

І.Г. Цмоць, д.т.н., проф., зав. кафедри АСУ, НУ “Львівська політехніка”,  
Т.В. Теслюк, аспірант кафедри АСУ, НУ “Львівська політехніка”,  
В.Я. Коваль, асистент кафедри ІСТ, НУ “Львівська політехніка”,  
Д. П. Веніков, к.т.н., доц. кафедри ІСТ, ШПТ, НУ “Львівська політехніка”,  
Я. Г. Притуляк, к.т.н., доц. кафедри ІСТ, ШПТ, НУ “Львівська політехніка”.

## РОЗРОБЛЕННЯ ІНФОРМАЦІЙНИХ МОДЕЛЕЙ ТА БД ДЛЯ СИСТЕМИ АВТОМАТИЗАЦІЇ СТРУКТУРНОГО СИНТЕЗУ З ВИКОРИСТАННЯМ МОРФОЛОГІЧНОГО МЕТОДУ

**Abstract.** Information models for formalizing the structure of the developed system and I-OR tree are constructed. These models are used in the process of applying morphological synthesis. The developed models are based on the theory of graphs, which makes it possible to apply them to the structural schemes of systems of any nature. In addition, information support has been developed on the basis of the use of the MongoDB database management system for the structural synthesis of automation system. The results of structural synthesis of devices based on microcontrollers are presented.

**Анотація.** В роботі побудовано інформаційні моделі для формалізації структури розроблюваної системи та I-АБО дерева, яке використано в процесі застосування морфологічного синтезу. Розроблені моделі ґрунтуються на теорії графів. Окрім того, розроблено інформаційне забезпечення на основі використання системи керування базами даних MongoDB для системи автоматизації структурного синтезу та наведено результати структурного синтезу пристроїв на базі мікроконтролерів.

### Вступ

Складність технічних систем зростає з кожним роком. Відповідно, синтез та аналіз таких систем неможливий без використання програмно-технічних засобів автоматизації. На сьогодні, існує багато методів та підходів синтезу технічних систем [1], але найбільш відомим є морфологічний метод [2]. Цей метод використовується в різних областях науки та техніки для розв’язання задач структурного синтезу, зокрема: [3] - в задачах технологічного передбачення для пошуку принципово нових або удосконалення існуючих соціально-економічних, організаційних або інших складних систем і для аналізу ситуації з деяким ступенем невизначеності; [4] - для розробки управлінських рішень; [5] – для структурного синтезу мікроелектромеханічних систем; [6, 7] - аналізу ситуації в майбутньому з певним ступенем невизначеності та ін.

З цих позицій, розроблення системи структурного синтезу з використанням морфологічного методу – є актуальним завданням.

## 1. Інформаційна модель структури системи

Важливим етапом розроблення програмних систем є побудова інформаційного забезпечення, яке включає інформаційні моделі та базу даних. Однією з основних є інформаційна модель структури об'єкта дослідження чи розроблення. В загальному випадку, під структурою об'єкта проектування розуміємо набір елементів та зв'язки між ними, що можна описати за допомогою моделі з використанням теорії графів [8]:

$$G=(E,R), \quad (1)$$

де:  $E = (e_1, e_2, \dots, e_t)$  – множина елементів структури системи;  $t$  – кількість елементів структури системи;  $R = (r_1, r_2, \dots, r_s)$  – множина зв'язків між елементами структури (ребер);  $s$  – кількість зв'язків між елементами структури системи.

Окрім того, важливим елементом моделі структури системи, є матриці інцидентності та сумісності, які дають змогу відобразити інформацію про зв'язки між елементами структури та їх напрям. Тому, в даному випадку, мають справу з орієнтованим графом, а елементи матриці інцидентності можуть приймати значення 0, якщо елементи (вершини графа) не інцидентні, +1, якщо дуга (зв'язок) орієнтована від елемента та -1, якщо дуга орієнтована до елемента.

Наведена інформаційна модель дає можливість описати будь-яку структуру проектованої системи чи об'єкту та швидко її модифікувати. Окрім того, для зв'язку з іншими системами синтезу проектована структура записується в XML форматі [9].

## 2. Інформаційна модель I-АБО дерева

Важливим елементом процесу автоматизованого синтезу структур проектованої системи використанням морфологічного синтезу є побудова інформаційної моделі I-АБО дерева.

Відповідно, з математичної точки зору I-АБО дерево можна зобразити за допомогою наступного графа [8]:

$$G_{I-АБО} = (V_{I-АБО}, R_{I-АБО}), \quad (3)$$

де  $V_{I-АБО} = (v_1, v_2, \dots, v_n)$  – множина вузлів I-АБО дерева,  $n$  – кількість вузлів, а  $R_{I-АБО} = (r_1, r_2, \dots, r_m)$  – множина ребер I-АБО дерева,  $m$  – кількість ребер.

