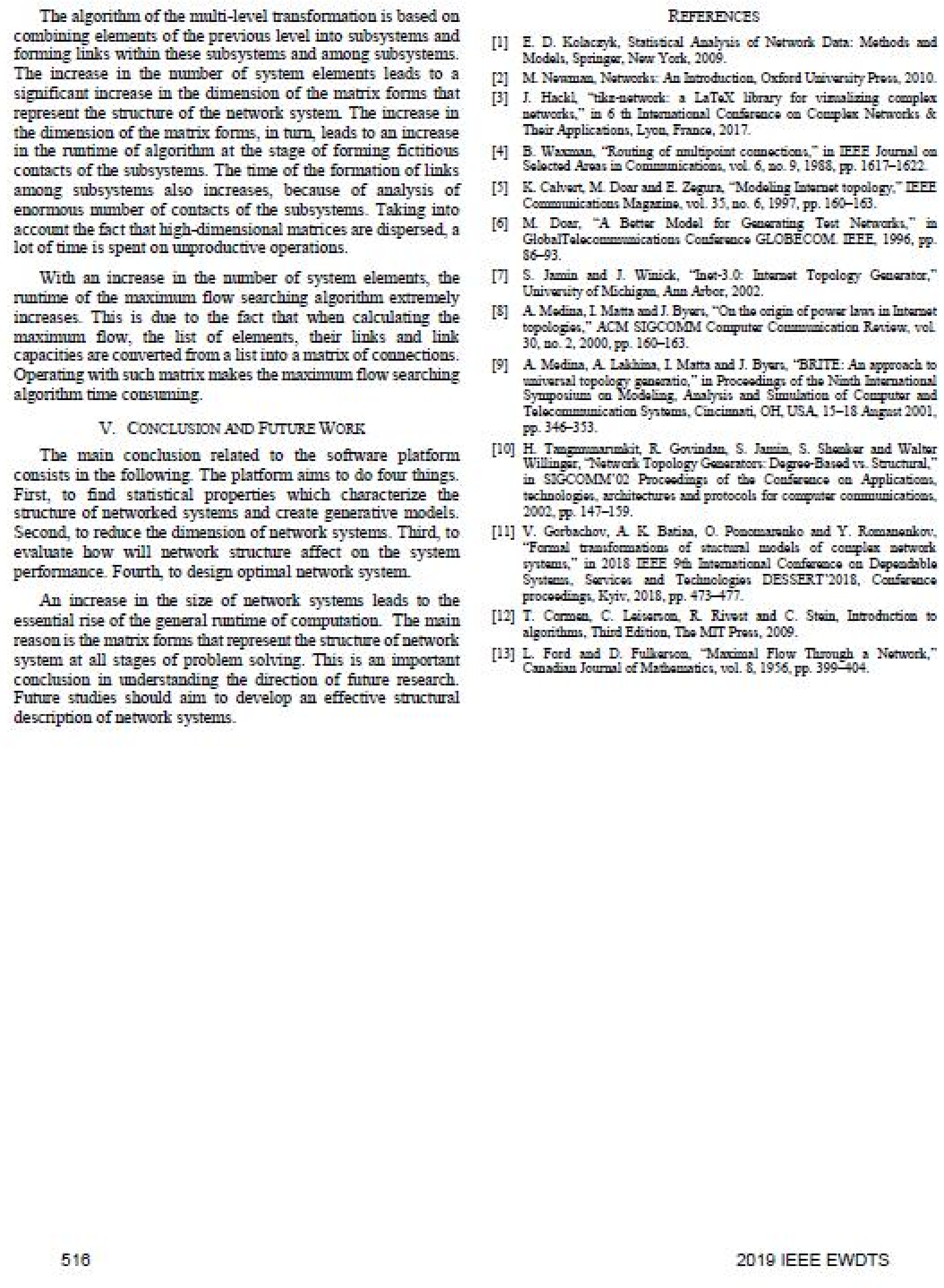
The analysis of the graphs of the runtime of modules leads to the conclusion that with the increase in the number of elements and the links among them, the runtime of the modules rises steeply. Perform an analysis of the results for each module.

The algorithm of the network systems generator, at the time of the formation of a new element, has to go through the collection that stores the arcs of the graph, and also check whether such element already exists. To do this, it needs to analyze all elements of the system. If the number of system elements increases, the runtime of generator also increases.

Fig. 3. The runtime of generator and Ford-Fulkerson algorithms

Fig. 4. The total runtime

Рисунок Б.20 – Тези доповіді конференції «The Software Platform for Evaluation of Effectiveness of Network Systems Analysis Technologies»



The algorithm of the multi-level transformation is based on combining elements of the previous level into subsystems and forming links within these subsystems and among subsystems. The increase in the number of system elements leads to a significant increase in the dimension of the matrix forms that represent the structure of the network system. The increase in the dimension of the matrix forms, in turn, leads to an increase in the runtime of algorithm at the stage of forming fictitious contacts of the subsystems. The time of the formation of links among subsystems also increases, because of analysis of enormous number of contacts of the subsystems. Taking into account the fact that high-dimensional matrices are dispersed, a lot of time is spent on unproductive operations.

With an increase in the number of system elements, the runtime of the maximum flow searching algorithm extremely increases. This is due to the fact that when calculating the maximum flow, the list of elements, their links and link capacities are converted from a list into a matrix of connections. Operating with such matrix makes the maximum flow searching algorithm time consuming.

V. CONCLUSION AND FUTURE WORK

The main conclusion related to the software platform consists in the following. The platform aims to do four things. First, to find statistical properties which characterize the structure of networked systems and create generative models. Second, to reduce the dimension of network systems. Third, to evaluate how will network structure affect on the system performance. Fourth, to design optimal network system.

An increase in the size of network systems leads to the essential rise of the general runtime of computation. The main reason is the matrix forms that represent the structure of network system at all stages of problem solving. This is an important conclusion in understanding the direction of future research. Future studies should aim to develop an effective structural description of network systems.

REFERENCES

[1] E. D. Kolaczyk, Statistical Analysis of Network Data: Methods and Models, Springer, New York, 2009.

[2] M. Newman, Networks: An Introduction, Oxford University Press, 2010.

[3] J. Hackl, "tikz-network: a LaTeX library for visualizing complex networks," in 6 th International Conference on Complex Networks & Their Applications, Lyon, France, 2017.

[4] B. Waxman, "Routing of multipoint connections," in IEEE Journal on Selected Areas in Communications, vol. 6, no. 9, 1988, pp. 1617–1622.

[5] K. Calvert, M. Doar and E. Zegura, "Modeling Internet topology," IEEE Communications Magazine, vol. 35, no. 6, 1997, pp. 160–163.

[6] M. Doar, "A Better Model for Generating Test Networks," in GlobalTelecommunications Conference GLOBECOM. IEEE, 1996, pp. 86–93.

[7] S. Jamin and J. Winick, "Inet-3.0: Internet Topology Generator," University of Michigan, Ann Arbor, 2002.

[8] A. Medina, I. Matta and J. Byers, "On the origin of power laws in Internet topologies," ACM SIGCOMM Computer Communication Review, vol. 30, no. 2, 2000, pp. 160–163.

[9] A. Medina, A. Lakhina, I. Matta and J. Byers, "BRITE: An approach to universal topology generatio," in Proceedings of the Ninth International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, Cincinnati, OH, USA, 15–18 August 2001, pp. 346–353.

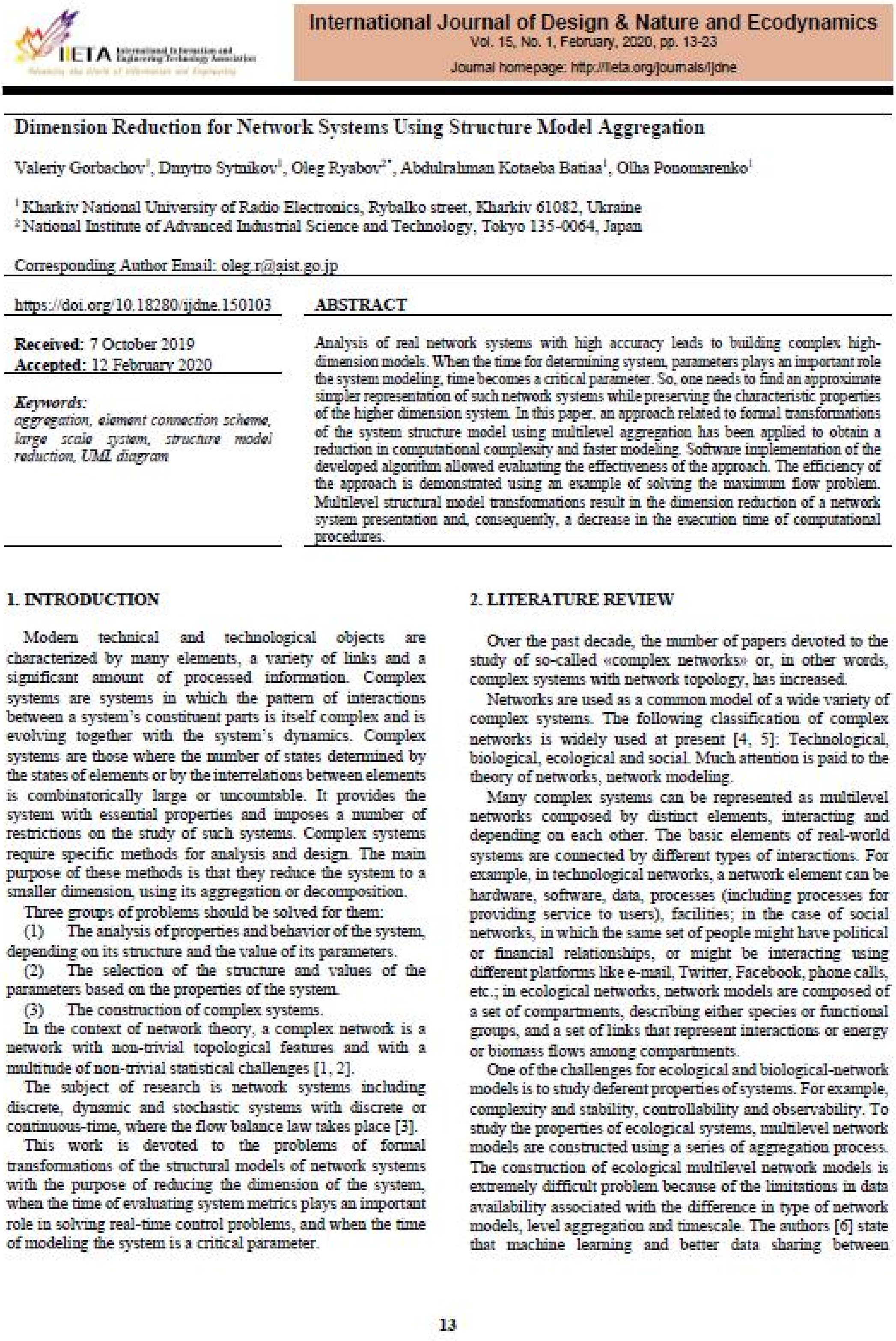
[10] H. Tangmunarunkit, R. Govindan, S. Jamin, S. Shenker and Walter Willinger, "Network Topology Generators: Degree-Based vs. Structural," in SIGCOMM'02 Proceedings of the Conference on Applications, technologies, architectures and protocols for computer communications, 2002, pp. 147–159.

[11] V. Gorbachov, A. K. Batiaa, O. Ponomarenko and Y. Romanenkov, "Formal transformations of stuctural models of complex network systems," in 2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies DESSERT'2018, Conference proceedings, Kyiv, 2018, pp. 473–477.

[12] T. Cormen, C. Leiserson, R. Rivest and C. Stein, Introduction to algorithms, Third Edition, The MIT Press, 2009.

[13] L. Ford and D. Fulkerson, "Maximal Flow Through a Network," Canadian Journal of Mathematics, vol. 8, 1956, pp. 399–404.

