блокировкой, которая вызвана захватом буфера пакетами или виртуальным каналом. В результате реализуется множество одновременно функционирующих виртуальных безопасных каналов для передачи критичных данных, а также открытых каналов связи для передачи открытой информации без их взаимной блокировки, что повышает эффективность функционирования и защищенность обработки данных в распределенных компьютерных системах.

Список использованных источников

1. *Nelakuditi S.,* Lee S., Yu Y., Zhang Z.-L., and Chuah C.-N. Fast local rerouting for handling transient link failures," IEEE/ACM Trans. Netw. 15(2), pp. 359-372.
2. *Labovitz C.*, Ahuja A., Bose A., and Jahanian F. Delayed internet routing convergence. IEEE/ACM Trans. Networking, 9(3), pp. 293-306.
3. *Yashpaul Singh E.R.*, and Swarup A. Analysis of Equal cost Adaptive Routing Algorithms using Connection-Oriented and Connectionless Protocols. World Academy of Science, Engineering and Technology 51 2009, pp.299-302.
4. *Hansen A. F.*, Cicic T., Gjessing S., Kvalbein A., and Lysne O. Multiple Routing Configurations For Fast IP Network Recovery, IEEE/ACM Trans. Netw. 17 (2), pp. 473-486.

УДК 681.518.5

Є. В. Нікітенко, канд. фіз.-мат. наук
Чернігівський національний технологічний університет, м. Чернігів, Україна

ПРОГРАМНА СИСТЕМА ПОШУКУ НЕСПРАВНОСТЕЙ У СКЛАДНИХ ЕЛЕКТРОННИХ ПРИСТРОЯХ

Е. В. Никитенко, канд. физ.-мат. наук
Черниговский национальный технологический университет, г. Чернигов, Украина

ПРОГРАММНАЯ СИСТЕМА ПОИСКА НЕИСПРАВНОСТЕЙ В СЛОЖНЫХ ЭЛЕКТРОННЫХ УСТРОЙСТВАХ

Yevhen Nikitenko, PhD in Physics and Mathematical Sciences
Chernihiv National Technological University, Chernihiv, Ukraine

SOFTWARE SYSTEM OF TROUBLESHOOTING IN COMPLEX ELECTRONIC DEVICES

Представлено архітектуру модульної системи, що реалізує покращений метод пошуку несправностей. Розглянуто роль кожного модуля у системі, представлена діаграма класів системи.

***Ключові слова**: несправність, функціональний блок, радіоелектронні пристрої.*

Представлена архитектура модульной системы, реализующей улучшенный метод поиска неисправностей. Рассмотрена роль каждого модуля в системе, представлена диаграмма классов системы.

***Ключевые слова:** неисправность, функциональный блок, радиоэлектронные устройства.*

The architecture of modular system implementing an improved method of troubleshooting was presented in the article. The role of each module in the system is examined, the class diagram of the system is represented.

***Key words:** fault, functional block, radioelectronic devices.*

Постановка проблеми. Несправність радіоелектронних пристроїв проявляється у вигляді спотворення вихідної інформації або її відсутності за наявності вхідного сигналу. Джерелом несправності можуть бути один або декілька елементів, а також зовнішні дії і чинники – температура, вологість і т. ін. Кожен функціональний елемент пристрою впливає на формування вихідних параметрів. Залежність між станами елементів і вихідними параметрами є неоднозначною. Більшість елементів впливає відразу на декілька параметрів, а самі параметри можуть залежати від багатьох елементів [1].

Роботу радіоелектронних пристроїв можна оцінювати різними показниками:

- фізичним станом елементів (оцінюється зовнішнім оглядом);
- якістю інформації, що видається;
- формою і значенням напружень у різних точках (оцінюються за показниками вимірювальних приладів).

Починати пошук несправностей необхідно з виявлення істотних протиріч у цих показниках. На визначенні цих протиріч засновані всі методи пошуку несправностей.

Аналіз останніх досліджень і публікацій. Система, що реалізує метод пошуку несправностей у складних електронних пристроях з врахуванням зовнішніх факторів, може бути поділена на взаємодіючі між собою модулі, що реалізують необхідні дії [2]. Така схема взаємодії модулів представлена на рис. 1.