На практиці – правила побудови такого дерева задаються користувачем у вхідному \*.ТХТ файлі. Для прикладу, на рисунку 1 зображено I-АБО дерево, що використовується для синтезу пристрою на базі мікроконтролерів.

Кожен рядок, розробленої структури файлу правил побудови I-АБО дерева, відповідає за відповідний вузол в дереві. Для зручності користувача, ключовими словами обрано позначення відповідних логічних операцій і - & та або - |.

Модель структур одного правила має бути представлена у такому вигляді:

```
{[Назва_вузла]:[Правило_розгалуження]:{[Елемент1],[Елемент2],[Елемент3]}}, де: [Назва_вузла] – назва вузла в дереві (може бути будь-яким на вибір); [Правило_розгалуження] – правило за яким порядком будуть проходити через дочірні вузли; [Елемент1], [Елемент2], [Елемент3] – назви дочірніх вузлів.
```

```
root:&:{a,b,c}  
a:&:{a1,a2,a3}  
b:/{b1,b2}  
a1:/{Display}  
a2:/{Power sources}  
a3:/{Buttons}  
b1:/{Servos}  
b2:/{Steppers}  
c:/{Boards}
```

Рис. 1. Структура вхідного файлу з правилами побудови І-АБО дерева

Назва вузла одного з правил має мати назву ‘root’, що відповідає за корінь всього дерева. Назви вузлів дерева, які є листками, повинні співпадати з назвами відповідних колекцій в підключеній базі даних, у протилежному випадку назва буде розглядатись як окремий елемент під час структурного синтезу. Якщо ж назва співпадає з колекцією в базі даних, то під час структурного синтезу всі записи цієї колекції будуть розглядатись як дочірні елементи в дереві і їх будуть проходити за правилом АБО.

Для реалізації програмного забезпечення системи автоматизованого структурного синтезу з використанням морфологічного методу було застосовано ієрархічну деревовидну модель даних.

Ієрархічна деревовидна структура даних представляє сукупність елементів, пов'язаних між собою за певними правилами. Об'єкти, пов'язані ієрархічними відношеннями і утворюють орієнтований граф. До основних понять ієрархічної структури відносяться: рівень, елемент (вузол), зв'язок. Вузол - це сукупність атрибутів даних, що описують деякий об'єкт. На схемі ієрархічного дерева вузли представляються вершинами графа. Кожен вузол на нижчому рівні пов'язаний лише з одним вузлом, що знаходиться на вищому рівні. Ієрархічне дерево має тільки одну вершину, не підпорядковану жодній іншій вершині і знаходиться на самому верхньому (першому) рівні. Залежні вузли знаходяться на другому, третьому і т.д. рівнях. Кількість дерев у базі даних визначається числом кореневих записів. До кожного запису бази даних існує тільки один (ієрархічний) шлях від кореневого запису.

Побудова І-АБО дерева для генерації альтернатив структури мікроелектронних пристроїв відбувається наступним чином. Спершу на верхньому рівні маємо вершину І, яка визначає основні модулі пристрою, а саме: *a*, *b* і *c*. Кожен модуль, в свою чергу, включає певні підмодулі (їх може бути безліч, залежно від потреб проєктанта). Модуль *a* є вершиною типу І,

відповідно модулі  $a_1$ ,  $a_2$ ,  $a_3$  не мають власних підмодулів отже вони є кінцевими (листочками). Якщо модуль є кінцевим – тоді відбувається перебір його значень, тобто, для прикладу, для модуля  $a_1$  перебір всіх можливих комбінацій з бази даних дисплеїв (displays).

Аналогічно відбувається побудова решти правил заданих користувачем, і І-АБО дерево побудоване за правилами заданими на рисунку 2 матиме наступний вигляд (див.рис.2).

Розроблена інформаційна модель І-АБО дерева дає можливість швидко реалізувати операцію генерування можливих альтернатив структури проєктованого виробу.

### 3. Алгоритм синтезу структурних рішень

Для отримання всіх можливих варіантів структури проєктованого електронного пристрою необхідно здійснити обхід побудованого дерева. В процесі програмної реалізації було вибрано алгоритм обходу вглиб [10]. Цей алгоритм є простим в реалізації, наочно зрозумілим і доволі ефективним при невеликій глибині дерева.