Рисунок Б.21 – Тези доповіді конференції «The Software Platform for Evaluation of Effectiveness of Network Systems Analysis Technologies»



International Journal of Design & Nature and Ecodynamics
Vol. 15, No. 1, February, 2020, pp. 13-23
Journal homepage: http://iieta.org/journals/ijdne

Dimension Reduction for Network Systems Using Structure Model Aggregation

Valeriy Gorbachov[1], Dmytro Sytnikov[1], Oleg Ryabov[2*], Abdulrahman Kotaeba Batiaa[1], Olha Ponomarenko[1]

[1] Kharkiv National University of Radio Electronics, Rybalko street, Kharkiv 61082, Ukraine
[2] National Institute of Advanced Industrial Science and Technology, Tokyo 135-0064, Japan

Corresponding Author Email: oleg.r@aist.go.jp

https://doi.org/10.18280/ijdne.150103

Received: 7 October 2019
Accepted: 12 February 2020

Keywords:
aggregation, element connection scheme, large scale system, structure model reduction, UML diagram

ABSTRACT

Analysis of real network systems with high accuracy leads to building complex high-dimension models. When the time for determining system, parameters plays an important role the system modeling, time becomes a critical parameter. So, one needs to find an approximate simpler representation of such network systems while preserving the characteristic properties of the higher dimension system. In this paper, an approach related to formal transformations of the system structure model using multilevel aggregation has been applied to obtain a reduction in computational complexity and faster modeling. Software implementation of the developed algorithm allowed evaluating the effectiveness of the approach. The efficiency of the approach is demonstrated using an example of solving the maximum flow problem. Multilevel structural model transformations result in the dimension reduction of a network system presentation and, consequently, a decrease in the execution time of computational procedures.

1. INTRODUCTION

Modern technical and technological objects are characterized by many elements, a variety of links and a significant amount of processed information. Complex systems are systems in which the pattern of interactions between a system's constituent parts is itself complex and is evolving together with the system's dynamics. Complex systems are those where the number of states determined by the states of elements or by the interrelations between elements is combinatorically large or uncountable. It provides the system with essential properties and imposes a number of restrictions on the study of such systems. Complex systems require specific methods for analysis and design. The main purpose of these methods is that they reduce the system to a smaller dimension, using its aggregation or decomposition.

Three groups of problems should be solved for them:

(1) The analysis of properties and behavior of the system, depending on its structure and the value of its parameters.

(2) The selection of the structure and values of the parameters based on the properties of the system.

(3) The construction of complex systems.

In the context of network theory, a complex network is a network with non-trivial topological features and with a multitude of non-trivial statistical challenges [1, 2].

The subject of research is network systems including discrete, dynamic and stochastic systems with discrete or continuous-time, where the flow balance law takes place [3].

This work is devoted to the problems of formal transformations of the structural models of network systems with the purpose of reducing the dimension of the system, when the time of evaluating system metrics plays an important role in solving real-time control problems, and when the time of modeling the system is a critical parameter.

2. LITERATURE REVIEW

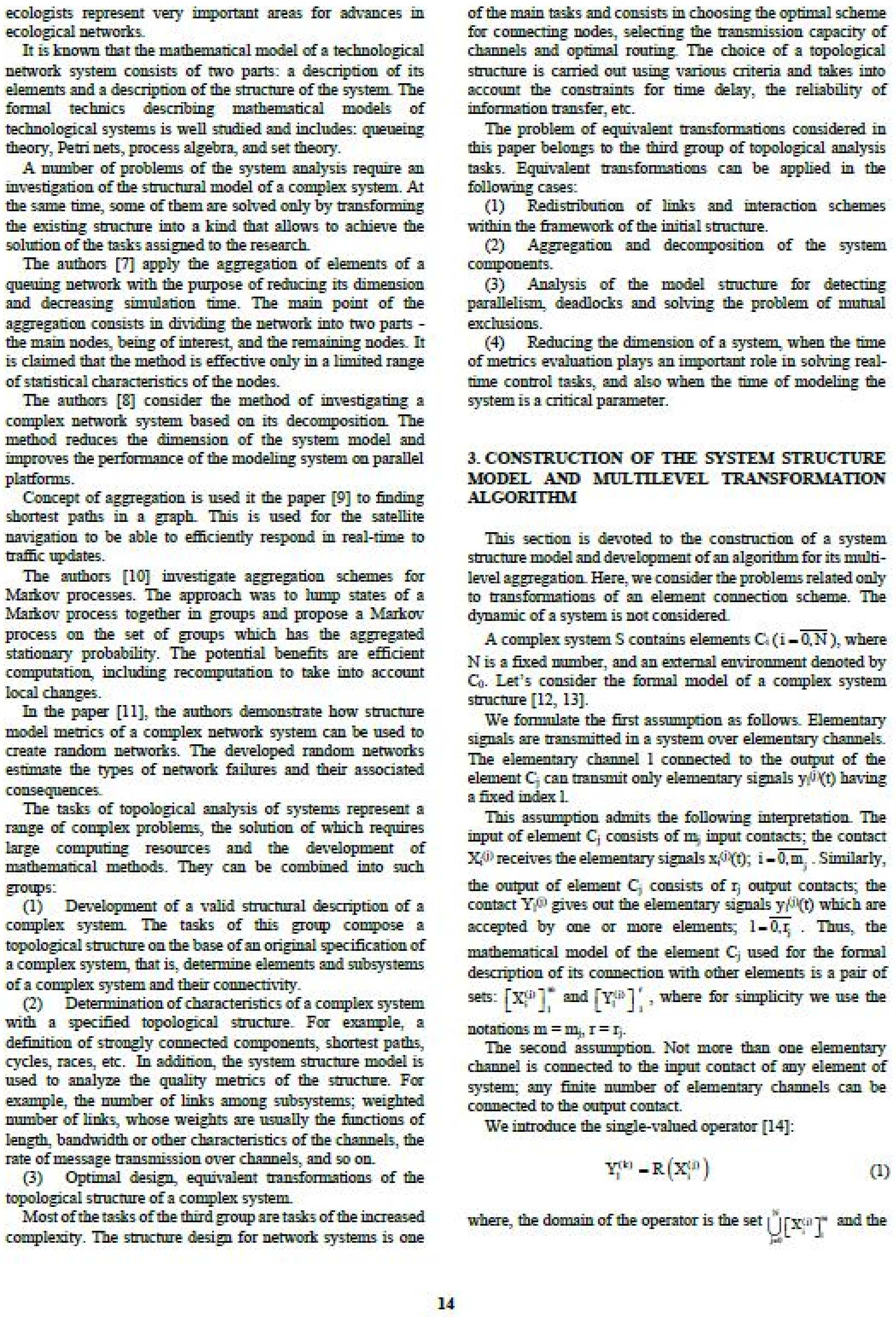
Over the past decade, the number of papers devoted to the study of so-called «complex networks» or, in other words, complex systems with network topology, has increased.

Networks are used as a common model of a wide variety of complex systems. The following classification of complex networks is widely used at present [4, 5]: Technological, biological, ecological and social. Much attention is paid to the theory of networks, network modeling.

Many complex systems can be represented as multilevel networks composed by distinct elements, interacting and depending on each other. The basic elements of real-world systems are connected by different types of interactions. For example, in technological networks, a network element can be hardware, software, data, processes (including processes for providing service to users), facilities; in the case of social networks, in which the same set of people might have political or financial relationships, or might be interacting using different platforms like e-mail, Twitter, Facebook, phone calls, etc.; in ecological networks, network models are composed of a set of compartments, describing either species or functional groups, and a set of links that represent interactions or energy or biomass flows among compartments.

One of the challenges for ecological and biological-network models is to study deferent properties of systems. For example, complexity and stability, controllability and observability. To study the properties of ecological systems, multilevel network models are constructed using a series of aggregation process. The construction of ecological multilevel network models is extremely difficult problem because of the limitations in data availability associated with the difference in type of network models, level aggregation and timescale. The authors [6] state that machine learning and better data sharing between

Рисунок Б.22 – Стаття «Dimension Reduction for Network Systems Using Structure Model Aggregation»



ecologists represent very important areas for advances in ecological networks.

It is known that the mathematical model of a technological network system consists of two parts: a description of its elements and a description of the structure of the system. The formal technics describing mathematical models of technological systems is well studied and includes: queueing theory, Petri nets, process algebra, and set theory.

A number of problems of the system analysis require an investigation of the structural model of a complex system. At the same time, some of them are solved only by transforming the existing structure into a kind that allows to achieve the solution of the tasks assigned to the research.

The authors [7] apply the aggregation of elements of a queuing network with the purpose of reducing its dimension and decreasing simulation time. The main point of the aggregation consists in dividing the network into two parts - the main nodes, being of interest, and the remaining nodes. It is claimed that the method is effective only in a limited range of statistical characteristics of the nodes.

The authors [8] consider the method of investigating a complex network system based on its decomposition. The method reduces the dimension of the system model and improves the performance of the modeling system on parallel platforms.

Concept of aggregation is used it the paper [9] to finding shortest paths in a graph. This is used for the satellite navigation to be able to efficiently respond in real-time to traffic updates.

The authors [10] investigate aggregation schemes for Markov processes. The approach was to lump states of a Markov process together in groups and propose a Markov process on the set of groups which has the aggregated stationary probability. The potential benefits are efficient computation, including recomputation to take into account local changes.

In the paper [11], the authors demonstrate how structure model metrics of a complex network system can be used to create random networks. The developed random networks estimate the types of network failures and their associated consequences.

The tasks of topological analysis of systems represent a range of complex problems, the solution of which requires large computing resources and the development of mathematical methods. They can be combined into such groups:

(1) Development of a valid structural description of a complex system. The tasks of this group compose a topological structure on the base of an original specification of a complex system, that is, determine elements and subsystems of a complex system and their connectivity.

(2) Determination of characteristics of a complex system with a specified topological structure. For example, a definition of strongly connected components, shortest paths, cycles, races, etc. In addition, the system structure model is used to analyze the quality metrics of the structure. For example, the number of links among subsystems; weighted number of links, whose weights are usually the functions of length, bandwidth or other characteristics of the channels, the rate of message transmission over channels, and so on.

(3) Optimal design, equivalent transformations of the topological structure of a complex system.

Most of the tasks of the third group are tasks of the increased complexity. The structure design for network systems is one of the main tasks and consists in choosing the optimal scheme for connecting nodes, selecting the transmission capacity of channels and optimal routing. The choice of a topological structure is carried out using various criteria and takes into account the constraints for time delay, the reliability of information transfer, etc.

The problem of equivalent transformations considered in this paper belongs to the third group of topological analysis tasks. Equivalent transformations can be applied in the following cases:

(1) Redistribution of links and interaction schemes within the framework of the initial structure.

(2) Aggregation and decomposition of the system components.

(3) Analysis of the model structure for detecting parallelism, deadlocks and solving the problem of mutual exclusions.

(4) Reducing the dimension of a system, when the time of metrics evaluation plays an important role in solving real-time control tasks, and also when the time of modeling the system is a critical parameter.

3. CONSTRUCTION OF THE SYSTEM STRUCTURE MODEL AND MULTILEVEL TRANSFORMATION ALGORITHM

This section is devoted to the construction of a system structure model and development of an algorithm for its multi-level aggregation. Here, we consider the problems related only to transformations of an element connection scheme. The dynamic of a system is not considered.

A complex system S contains elements C_i ($i = \overline{0,N}$), where N is a fixed number, and an external environment denoted by C_0. Let's consider the formal model of a complex system structure [12, 13].

We formulate the first assumption as follows. Elementary signals are transmitted in a system over elementary channels. The elementary channel l connected to the output of the element C_j can transmit only elementary signals $y_l^{(j)}(t)$ having a fixed index l.

This assumption admits the following interpretation. The input of element C_j consists of m_j input contacts; the contact $X_i^{(j)}$ receives the elementary signals $x_i^{(j)}(t)$; $i = \overline{0,m_j}$. Similarly, the output of element C_j consists of r_j output contacts; the contact $Y_l^{(j)}$ gives out the elementary signals $y_l^{(j)}(t)$ which are accepted by one or more elements; $l = \overline{0,r_j}$. Thus, the mathematical model of the element C_j used for the formal description of its connection with other elements is a pair of sets: $\left[X_i^{(j)}\right]_1^m$ and $\left[Y_l^{(j)}\right]_1^r$, where for simplicity we use the notations $m = m_j$, $r = r_j$.