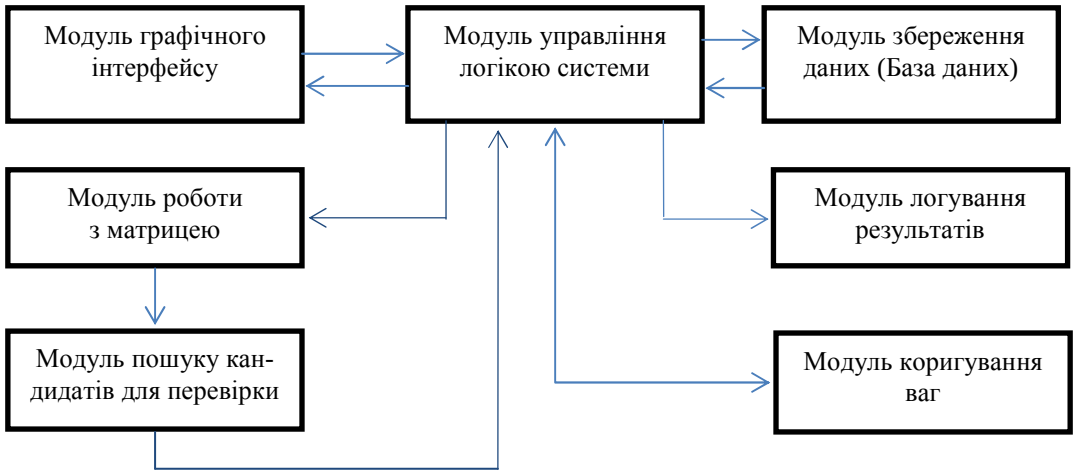

Рис. 1. Схема взаємодії модулів системи, що реалізує метод пошуку несправностей з врахуванням зовнішніх факторів

Модуль графічного інтерфейсу. Модуль графічного інтерфейсу, що зображений на рис. 1, призначений для взаємодії системи з користувачем через введення і відображення інформації. Приклад графічного інтерфейсу представлений на рис. 2.

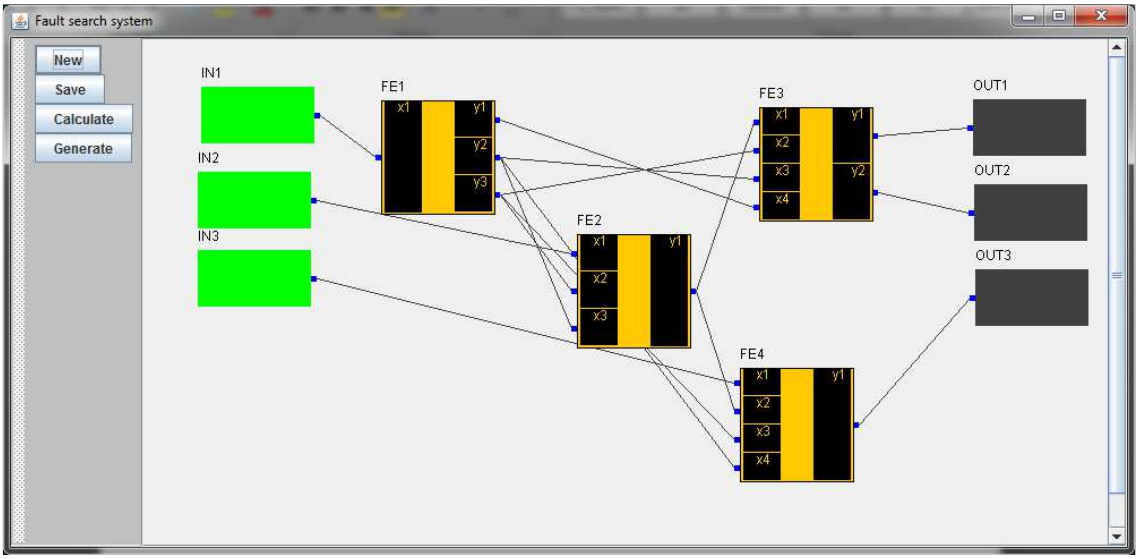

Рис. 2. Приклад графічного інтерфейсу системи, що реалізує метод

Інформація, яку необхідно отримати від користувача на першому етапі, являє собою перелік функціональних елементів, з яких складається пристрій, зв'язки між блоками, а також зв'язки всередині функціональних блоків.

Слід зауважити, що у системі представлено три типи блоків, за допомогою яких можна описати структуру пристрою, а саме:

1) TYPICAL – блок, який представляє сам функціональний елемент, він може мати кінцеве число входів і виходів.

2) IN – блок, який вказує на сигнальні входи електронного пристрою.

3) OUT – блок, який позначає виходи пристрою, що спостерігаються.