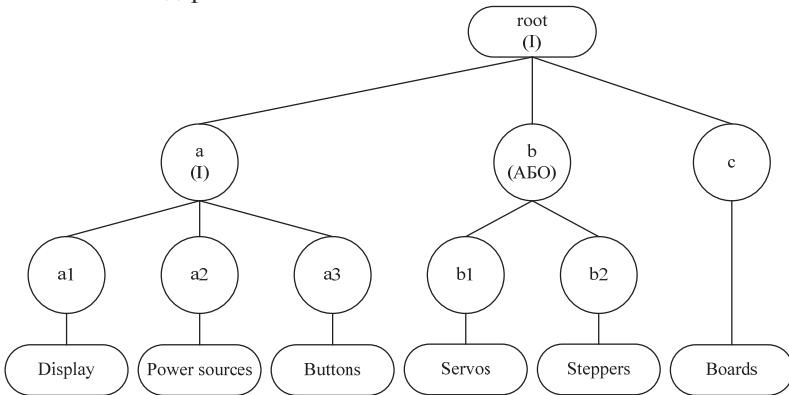


Рис.2. Структура І-АБО дерева

Для даного прикладу обхід буде відбуватися наступним чином:

Крок 1. З вершини "root" переходимо в крайню гілку зліва, і потрапляємо до вершини a.

Крок 2. З вершини a переходимо в крайню ліву гілку, і потрапляємо до вершини a1.

Крок 3. З вершини a1 потрапляємо в вершину "Displays".

Крок 4. Так як вершина "Displays" є кінцевою ми повертаємось вище і вище по ієрархії аж доки не досягнемо вершини a.

Крок 5. З вершини a переходимо середню гілку, звідки потрапляємо в вершину a2 а потім в "Power sources".

Крок 6. Аналогічно відбувається обхід усіх інших вершин даного дерева.

#### 4. Розроблення структур бази даних

В процесі розроблення системи синтезу використано документно-орієнтовану систему керування базами даних MongoDB [11], так як для неї, таблиці не мають зв'язків між собою, та не мають строгої структури полів.

Для структурного синтезу електронних пристроїв створено такі колекції даних як: Boards (керуючі плати), Buttons (кнопки), DC's (мотори змінного струму), Displays (дисплеї), Lights (лампочки), Power sources (джерела струму), Sensors (давачі), Servos (серво-приводи), Shields (плати розширення), Sounds (звукові пристрої), Steppers (крокові двигуни). Всі колекції було наповнено даними з мережі інтернет через програмний модуль написаний на мові Java.

Для кожної колекції розроблено окрему структуру записів.

Структура колекції Boards:

Name; Microcontroller; Operating Voltage (V); Input Voltage (recommended) (V); Input Voltage (limits) (V); Digital I/O Pins; PWM output; Analog Input Pins; DC Current per I/O Pin (mA); DC Current for 3.3V Pin (mA); Flash Memory (kb); SRAM (kb); EEPROM (kb); Clock Speed (MHz); Linux microprocessor; Ethernet; WiFi; USB Type-A; Card Reader (mm); Length (mm); Width (mm); Weight (g).

Структура колекції Buttons: Name; Type; Light Type; Light Color; Number of states.

Структура колекції DCs: Name; Type; Torque (Kg/cm); Speed; Operating voltage (V); Length (mm); Width (mm); Height (mm); Weight (g).

Структура колекції Displays: Name; Type; Screen size (Inch); Resolution (K); Number of Dots; Color; Operating voltage (V); Touchscreen.

Структура колекції Lights: Name; Type; Diameter (mm); Color; Operating voltage.

Структура колекції Power sources: Name; Type; Voltage (V); Portable; Capacity (mAh); Length (mm); Width (mm); Height (mm); Weight (g).

Структура колекції Sensors: Name; Type; Responds to; Analog or digital; Ambient; Operating voltage (V).

Структура колекції Servos: Name; Type; Torque (Kg/cm 5 V); Speed (sec/60 5 V); Rotation Angle (degrees); Operating voltage (V); Length (mm); Width (mm); Height (mm); Weight (g).

Структура колекції Shields: Name; Operating voltage (V); Ethernet; Connection speed (Mbps); WiFi; USB Type-A; Card Reader; GSM (MHz); GPRS; Motor controller; CAN-BUS; Voice recognition.

Структура колекції Sounds: Name; Type; Diameter (mm); Operating voltage (V); Power (W); Impedance (Ohm).

Структура колекції Steppers: Name; Type; Stepper Motor; Step Angle (degrees); Operating voltage (V); Length (mm); Width (mm); Height (mm); Weight (g).