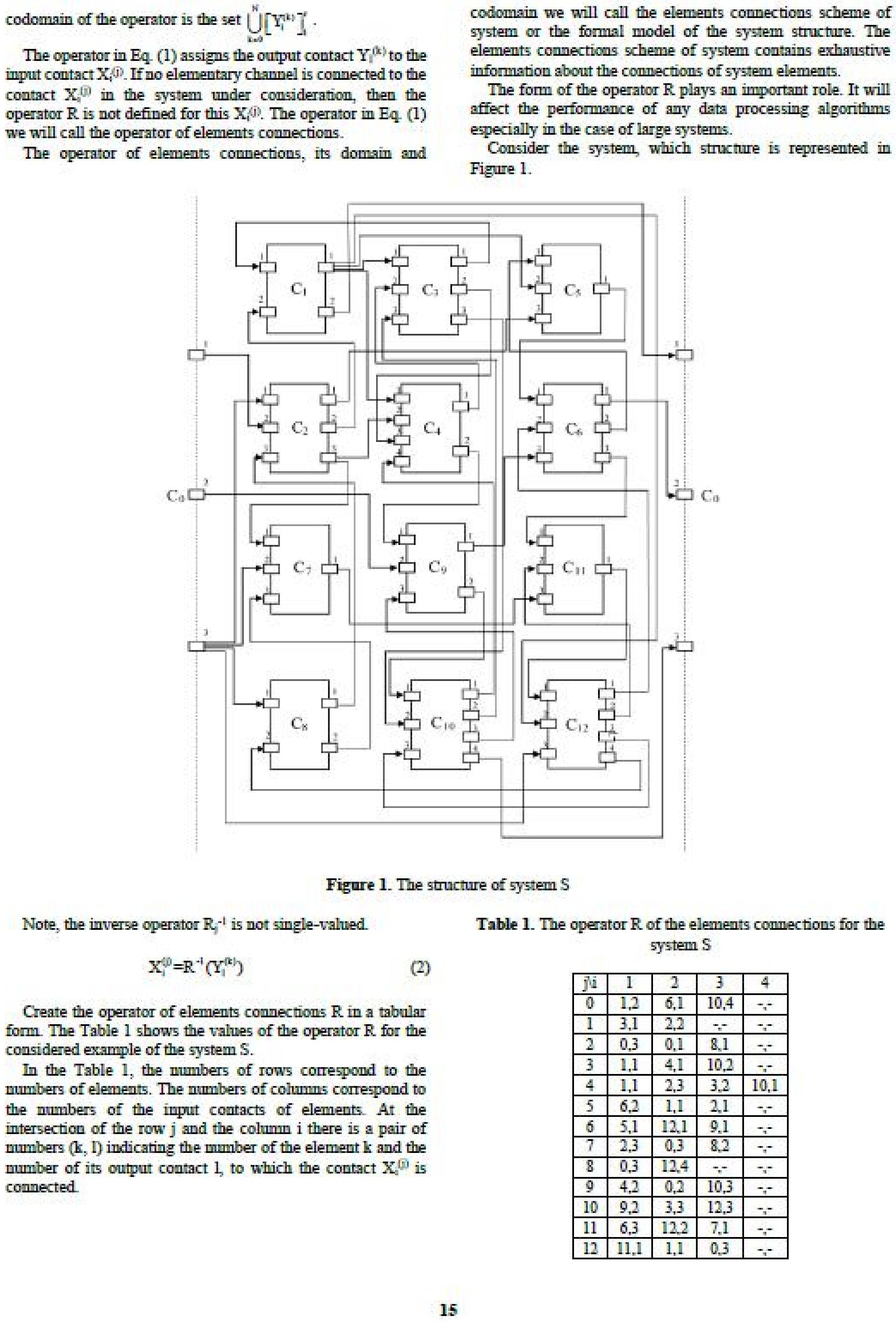
The second assumption. Not more than one elementary channel is connected to the input contact of any element of system; any finite number of elementary channels can be connected to the output contact.

We introduce the single-valued operator [14]:

$$Y_l^{(k)} = R\left(X_i^{(j)}\right) \tag{1}$$

where, the domain of the operator is the set $\bigcup_{j=0}^{N}\left[X_i^{(j)}\right]_1^m$ and the

Рисунок Б.23 – Стаття «Dimension Reduction for Network Systems Using Structure Model Aggregation»



codomain of the operator is the set $\bigcup_{k=0}^{N}\left[Y_l^{(k)}\right]_l$.

The operator in Eq. (1) assigns the output contact $Y_l^{(k)}$ to the input contact $X_i^{(j)}$. If no elementary channel is connected to the contact $X_i^{(j)}$ in the system under consideration, then the operator R is not defined for this $X_i^{(j)}$. The operator in Eq. (1) we will call the operator of elements connections.

The operator of elements connections, its domain and codomain we will call the elements connections scheme of system or the formal model of the system structure. The elements connections scheme of system contains exhaustive information about the connections of system elements.

The form of the operator R plays an important role. It will affect the performance of any data processing algorithms especially in the case of large systems.

Consider the system, which structure is represented in Figure 1.

Figure 1. The structure of system S

Note, the inverse operator R_j^{-1} is not single-valued.

$$X_i^{(j)}=R^{-1}(Y_l^{(k)}) \qquad (2)$$

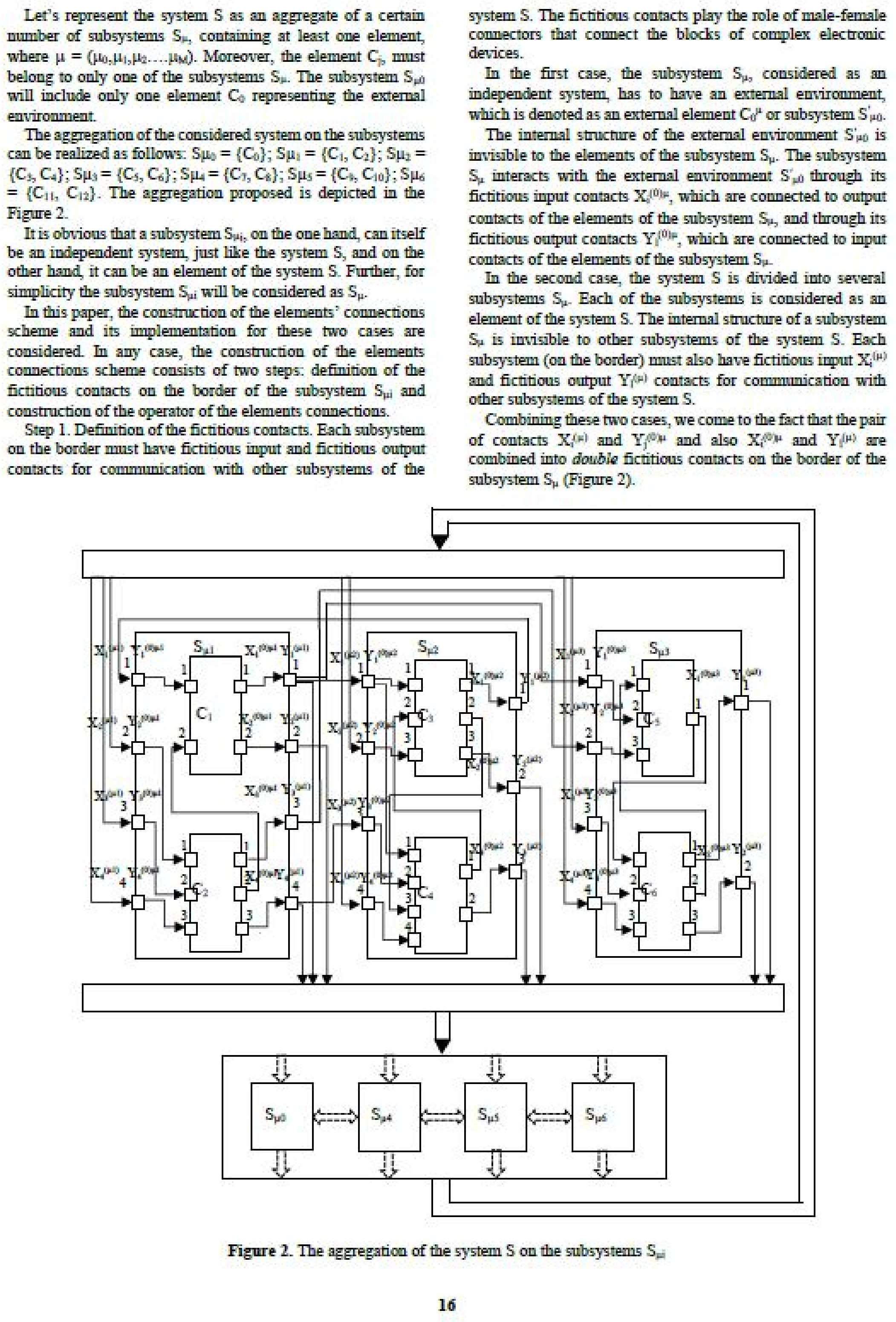
Create the operator of elements connections R in a tabular form. The Table 1 shows the values of the operator R for the considered example of the system S.

In the Table 1, the numbers of rows correspond to the numbers of elements. The numbers of columns correspond to the numbers of the input contacts of elements. At the intersection of the row j and the column i there is a pair of numbers (k, l) indicating the number of the element k and the number of its output contact l, to which the contact $X_i^{(j)}$ is connected.

Table 1. The operator R of the elements connections for the system S

j\i	1	2	3	4
0	1,2	6,1	10,4	-,-
1	3,1	2,2	-,-	-,-
2	0,3	0,1	8,1	-,-
3	1,1	4,1	10,2	-,-
4	1,1	2,3	3,2	10,1
5	6,2	1,1	2,1	-,-
6	5,1	12,1	9,1	-,-
7	2,3	0,3	8,2	-,-
8	0,3	12,4	-,-	-,-
9	4,2	0,2	10,3	-,-
10	9,2	3,3	12,3	-,-
11	6,3	12,2	7,1	-,-
12	11,1	1,1	0,3	-,-

Рисунок Б.24 – Стаття «Dimension Reduction for Network Systems Using Structure Model Aggregation»



Let's represent the system S as an aggregate of a certain number of subsystems S_μ, containing at least one element, where $\mu = (\mu_0, \mu_1, \mu_2 \ldots \mu_M)$. Moreover, the element C_j, must belong to only one of the subsystems S_μ. The subsystem $S_{\mu 0}$ will include only one element C_0 representing the external environment.

The aggregation of the considered system on the subsystems can be realized as follows: $S\mu_0 = \{C_0\}$; $S\mu_1 = \{C_1, C_2\}$; $S\mu_2 = \{C_3, C_4\}$; $S\mu_3 = \{C_5, C_6\}$; $S\mu_4 = \{C_7, C_8\}$; $S\mu_5 = \{C_9, C_{10}\}$; $S\mu_6 = \{C_{11}, C_{12}\}$. The aggregation proposed is depicted in the Figure 2.

It is obvious that a subsystem $S_{\mu i}$, on the one hand, can itself be an independent system, just like the system S, and on the other hand, it can be an element of the system S. Further, for simplicity the subsystem $S_{\mu i}$ will be considered as S_μ.

In this paper, the construction of the elements' connections scheme and its implementation for these two cases are considered. In any case, the construction of the elements connections scheme consists of two steps: definition of the fictitious contacts on the border of the subsystem $S_{\mu i}$ and construction of the operator of the elements connections.

Step 1. Definition of the fictitious contacts. Each subsystem on the border must have fictitious input and fictitious output contacts for communication with other subsystems of the system S. The fictitious contacts play the role of male-female connectors that connect the blocks of complex electronic devices.

In the first case, the subsystem S_μ, considered as an independent system, has to have an external environment, which is denoted as an external element $C_0{}^\mu$ or subsystem $S'_{\mu 0}$.