Доступ до даних може здійснюватись також через консоль керування MongoDB. Наповнення колекцій бази даних MongoDB здійснювалось за допомогою програмного модуля доступу до бази даних з використанням

бібліотеки MongoDB для Java.

Щоб приєднатись до сервера бази даних MongoDB було створено об'єкт класу MongoClient з бібліотеки MongoDB для Java, який приймає як параметри адресу хосту сервера БД та порт за яким буде проводитись з'єднання. Для роботи з базою даних було створено об'єкт класу DB, який відповідає за існуючу базу даних.

```
MongoClient mongoClient = new MongoClient("localhost", 27017);  
DB db = mongoClient.getDB("mydb");
```

Розроблена база даних дає можливість ефективно працювати з базовими елементами структур мікроелектронних пристроїв.

## 5. Особливості розроблення системи синтезу та основні результати

Для початку роботи з системою головному вибираємо необхідний файл з правилами побудови I-АБО дерева. Потім виконуємо під'єднання до бази даних MongoDB [11] (рис.3).

На наступному кроці необхідно виконати операцію структурного синтезу, для цього натискаємо на кнопку "Структурний синтез". Результат виконання – записується у файловій системі після натиску кнопки "Синтезувати".

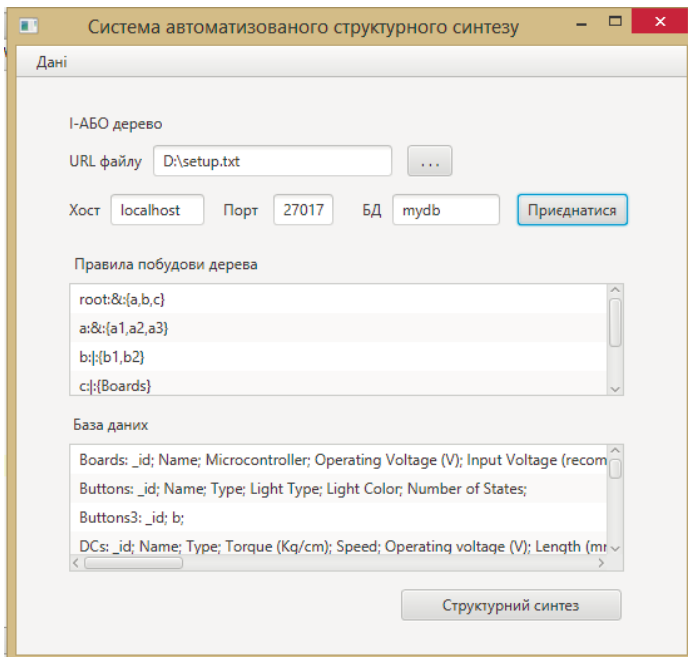


Рис. 3. Головне вікно після під'єднання до бази даних

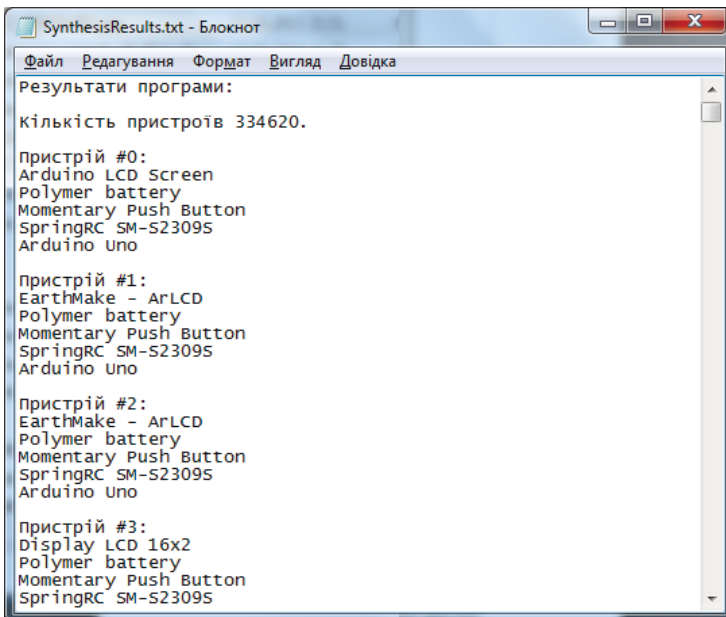
Приклад файлу із згенерованою множиною альтернатив структури електронного пристрою зображено на рис.4.

Отже, розроблена система синтезу структурних схем дає можливість автоматизувати процес структурного синтезу компонентів багаторівневої системи управління технологічними процесами.