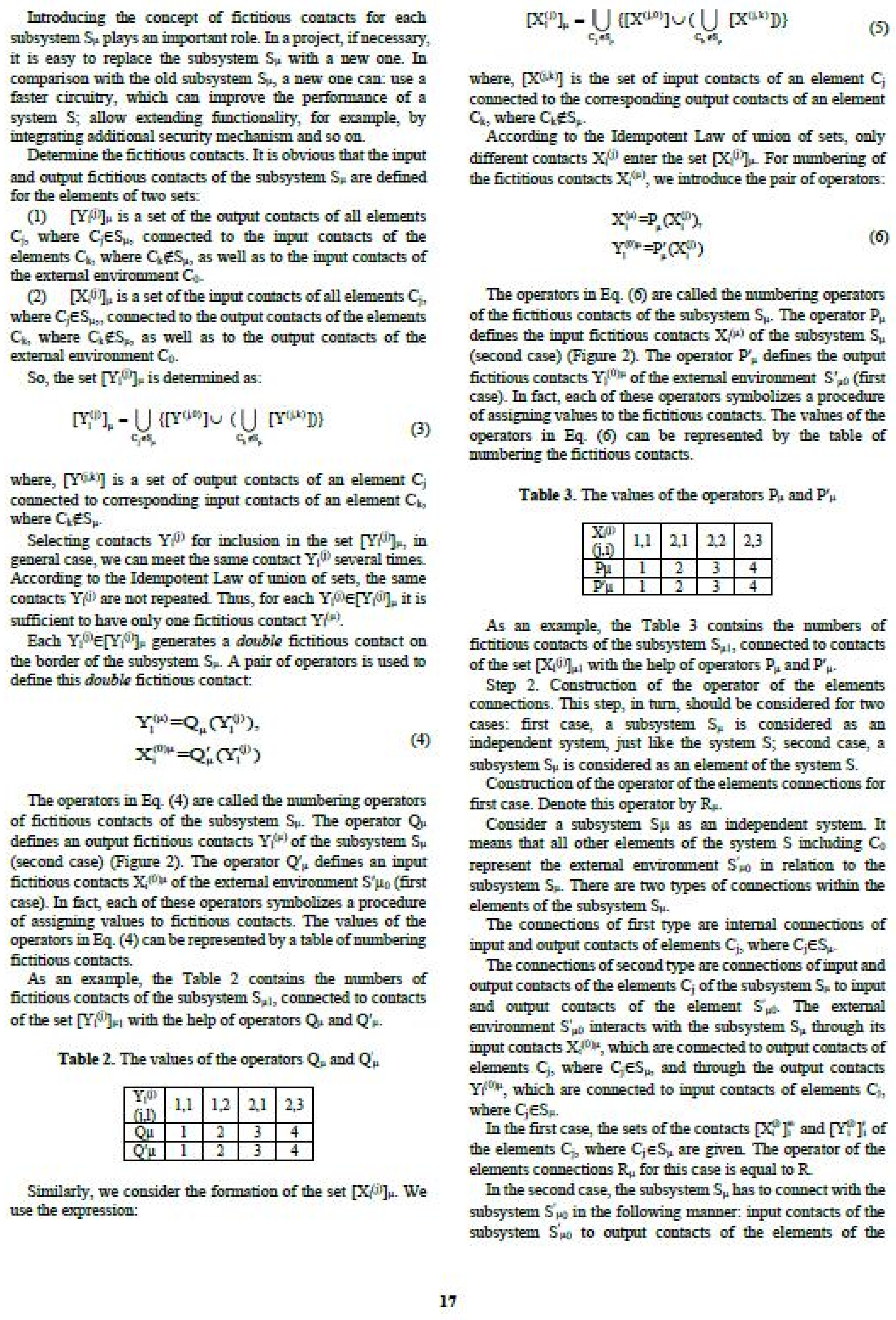
The internal structure of the external environment $S'_{\mu 0}$ is invisible to the elements of the subsystem S_μ. The subsystem S_μ interacts with the external environment $S'_{\mu 0}$ through its fictitious input contacts $X_i^{(0)\mu}$, which are connected to output contacts of the elements of the subsystem S_μ, and through its fictitious output contacts $Y_j^{(0)\mu}$, which are connected to input contacts of the elements of the subsystem S_μ.

In the second case, the system S is divided into several subsystems S_μ. Each of the subsystems is considered as an element of the system S. The internal structure of a subsystem S_μ is invisible to other subsystems of the system S. Each subsystem (on the border) must also have fictitious input $X_i^{(\mu)}$ and fictitious output $Y_j^{(\mu)}$ contacts for communication with other subsystems of the system S.

Combining these two cases, we come to the fact that the pair of contacts $X_i^{(\mu)}$ and $Y_j^{(0)\mu}$ and also $X_i^{(0)\mu}$ and $Y_j^{(\mu)}$ are combined into *double* fictitious contacts on the border of the subsystem S_μ (Figure 2).

Figure 2. The aggregation of the system S on the subsystems $S_{\mu i}$

Рисунок Б.25 – Стаття «Dimension Reduction for Network Systems Using Structure Model Aggregation»



Introducing the concept of fictitious contacts for each subsystem S_μ plays an important role. In a project, if necessary, it is easy to replace the subsystem S_μ with a new one. In comparison with the old subsystem S_μ, a new one can: use a faster circuitry, which can improve the performance of a system S; allow extending functionality, for example, by integrating additional security mechanism and so on.

Determine the fictitious contacts. It is obvious that the input and output fictitious contacts of the subsystem S_μ are defined for the elements of two sets:

(1) $[Y_l^{(j)}]_\mu$ is a set of the output contacts of all elements C_j, where $C_j \in S_\mu$, connected to the input contacts of the elements C_k, where $C_k \notin S_\mu$, as well as to the input contacts of the external environment C_0.

(2) $[X_i^{(j)}]_\mu$ is a set of the input contacts of all elements C_j, where $C_j \in S_\mu$,, connected to the output contacts of the elements C_k, where $C_k \notin S_\mu$, as well as to the output contacts of the external environment C_0.

So, the set $[Y_l^{(j)}]_\mu$ is determined as:

$$[Y_l^{(j)}]_\mu = \bigcup_{C_j \in S_\mu} \{[Y^{(j,0)}] \cup (\bigcup_{C_k \notin S_\mu} [Y^{(j,k)}])\} \tag{3}$$

where, $[Y^{(j,k)}]$ is a set of output contacts of an element C_j connected to corresponding input contacts of an element C_k, where $C_k \notin S_\mu$.

Selecting contacts $Y_l^{(j)}$ for inclusion in the set $[Y_l^{(j)}]_\mu$, in general case, we can meet the same contact $Y_l^{(j)}$ several times. According to the Idempotent Law of union of sets, the same contacts $Y_l^{(j)}$ are not repeated. Thus, for each $Y_l^{(j)} \in [Y_l^{(j)}]_\mu$ it is sufficient to have only one fictitious contact $Y_l^{(\mu)}$.

Each $Y_l^{(j)} \in [Y_l^{(j)}]_\mu$ generates a *double* fictitious contact on the border of the subsystem S_μ. A pair of operators is used to define this *double* fictitious contact:

$$\begin{aligned} Y_l^{(\mu)} &= Q_\mu(Y_l^{(j)}), \\ X_i^{(0)\mu} &= Q'_\mu(Y_l^{(j)}) \end{aligned} \tag{4}$$

The operators in Eq. (4) are called the numbering operators of fictitious contacts of the subsystem S_μ. The operator Q_μ defines an output fictitious contacts $Y_l^{(\mu)}$ of the subsystem S_μ (second case) (Figure 2). The operator Q'_μ defines an input fictitious contacts $X_i^{(0)\mu}$ of the external environment $S'_{\mu 0}$ (first case). In fact, each of these operators symbolizes a procedure of assigning values to fictitious contacts. The values of the operators in Eq. (4) can be represented by a table of numbering fictitious contacts.

As an example, the Table 2 contains the numbers of fictitious contacts of the subsystem $S_{\mu 1}$, connected to contacts of the set $[Y_l^{(j)}]_{\mu 1}$ with the help of operators Q_μ and Q'_μ.

Table 2. The values of the operators Q_μ and Q'_μ

$Y_l^{(j)}$ (j,l)	1,1	1,2	2,1	2,3
Qμ	1	2	3	4
Q′μ	1	2	3	4

Similarly, we consider the formation of the set $[X_i^{(j)}]_\mu$. We use the expression:

$$[X_i^{(j)}]_\mu = \bigcup_{C_j \in S_\mu} \{[X^{(j,0)}] \cup (\bigcup_{C_k \notin S_\mu} [X^{(j,k)}])\} \tag{5}$$

where, $[X^{(j,k)}]$ is the set of input contacts of an element C_j connected to the corresponding output contacts of an element C_k, where $C_k \notin S_\mu$.

According to the Idempotent Law of union of sets, only different contacts $X_i^{(j)}$ enter the set $[X_i^{(j)}]_\mu$. For numbering of the fictitious contacts $X_i^{(\mu)}$, we introduce the pair of operators:

$$\begin{aligned} X_i^{(\mu)} &= P_\mu(X_i^{(j)}), \\ Y_l^{(0)\mu} &= P'_\mu(X_i^{(j)}) \end{aligned} \tag{6}$$

The operators in Eq. (6) are called the numbering operators of the fictitious contacts of the subsystem S_μ. The operator P_μ defines the input fictitious contacts $X_i^{(\mu)}$ of the subsystem S_μ (second case) (Figure 2). The operator P'_μ defines the output fictitious contacts $Y_l^{(0)\mu}$ of the external environment $S'_{\mu 0}$ (first case). In fact, each of these operators symbolizes a procedure of assigning values to the fictitious contacts. The values of the operators in Eq. (6) can be represented by the table of numbering the fictitious contacts.

Table 3. The values of the operators P_μ and P'_μ

$X_i^{(j)}$ (j,i)	1,1	2,1	2,2	2,3
Pμ	1	2	3	4
P′μ	1	2	3	4

As an example, the Table 3 contains the numbers of fictitious contacts of the subsystem $S_{\mu 1}$, connected to contacts of the set $[X_i^{(j)}]_{\mu 1}$ with the help of operators P_μ and P'_μ.

Step 2. Construction of the operator of the elements connections. This step, in turn, should be considered for two cases: first case, a subsystem S_μ is considered as an independent system, just like the system S; second case, a subsystem S_μ is considered as an element of the system S.

Construction of the operator of the elements connections for first case. Denote this operator by R_μ.

Consider a subsystem Sμ as an independent system. It means that all other elements of the system S including C_0 represent the external environment $S'_{\mu 0}$ in relation to the subsystem S_μ. There are two types of connections within the elements of the subsystem S_μ.

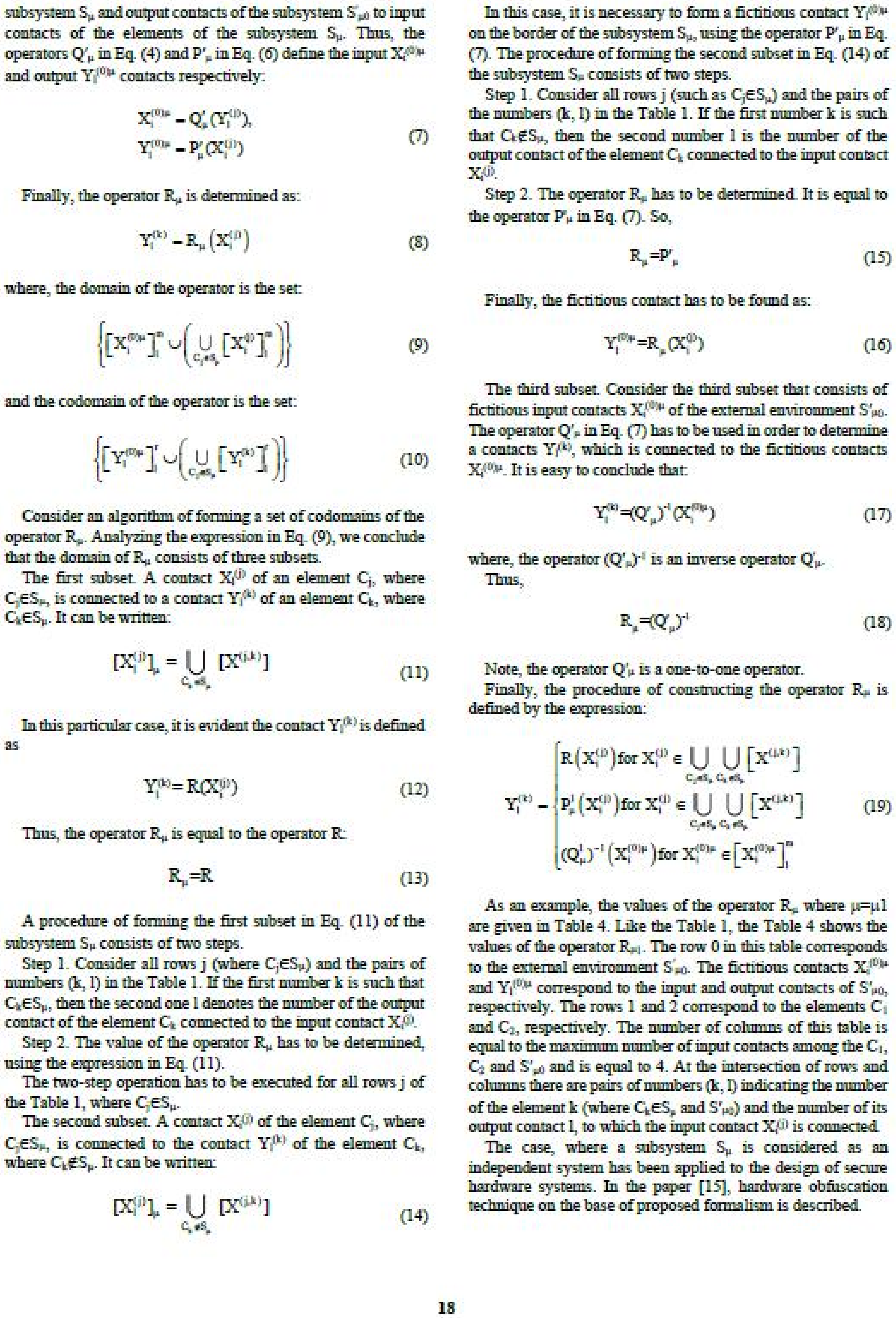
The connections of first type are internal connections of input and output contacts of elements C_j, where $C_j \in S_\mu$.

The connections of second type are connections of input and output contacts of the elements C_j of the subsystem S_μ to input and output contacts of the element $S'_{\mu 0}$. The external environment $S'_{\mu 0}$ interacts with the subsystem S_μ through its input contacts $X_i^{(0)\mu}$, which are connected to output contacts of elements C_j, where $C_j \in S_\mu$, and through the output contacts $Y_l^{(0)\mu}$, which are connected to input contacts of elements C_j, where $C_j \in S_\mu$.

In the first case, the sets of the contacts $[X_i^{(j)}]_i^{\mu}$ and $[Y_l^{(j)}]_l^{j}$ of the elements C_j, where $C_j \in S_\mu$ are given. The operator of the elements connections R_μ for this case is equal to R.

In the second case, the subsystem S_μ has to connect with the subsystem $S'_{\mu 0}$ in the following manner: input contacts of the subsystem $S'_{\mu 0}$ to output contacts of the elements of the

Рисунок Б.26 – Стаття «Dimension Reduction for Network Systems Using Structure Model Aggregation»



subsystem S_μ and output contacts of the subsystem $S'_{\mu 0}$ to input contacts of the elements of the subsystem S_μ. Thus, the operators Q'_μ in Eq. (4) and P'_μ in Eq. (6) define the input $X_l^{(0)\mu}$ and output $Y_l^{(0)\mu}$ contacts respectively:

$$\begin{aligned} X_l^{(0)\mu} &= Q'_\mu(Y_l^{(j)}), \\ Y_l^{(0)\mu} &= P'_\mu(X_l^{(j)}) \end{aligned} \tag{7}$$

Finally, the operator R_μ is determined as:

$$Y_l^{(k)} = R_\mu\left(X_l^{(j)}\right) \tag{8}$$

where, the domain of the operator is the set:

$$\left\{\left[X_l^{(0)\mu}\right]_l^m \cup \left(\bigcup_{C_j \in S_\mu} \left[X_l^{(j)}\right]_l^m\right)\right\} \tag{9}$$

and the codomain of the operator is the set:

$$\left\{\left[Y_l^{(0)\mu}\right]_l^r \cup \left(\bigcup_{C_j \in S_\mu} \left[Y_l^{(k)}\right]_l^r\right)\right\} \tag{10}$$

Consider an algorithm of forming a set of codomains of the operator R_μ. Analyzing the expression in Eq. (9), we conclude that the domain of R_μ consists of three subsets.

The first subset. A contact $X_l^{(j)}$ of an element C_j, where $C_j \in S_\mu$, is connected to a contact $Y_l^{(k)}$ of an element C_k, where $C_k \in S_\mu$. It can be written:

$$[X_l^{(j)}]_\mu = \bigcup_{C_k \in S_\mu} [X^{(j,k)}] \tag{11}$$

In this particular case, it is evident the contact $Y_l^{(k)}$ is defined as

$$Y_l^{(k)} = R(X_l^{(j)}) \tag{12}$$

Thus, the operator R_μ is equal to the operator R:

$$R_\mu = R \tag{13}$$

A procedure of forming the first subset in Eq. (11) of the subsystem S_μ consists of two steps.

Step 1. Consider all rows j (where $C_j \in S_\mu$) and the pairs of numbers (k, l) in the Table 1. If the first number k is such that $C_k \in S_\mu$, then the second one l denotes the number of the output contact of the element C_k connected to the input contact $X_l^{(j)}$.

Step 2. The value of the operator R_μ has to be determined, using the expression in Eq. (11).

The two-step operation has to be executed for all rows j of the Table 1, where $C_j \in S_\mu$.

The second subset. A contact $X_l^{(j)}$ of the element C_j, where $C_j \in S_\mu$, is connected to the contact $Y_l^{(k)}$ of the element C_k, where $C_k \notin S_\mu$. It can be written:

$$[X_l^{(j)}]_\mu = \bigcup_{C_k \notin S_\mu} [X^{(j,k)}] \tag{14}$$

In this case, it is necessary to form a fictitious contact $Y_l^{(0)\mu}$ on the border of the subsystem S_μ, using the operator P'_μ in Eq. (7). The procedure of forming the second subset in Eq. (14) of the subsystem S_μ consists of two steps.

Step 1. Consider all rows j (such as $C_j \in S_\mu$) and the pairs of the numbers (k, l) in the Table 1. If the first number k is such that $C_k \notin S_\mu$, then the second number l is the number of the output contact of the element C_k connected to the input contact $X_l^{(j)}$.

Step 2. The operator R_μ has to be determined. It is equal to the operator P'_μ in Eq. (7). So,

$$R_\mu = P'_\mu \tag{15}$$

Finally, the fictitious contact has to be found as:

$$Y_l^{(0)\mu} = R_\mu(X_l^{(j)}) \tag{16}$$

The third subset. Consider the third subset that consists of fictitious input contacts $X_l^{(0)\mu}$ of the external environment $S'_{\mu 0}$. The operator Q'_μ in Eq. (7) has to be used in order to determine a contacts $Y_l^{(k)}$, which is connected to the fictitious contacts $X_l^{(0)\mu}$. It is easy to conclude that:

$$Y_l^{(k)} = (Q'_\mu)^{-1}(X_l^{(0)\mu}) \tag{17}$$

where, the operator $(Q'_\mu)^{-1}$ is an inverse operator Q'_μ.

Thus,

$$R_\mu = (Q'_\mu)^{-1} \tag{18}$$

Note, the operator Q'_μ is a one-to-one operator.

Finally, the procedure of constructing the operator R_μ is defined by the expression:

$$Y_l^{(k)} = \begin{cases} R\left(X_l^{(j)}\right) \text{for } X_l^{(j)} \in \bigcup_{C_j \in S_\mu} \bigcup_{C_k \in S_\mu} \left[X^{(j,k)}\right] \\ P'_\mu\left(X_l^{(j)}\right) \text{for } X_l^{(j)} \in \bigcup_{C_j \in S_\mu} \bigcup_{C_k \notin S_\mu} \left[X^{(j,k)}\right] \\ (Q'_\mu)^{-1}\left(X_l^{(0)\mu}\right) \text{for } X_l^{(0)\mu} \in \left[X_l^{(0)\mu}\right]_l^m \end{cases} \tag{19}$$

As an example, the values of the operator R_μ where $\mu=\mu 1$ are given in Table 4. Like the Table 1, the Table 4 shows the values of the operator $R_{\mu 1}$. The row 0 in this table corresponds to the external environment $S'_{\mu 0}$. The fictitious contacts $X_l^{(0)\mu}$ and $Y_l^{(0)\mu}$ correspond to the input and output contacts of $S'_{\mu 0}$, respectively. The rows 1 and 2 correspond to the elements C_1 and C_2, respectively. The number of columns of this table is equal to the maximum number of input contacts among the C_1, C_2 and $S'_{\mu 0}$ and is equal to 4. At the intersection of rows and columns there are pairs of numbers (k, l) indicating the number of the element k (where $C_k \in S_\mu$ and $S'_{\mu 0}$) and the number of its output contact l, to which the input contact $X_l^{(j)}$ is connected.

The case, where a subsystem S_μ is considered as an independent system has been applied to the design of secure hardware systems. In the paper [15], hardware obfuscation technique on the base of proposed formalism is described.

Рисунок Б.27 – Стаття «Dimension Reduction for Network Systems Using Structure Model Aggregation»

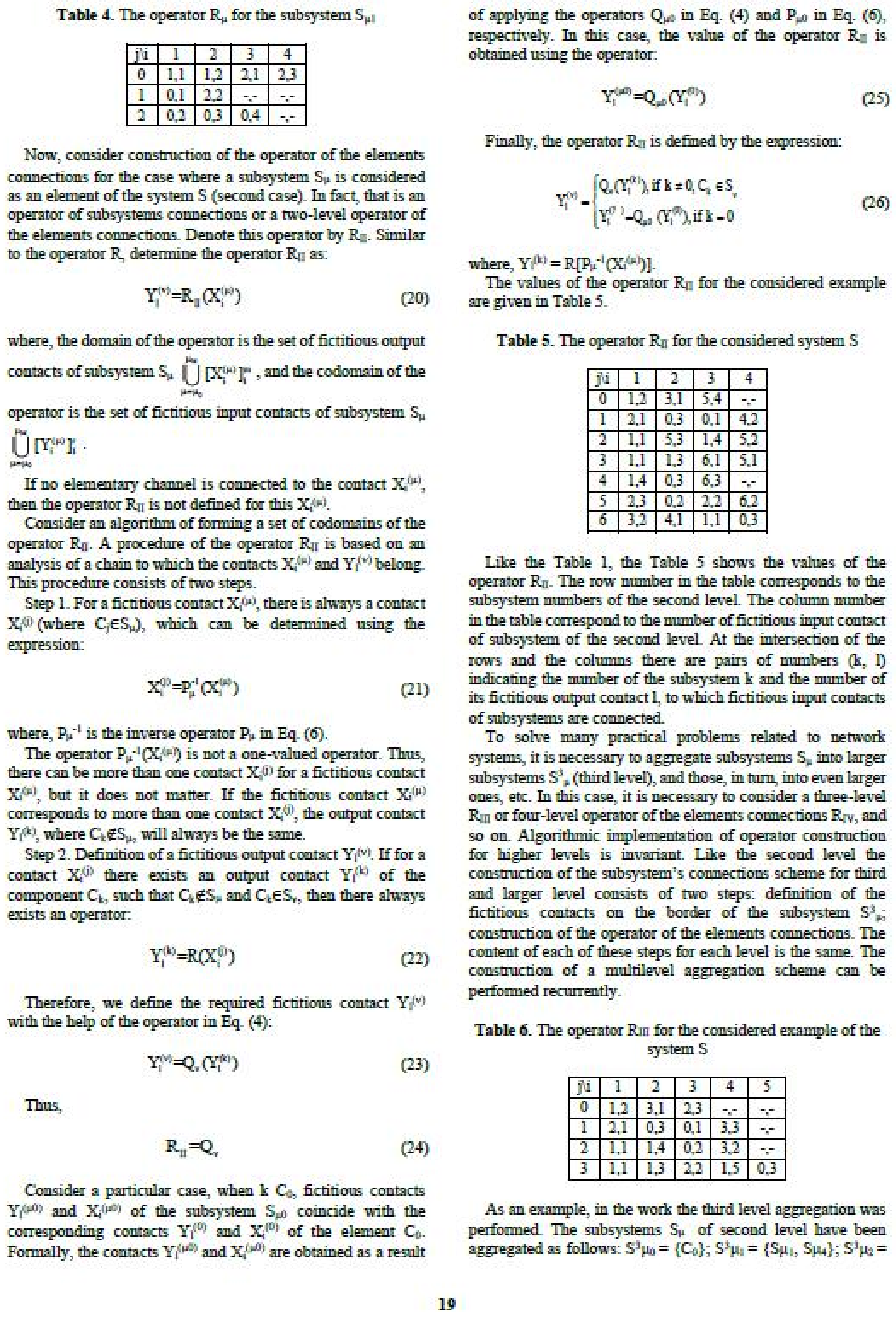


Table 4. The operator R_μ for the subsystem $S_{\mu 1}$

j\i	1	2	3	4
0	1,1	1,2	2,1	2,3
1	0,1	2,2	-,-	-,-
2	0,2	0,3	0,4	-,-

Now, consider construction of the operator of the elements connections for the case where a subsystem S_μ is considered as an element of the system S (second case). In fact, that is an operator of subsystems connections or a two-level operator of the elements connections. Denote this operator by R_{II}. Similar to the operator R, determine the operator R_{II} as:

$$Y_l^{(\nu)}=R_{II}(X_i^{(\mu)}) \tag{20}$$

where, the domain of the operator is the set of fictitious output contacts of subsystem S_μ $\bigcup\limits_{\mu=\mu_0}^{\mu_M}[X_i^{(\mu)}]_l^m$, and the codomain of the operator is the set of fictitious input contacts of subsystem S_μ $\bigcup\limits_{\mu=\mu_0}^{\mu_M}[Y_l^{(\mu)}]_l$.

If no elementary channel is connected to the contact $X_i^{(\mu)}$, then the operator R_{II} is not defined for this $X_i^{(\mu)}$.

Consider an algorithm of forming a set of codomains of the operator R_{II}. A procedure of the operator R_{II} is based on an analysis of a chain to which the contacts $X_i^{(\mu)}$ and $Y_l^{(\nu)}$ belong. This procedure consists of two steps.

Step 1. For a fictitious contact $X_i^{(\mu)}$, there is always a contact $X_i^{(j)}$ (where $C_j \in S_\mu$), which can be determined using the expression:

$$X_i^{(j)}=P_\mu^{-1}(X_i^{(\mu)}) \tag{21}$$

where, P_μ^{-1} is the inverse operator P_μ in Eq. (6).

The operator $P_\mu^{-1}(X_i^{(\mu)})$ is not a one-valued operator. Thus, there can be more than one contact $X_i^{(j)}$ for a fictitious contact $X_i^{(\mu)}$, but it does not matter. If the fictitious contact $X_i^{(\mu)}$ corresponds to more than one contact $X_i^{(j)}$, the output contact $Y_l^{(k)}$, where $C_k \notin S_\mu$, will always be the same.

Step 2. Definition of a fictitious output contact $Y_l^{(\nu)}$. If for a contact $X_i^{(j)}$ there exists an output contact $Y_l^{(k)}$ of the component C_k, such that $C_k \notin S_\mu$ and $C_k \in S_\nu$, then there always exists an operator:

$$Y_l^{(k)}=R(X_i^{(j)}) \tag{22}$$

Therefore, we define the required fictitious contact $Y_l^{(\nu)}$ with the help of the operator in Eq. (4):

$$Y_l^{(\nu)}=Q_\nu(Y_l^{(k)}) \tag{23}$$

Thus,

$$R_{II}=Q_\nu \tag{24}$$

Consider a particular case, when k C_0, fictitious contacts $Y_l^{(\mu 0)}$ and $X_i^{(\mu 0)}$ of the subsystem $S_{\mu 0}$ coincide with the corresponding contacts $Y_l^{(0)}$ and $X_i^{(0)}$ of the element C_0. Formally, the contacts $Y_l^{(\mu 0)}$ and $X_i^{(\mu 0)}$ are obtained as a result of applying the operators $Q_{\mu 0}$ in Eq. (4) and $P_{\mu 0}$ in Eq. (6), respectively. In this case, the value of the operator R_{II} is obtained using the operator:

$$Y_l^{(\mu 0)}=Q_{\mu 0}(Y_l^{(0)}) \tag{25}$$

Finally, the operator R_{II} is defined by the expression:

$$Y_l^{(\nu)}=\begin{cases} Q_\nu(Y_l^{(k)}), \text{ if } k \neq 0, C_k \in S_\nu \\ Y_l^{(\mu 0)}=Q_{\mu 0}(Y_l^{(0)}), \text{ if } k=0 \end{cases} \tag{26}$$

where, $Y_l^{(k)} = R[P_\mu^{-1}(X_i^{(\mu)})]$.

The values of the operator R_{II} for the considered example are given in Table 5.

Table 5. The operator R_{II} for the considered system S

j\i	1	2	3	4
0	1,2	3,1	5,4	-,-
1	2,1	0,3	0,1	4,2
2	1,1	5,3	1,4	5,2
3	1,1	1,3	6,1	5,1
4	1,4	0,3	6,3	-,-
5	2,3	0,2	2,2	6,2
6	3,2	4,1	1,1	0,3

Like the Table 1, the Table 5 shows the values of the operator R_{II}. The row number in the table corresponds to the subsystem numbers of the second level. The column number in the table correspond to the number of fictitious input contact of subsystem of the second level. At the intersection of the rows and the columns there are pairs of numbers (k, l) indicating the number of the subsystem k and the number of its fictitious output contact l, to which fictitious input contacts of subsystems are connected.

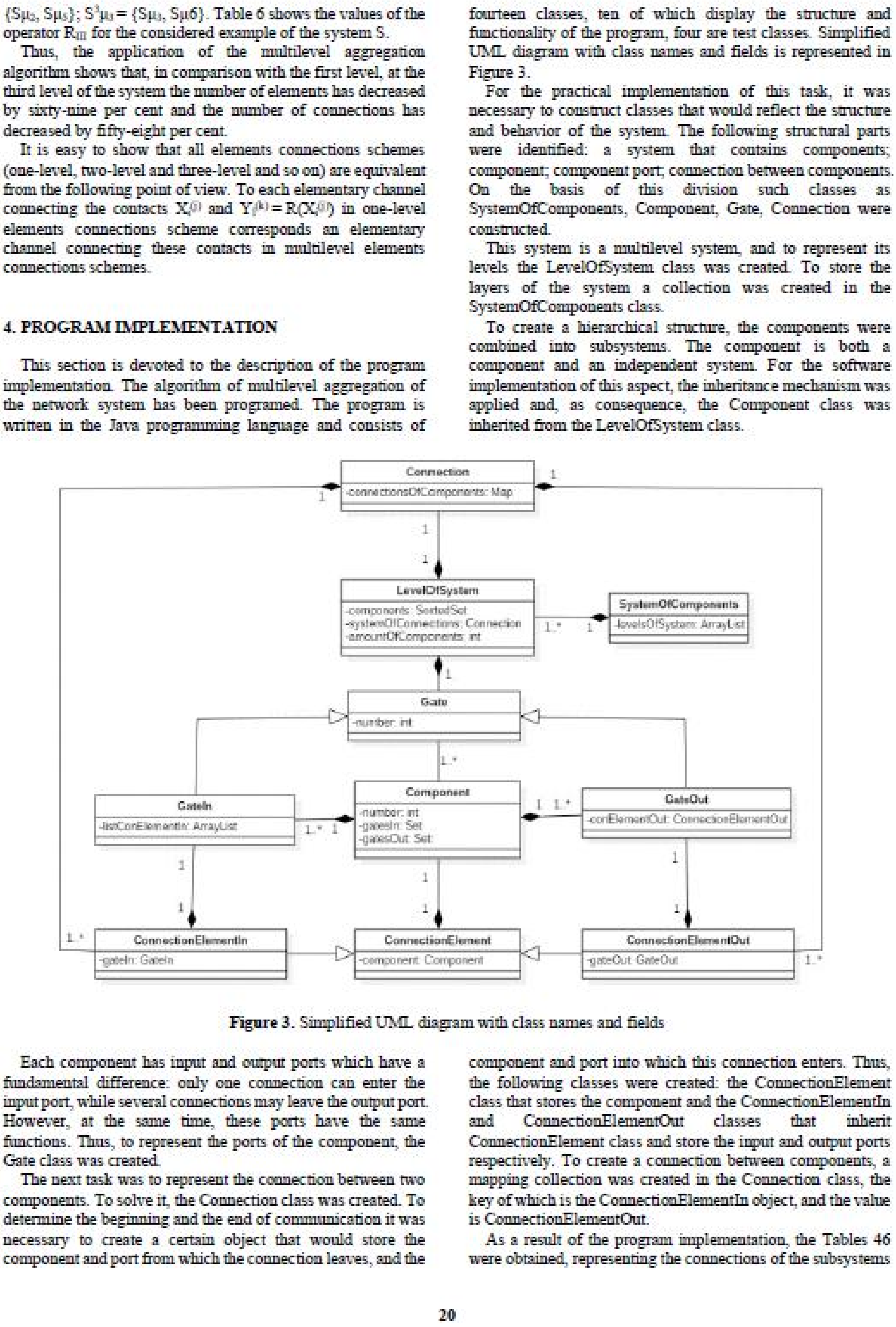
To solve many practical problems related to network systems, it is necessary to aggregate subsystems S_μ into larger subsystems S^3_μ (third level), and those, in turn, into even larger ones, etc. In this case, it is necessary to consider a three-level R_{III} or four-level operator of the elements connections R_{IV}, and so on. Algorithmic implementation of operator construction for higher levels is invariant. Like the second level the construction of the subsystem's connections scheme for third and larger level consists of two steps: definition of the fictitious contacts on the border of the subsystem S^3_μ; construction of the operator of the elements connections. The content of each of these steps for each level is the same. The construction of a multilevel aggregation scheme can be performed recurrently.

Table 6. The operator R_{III} for the considered example of the system S

j\i	1	2	3	4	5
0	1,2	3,1	2,3	-,-	-,-
1	2,1	0,3	0,1	3,3	-,-
2	1,1	1,4	0,2	3,2	-,-
3	1,1	1,3	2,2	1,5	0,3

As an example, in the work the third level aggregation was performed. The subsystems S_μ of second level have been aggregated as follows: $S^3\mu_0 = \{C_0\}$; $S^3\mu_1 = \{S\mu_1, S\mu_4\}$; $S^3\mu_2 =$

Рисунок Б.28 – Стаття «Dimension Reduction for Network Systems Using Structure Model Aggregation»



$\{S\mu_2, S\mu_5\}$; $S^3\mu_3 = \{S\mu_3, S\mu_6\}$. Table 6 shows the values of the operator R_{III} for the considered example of the system S.

Thus, the application of the multilevel aggregation algorithm shows that, in comparison with the first level, at the third level of the system the number of elements has decreased by sixty-nine per cent and the number of connections has decreased by fifty-eight per cent.

It is easy to show that all elements connections schemes (one-level, two-level and three-level and so on) are equivalent from the following point of view. To each elementary channel connecting the contacts $X_i^{(j)}$ and $Y_j^{(k)} = R(X_i^{(j)})$ in one-level elements connections scheme corresponds an elementary channel connecting these contacts in multilevel elements connections schemes.

4. PROGRAM IMPLEMENTATION

This section is devoted to the description of the program implementation. The algorithm of multilevel aggregation of the network system has been programed. The program is written in the Java programming language and consists of fourteen classes, ten of which display the structure and functionality of the program, four are test classes. Simplified UML diagram with class names and fields is represented in Figure 3.