На наступному етапі необхідно зменшити потужність множини альтернативних рішень, що можливо досягнути шляхом використання методів багатокритеріальної оптимізації [12].

## Висновки

Розроблено інформаційні моделі структури синтезованої системи та I-АБО дерева на основі використання теорії графів, що дає можливість ефективно організувати опрацювання даних в проєктованій програмній системі.



```
SynthesisResults.txt - Блокнот
Файл Редагування Формат Вигляд Довідка
Результати програми:
Кількість пристроїв 334620.
Пристрій #0:
Arduino LCD Screen
Polymer battery
Momentary Push Button
SpringRC SM-S2309S
Arduino uno
Пристрій #1:
EarthMake - ArLCD
Polymer battery
Momentary Push Button
SpringRC SM-S2309S
Arduino Uno
Пристрій #2:
EarthMake - ArLCD
Polymer battery
Momentary Push Button
SpringRC SM-S2309S
Arduino Uno
Пристрій #3:
Display LCD 16x2
Polymer battery
Momentary Push Button
SpringRC SM-S2309S
```

Рис. 4. Файл зі згенерованою множиною альтернатив структури електронного пристрою

Розроблено базу даних системи синтезу елементів для системи управління енергоефективністю регіону, яка ґрунтується на використанні БД MongoDB і дає тезу можливість ефективно організувати обмін даних між системою синтезу та сховищем даних.

Наведено результати синтезу множини альтернативних рішень, що підтверджує правильність та коректність функціонування розробленої програмної системи синтезу елементів для системи управління енергоефективністю регіону.

1. Катренко А.В. Системний аналіз об'єктів та процесів комп'ютеризації : Навчальний посібник. – Львів: “Новий світ– 2000”.
2. Zwicky F, Wilson A.G. New Methods of Thought and Procedure. –Contributions to the Symposium on Methodologies, Pasadena, May 22-24, 1967.
3. Панкратова Н.Д., Савченко І.О. Застосування методу морфологічного аналізу до задач технологічного передбачення // Наукові праці / Миколаївський держ. гуманітарний ун-т ім. Петра Могили комплексу НаУКМА. Сер. Комп'ютерні технології, системний аналіз, моделювання. — 2008. — 90, вип. 77. — С. 6—13.
4. Бондарева, Г. Г., et al. Метод морфологічного ящика при розробці управлінських рішень // Науковий диспут: питання економіки та фінансів: збірник тез наукових робіт VIII Міжнародної науково-практичної конференції (Київ-Будапешт-Відень, «28» грудня 2016 року / Фінансово-еко-номічна наукова рада, 2016—98 с.
5. Теслюк В.М., Керницький А.Б., Зелінський А.Я., Сергієнко В.Р. Застосування морфологічного методу для синтезу множини альтернативних рішень MEMC // Моделювання та інформаційні технології. Зб. наук. пр. ІППМЕ ім.Г.С.Пухова НАН України. – Київ, 2011, Вип. 62. – С.105 – 112.
6. Ritchey T. Modeling Complex Socio-Technical Systems using Morphological Analysis. – Adapted from an address to the Swedish Parliamentary IT Commission, Stockholm, December 2002. <http://www.swemorph.com/pdf/it-webart.pdf>
7. Ritchey T. Futures Studies using Morphological Analysis. – Adapted from an article for the UN University Millennium Project: Futures Research Methodology Series, 2005. <http://www.swemorph.com/pdf/futures.pdf>
8. Diestel R. Graph Theory, Electronic Edition — NY: Springer-Verlag, 2005. — С. 422.
9. Denysyuk P. XML application for microfluidic devices description / Denysyuk P., Teslyuk V., Khimich I., Farmaga I. // Proceedings of 9th International Conference on the Experience of Designing and Application of CAD Systems in Microelectronics (CADSM'2013), February 19-23, 2013. – Lviv : Lviv Polytechnic Publishing House, 2013. – P. 567-569.
10. Кормен Т. Х. та ін. Частина VI. Алгоритми для роботи з графами // Алгоритми: побудова й аналіз = Introduction to Algorithms — 2-е изд. — М. : Вільямс, 2006. — С. 1296.
11. Shashank T. Professional NoSQL. — Packt Publishing, 2011. — 384 p.
12. Denysyuk P., Teslyuk V., T. A. AlOmari, Teslyuk T. Development and study of subsystem for solution of tasks of multicriterial optimization // Proceedings of the 5th International Conference on Perspective Technologies and Methods in MEMS Design, MEMSTECH 2009, Lviv – Polyana, Ukraine, 2009, pp. 166–167.

Поступила 8.02.2018р.