For the practical implementation of this task, it was necessary to construct classes that would reflect the structure and behavior of the system. The following structural parts were identified: a system that contains components; component; component port; connection between components. On the basis of this division such classes as SystemOfComponents, Component, Gate, Connection were constructed.

This system is a multilevel system, and to represent its levels the LevelOfSystem class was created. To store the layers of the system a collection was created in the SystemOfComponents class.

To create a hierarchical structure, the components were combined into subsystems. The component is both a component and an independent system. For the software implementation of this aspect, the inheritance mechanism was applied and, as consequence, the Component class was inherited from the LevelOfSystem class.

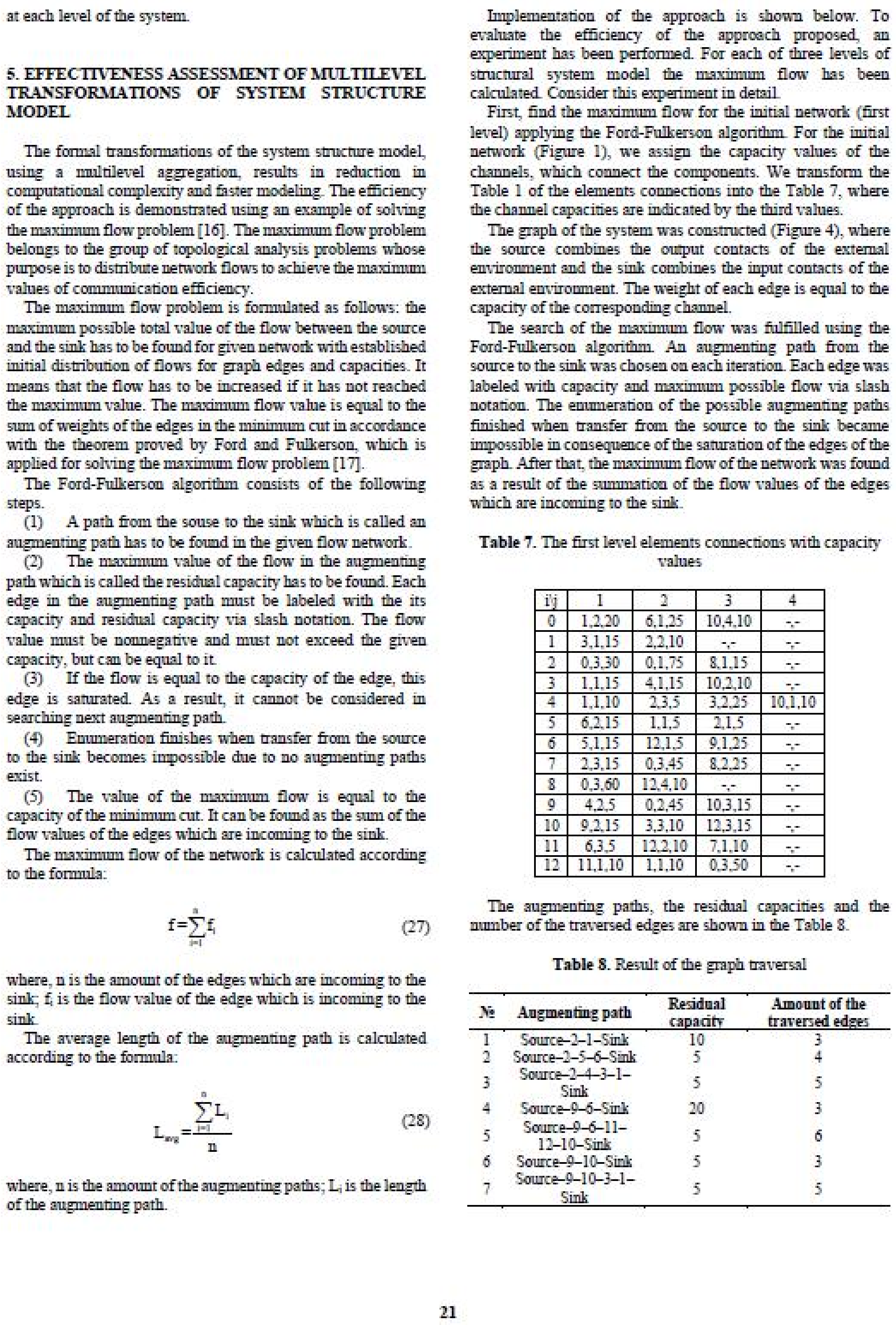
Figure 3. Simplified UML diagram with class names and fields

Each component has input and output ports which have a fundamental difference: only one connection can enter the input port, while several connections may leave the output port. However, at the same time, these ports have the same functions. Thus, to represent the ports of the component, the Gate class was created.

The next task was to represent the connection between two components. To solve it, the Connection class was created. To determine the beginning and the end of communication it was necessary to create a certain object that would store the component and port from which the connection leaves, and the component and port into which this connection enters. Thus, the following classes were created: the ConnectionElement class that stores the component and the ConnectionElementIn and ConnectionElementOut classes that inherit ConnectionElement class and store the input and output ports respectively. To create a connection between components, a mapping collection was created in the Connection class, the key of which is the ConnectionElementIn object, and the value is ConnectionElementOut.

As a result of the program implementation, the Tables 46 were obtained, representing the connections of the subsystems

Рисунок Б.29 – Стаття «Dimension Reduction for Network Systems Using Structure Model Aggregation»



at each level of the system.

5. EFFECTIVENESS ASSESSMENT OF MULTILEVEL TRANSFORMATIONS OF SYSTEM STRUCTURE MODEL

The formal transformations of the system structure model, using a multilevel aggregation, results in reduction in computational complexity and faster modeling. The efficiency of the approach is demonstrated using an example of solving the maximum flow problem [16]. The maximum flow problem belongs to the group of topological analysis problems whose purpose is to distribute network flows to achieve the maximum values of communication efficiency.

The maximum flow problem is formulated as follows: the maximum possible total value of the flow between the source and the sink has to be found for given network with established initial distribution of flows for graph edges and capacities. It means that the flow has to be increased if it has not reached the maximum value. The maximum flow value is equal to the sum of weights of the edges in the minimum cut in accordance with the theorem proved by Ford and Fulkerson, which is applied for solving the maximum flow problem [17].

The Ford-Fulkerson algorithm consists of the following steps.

(1) A path from the souse to the sink which is called an augmenting path has to be found in the given flow network.

(2) The maximum value of the flow in the augmenting path which is called the residual capacity has to be found. Each edge in the augmenting path must be labeled with its capacity and residual capacity via slash notation. The flow value must be nonnegative and must not exceed the given capacity, but can be equal to it.

(3) If the flow is equal to the capacity of the edge, this edge is saturated. As a result, it cannot be considered in searching next augmenting path.

(4) Enumeration finishes when transfer from the source to the sink becomes impossible due to no augmenting paths exist.

(5) The value of the maximum flow is equal to the capacity of the minimum cut. It can be found as the sum of the flow values of the edges which are incoming to the sink.

The maximum flow of the network is calculated according to the formula:

$$f=\sum_{i=1}^{n} f_i \tag{27}$$

where, n is the amount of the edges which are incoming to the sink; f_i is the flow value of the edge which is incoming to the sink.

The average length of the augmenting path is calculated according to the formula:

$$L_{avg}=\frac{\sum_{i=1}^{n} L_i}{n} \tag{28}$$

where, n is the amount of the augmenting paths; L_i is the length of the augmenting path.

Implementation of the approach is shown below. To evaluate the efficiency of the approach proposed, an experiment has been performed. For each of three levels of structural system model the maximum flow has been calculated. Consider this experiment in detail.

First, find the maximum flow for the initial network (first level) applying the Ford-Fulkerson algorithm. For the initial network (Figure 1), we assign the capacity values of the channels, which connect the components. We transform the Table 1 of the elements connections into the Table 7, where the channel capacities are indicated by the third values.

The graph of the system was constructed (Figure 4), where the source combines the output contacts of the external environment and the sink combines the input contacts of the external environment. The weight of each edge is equal to the capacity of the corresponding channel.

The search of the maximum flow was fulfilled using the Ford-Fulkerson algorithm. An augmenting path from the source to the sink was chosen on each iteration. Each edge was labeled with capacity and maximum possible flow via slash notation. The enumeration of the possible augmenting paths finished when transfer from the source to the sink became impossible in consequence of the saturation of the edges of the graph. After that, the maximum flow of the network was found as a result of the summation of the flow values of the edges which are incoming to the sink.

Table 7. The first level elements connections with capacity values

i\j	1	2	3	4
0	1,2,20	6,1,25	10,4,10	-,-
1	3,1,15	2,2,10	-,-	-,-
2	0,3,30	0,1,75	8,1,15	-,-
3	1,1,15	4,1,15	10,2,10	-,-
4	1,1,10	2,3,5	3,2,25	10,1,10
5	6,2,15	1,1,5	2,1,5	-,-
6	5,1,15	12,1,5	9,1,25	-,-
7	2,3,15	0,3,45	8,2,25	-,-
8	0,3,60	12,4,10	-,-	-,-
9	4,2,5	0,2,45	10,3,15	-,-
10	9,2,15	3,3,10	12,3,15	-,-
11	6,3,5	12,2,10	7,1,10	-,-
12	11,1,10	1,1,10	0,3,50	-,-

The augmenting paths, the residual capacities and the number of the traversed edges are shown in the Table 8.

Table 8. Result of the graph traversal

№	Augmenting path	Residual capacity	Amount of the traversed edges
1	Source–2–1–Sink	10	3
2	Source–2–5–6–Sink	5	4
3	Source–2–4–3–1–Sink	5	5
4	Source–9–6–Sink	20	3
5	Source–9–6–11–12–10–Sink	5	6
6	Source–9–10–Sink	5	3
7	Source–9–10–3–1–Sink	5	5

Рисунок Б.30 – Стаття «Dimension Reduction for Network Systems Using Structure Model Aggregation»

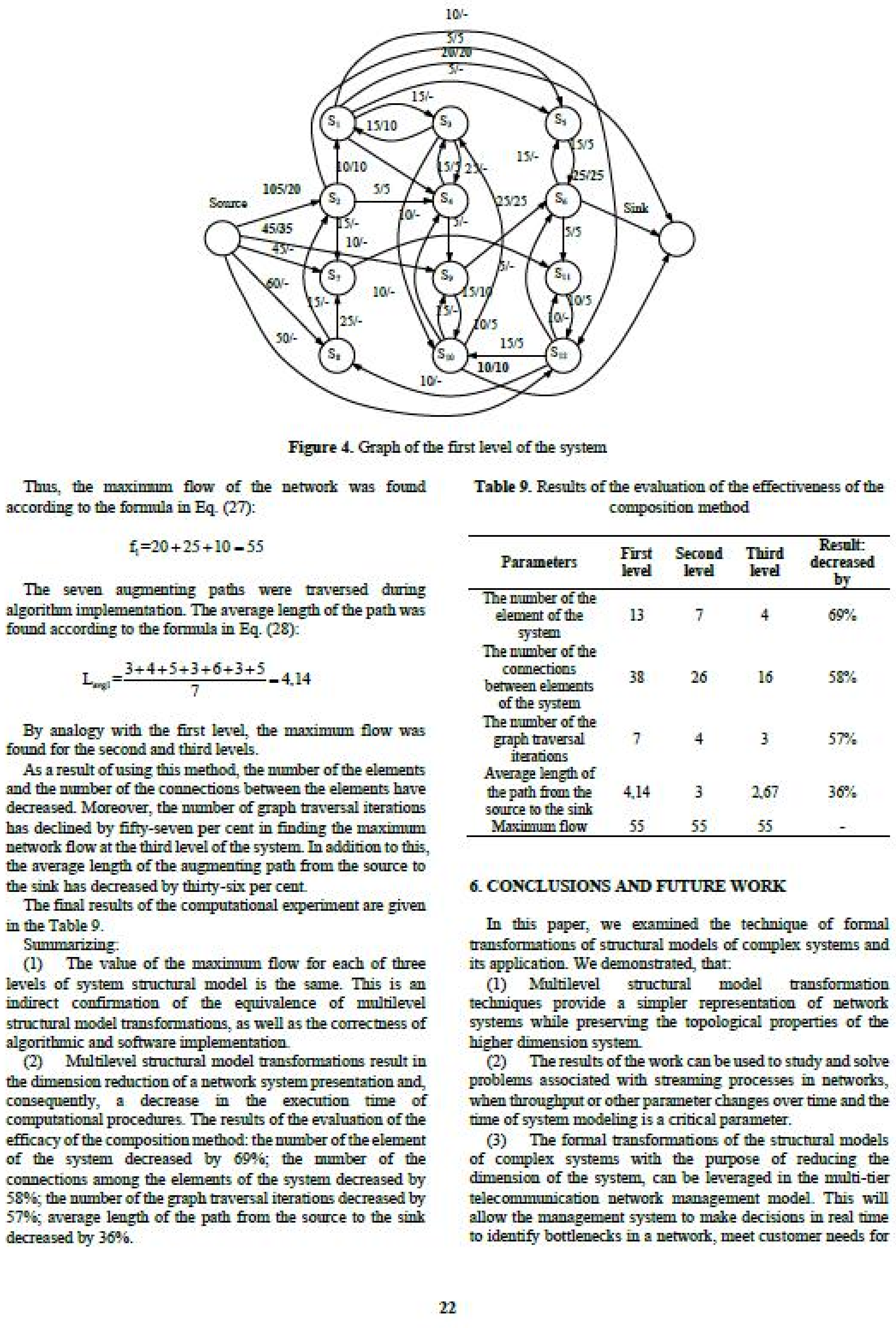


Figure 4. Graph of the first level of the system

Thus, the maximum flow of the network was found according to the formula in Eq. (27):

$$f_1 = 20 + 25 + 10 = 55$$

The seven augmenting paths were traversed during algorithm implementation. The average length of the path was found according to the formula in Eq. (28):

$$L_{avg1} = \frac{3+4+5+3+6+3+5}{7} = 4{,}14$$

By analogy with the first level, the maximum flow was found for the second and third levels.

As a result of using this method, the number of the elements and the number of the connections between the elements have decreased. Moreover, the number of graph traversal iterations has declined by fifty-seven per cent in finding the maximum network flow at the third level of the system. In addition to this, the average length of the augmenting path from the source to the sink has decreased by thirty-six per cent.

The final results of the computational experiment are given in the Table 9.

Summarizing:

(1) The value of the maximum flow for each of three levels of system structural model is the same. This is an indirect confirmation of the equivalence of multilevel structural model transformations, as well as the correctness of algorithmic and software implementation.

(2) Multilevel structural model transformations result in the dimension reduction of a network system presentation and, consequently, a decrease in the execution time of computational procedures. The results of the evaluation of the efficacy of the composition method: the number of the element of the system decreased by 69%; the number of the connections among the elements of the system decreased by 58%; the number of the graph traversal iterations decreased by 57%; average length of the path from the source to the sink decreased by 36%.

Table 9. Results of the evaluation of the effectiveness of the composition method

Parameters	First level	Second level	Third level	Result: decreased by
The number of the element of the system	13	7	4	69%
The number of the connections between elements of the system	38	26	16	58%
The number of the graph traversal iterations	7	4	3	57%
Average length of the path from the source to the sink	4,14	3	2,67	36%
Maximum flow	55	55	55	-

6. CONCLUSIONS AND FUTURE WORK

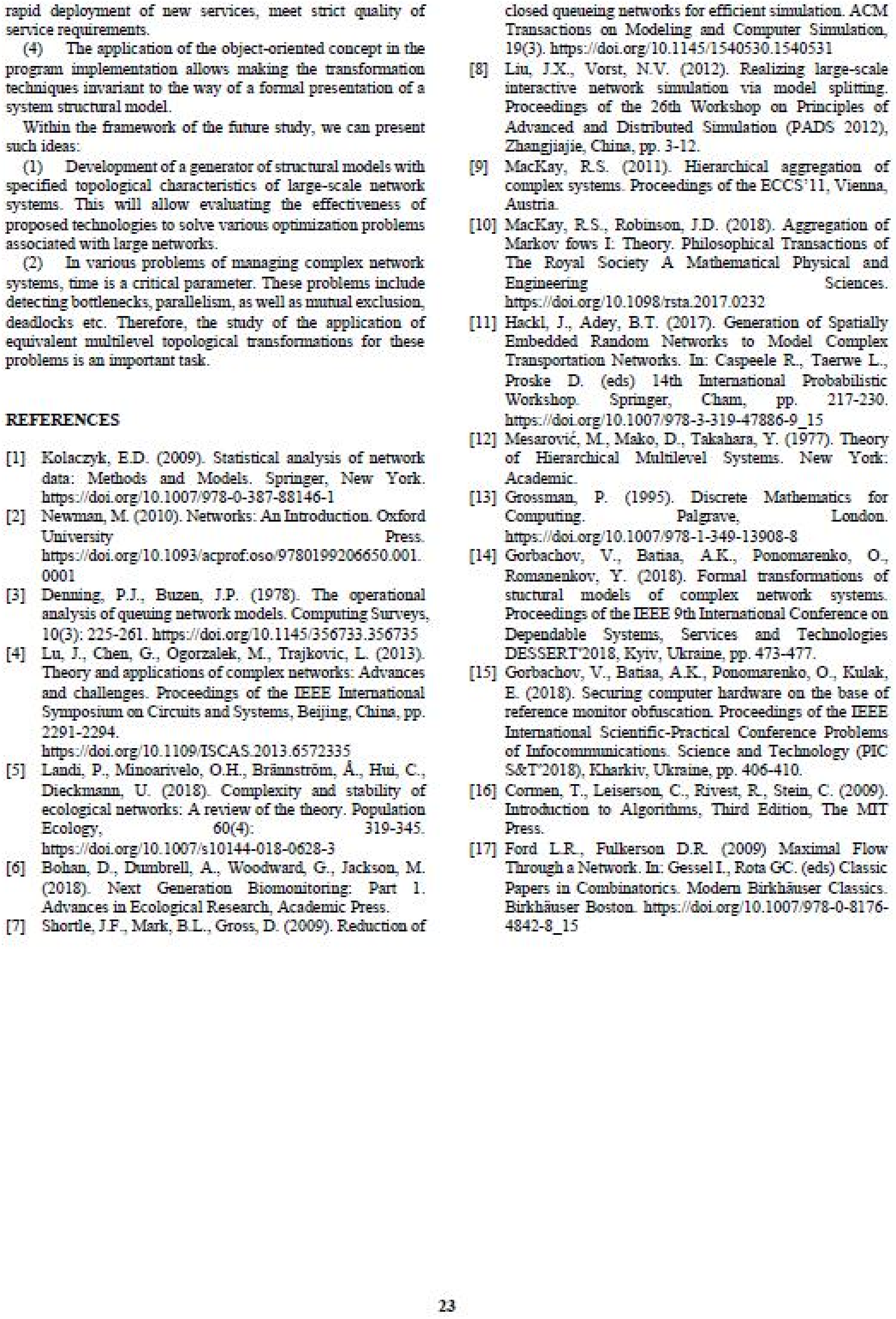
In this paper, we examined the technique of formal transformations of structural models of complex systems and its application. We demonstrated, that:

(1) Multilevel structural model transformation techniques provide a simpler representation of network systems while preserving the topological properties of the higher dimension system.

(2) The results of the work can be used to study and solve problems associated with streaming processes in networks, when throughput or other parameter changes over time and the time of system modeling is a critical parameter.

(3) The formal transformations of the structural models of complex systems with the purpose of reducing the dimension of the system, can be leveraged in the multi-tier telecommunication network management model. This will allow the management system to make decisions in real time to identify bottlenecks in a network, meet customer needs for

Рисунок Б.31 – Стаття «Dimension Reduction for Network Systems Using Structure Model Aggregation»



rapid deployment of new services, meet strict quality of service requirements.

(4) The application of the object-oriented concept in the program implementation allows making the transformation techniques invariant to the way of a formal presentation of a system structural model.

Within the framework of the future study, we can present such ideas:

(1) Development of a generator of structural models with specified topological characteristics of large-scale network systems. This will allow evaluating the effectiveness of proposed technologies to solve various optimization problems associated with large networks.

(2) In various problems of managing complex network systems, time is a critical parameter. These problems include detecting bottlenecks, parallelism, as well as mutual exclusion, deadlocks etc. Therefore, the study of the application of equivalent multilevel topological transformations for these problems is an important task.

REFERENCES

[1] Kolaczyk, E.D. (2009). Statistical analysis of network data: Methods and Models. Springer, New York. https://doi.org/10.1007/978-0-387-88146-1

[2] Newman, M. (2010). Networks: An Introduction. Oxford University Press. https://doi.org/10.1093/acprof:oso/9780199206650.001.0001

[3] Denning, P.J., Buzen, J.P. (1978). The operational analysis of queuing network models. Computing Surveys, 10(3): 225-261. https://doi.org/10.1145/356733.356735

[4] Lu, J., Chen, G., Ogorzalek, M., Trajkovic, L. (2013). Theory and applications of complex networks: Advances and challenges. Proceedings of the IEEE International Symposium on Circuits and Systems, Beijing, China, pp. 2291-2294. https://doi.org/10.1109/ISCAS.2013.6572335

[5] Landi, P., Minoarivelo, O.H., Brännström, Å., Hui, C., Dieckmann, U. (2018). Complexity and stability of ecological networks: A review of the theory. Population Ecology, 60(4): 319-345. https://doi.org/10.1007/s10144-018-0628-3

[6] Bohan, D., Dumbrell, A., Woodward, G., Jackson, M. (2018). Next Generation Biomonitoring: Part 1. Advances in Ecological Research, Academic Press.

[7] Shortle, J.F., Mark, B.L., Gross, D. (2009). Reduction of closed queueing networks for efficient simulation. ACM Transactions on Modeling and Computer Simulation, 19(3). https://doi.org/10.1145/1540530.1540531

[8] Liu, J.X., Vorst, N.V. (2012). Realizing large-scale interactive network simulation via model splitting. Proceedings of the 26th Workshop on Principles of Advanced and Distributed Simulation (PADS 2012), Zhangjiajie, China, pp. 3-12.

[9] MacKay, R.S. (2011). Hierarchical aggregation of complex systems. Proceedings of the ECCS'11, Vienna, Austria.

[10] MacKay, R.S., Robinson, J.D. (2018). Aggregation of Markov fows I: Theory. Philosophical Transactions of The Royal Society A Mathematical Physical and Engineering Sciences. https://doi.org/10.1098/rsta.2017.0232

[11] Hackl, J., Adey, B.T. (2017). Generation of Spatially Embedded Random Networks to Model Complex Transportation Networks. In: Caspeele R., Taerwe L., Proske D. (eds) 14th International Probabilistic Workshop. Springer, Cham, pp. 217-230. https://doi.org/10.1007/978-3-319-47886-9_15

[12] Mesarović, M., Mako, D., Takahara, Y. (1977). Theory of Hierarchical Multilevel Systems. New York: Academic.

[13] Grossman, P. (1995). Discrete Mathematics for Computing. Palgrave, London. https://doi.org/10.1007/978-1-349-13908-8

[14] Gorbachov, V., Batiaa, A.K., Ponomarenko, O., Romanenkov, Y. (2018). Formal transformations of stuctural models of complex network systems. Proceedings of the IEEE 9th International Conference on Dependable Systems, Services and Technologies DESSERT'2018, Kyiv, Ukraine, pp. 473-477.

[15] Gorbachov, V., Batiaa, A.K., Ponomarenko, O., Kulak, E. (2018). Securing computer hardware on the base of reference monitor obfuscation. Proceedings of the IEEE International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T'2018), Kharkiv, Ukraine, pp. 406-410.

[16] Cormen, T., Leiserson, C., Rivest, R., Stein, C. (2009). Introduction to Algorithms, Third Edition, The MIT Press.

[17] Ford L.R., Fulkerson D.R. (2009) Maximal Flow Through a Network. In: Gessel I., Rota GC. (eds) Classic Papers in Combinatorics. Modern Birkhäuser Classics. Birkhäuser Boston. https://doi.org/10.1007/978-0-8176-4842-8_15

Рисунок Б.32 – Стаття «Dimension Reduction for Network Systems Using Structure Model Aggregation»