Кожному блоку можна задати ім'я. Інтерфейс побудований таким чином, що користувач може переміщати елементи всередині робочої області та розташовувати їх так, як йому зручно.

На наступному кроці від користувача необхідно отримати інформацію про зовнішні чинники, а саме: мінімальне, максимальне і середнє значення для кожного зовнішнього фактора (рис. 3).

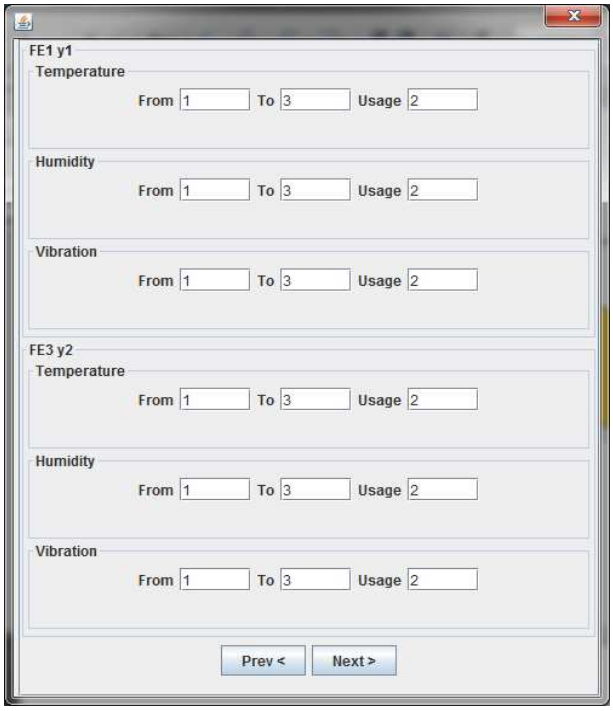

Рис. 3. Приклад вікна введення даних про зовнішні чинники

Ми отримуємо дані про зовнішні чинники не для всіх виводів пристрою, а тільки для тих, які були визначені алгоритмом як потенційно несправні.

Модуль роботи з матрицею. Модуль роботи з матрицею виконує операції, що пов'язані з побудовою і перетвореннями матриці. Він отримує від модуля управління логікою граф у вигляді з'єднаних між собою контактів досліджуваного пристрою і перетворює його в двовимірну матрицю так, як цього вимагає алгоритм для подальшої роботи. Ця операція необхідна, оскільки при отриманні спочатку даних про структуру пристрою від користувача потім набагато простіше будувати граф, ніж відразу матрицю, тому що при будь-якій зміні у структурі виникала б необхідність повного перерахунку матриці, а це досить ресурсномісткий процес. Якщо ж щось змінюється у структурі електронного пристрою, то необхідно внести зміни лише у вузлі, що є набагато простіше (рис. 4).

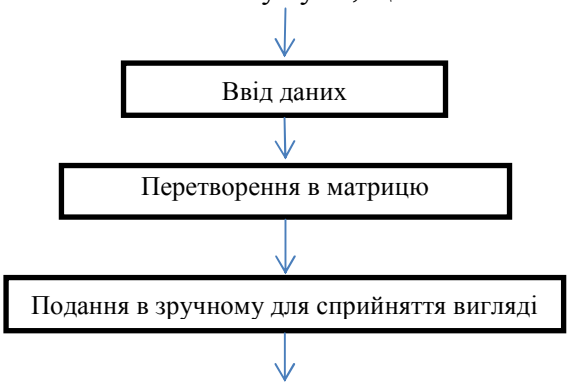

Рис. 4. Модуль роботи з матрицею

Приклад матриці у зручному для сприйняття вигляді представлений на рис. 5. Оскільки матриця досить велика, то на рис. 5 представлена тільки її частина.

	IN1-y1	IN2-y1	OUT1-x1	IN3-y1	FE1-x1	FE1-y1	FE1-y2
IN1-y1	0	0	0	0	1	0	0
IN2-y1	0	0	0	0	0	0	0
OUT1-x1	0	0	0	0	0	0	0
IN3-y1	0	0	0	0	0	0	0
FE1-x1	0	0	0	0	0	1	1
FE1-y1	0	0	0	0	0	0	0
FE1-y2	0	0	0	0	0	0	0
FE1-y3	0	0	0	0	0	0	0
FE2-x1	0	0	0	0	0	0	0
FE2-x2	0	0	0	0	0	0	0
FE2-x3	0	0	0	0	0	0	0
FE2-y1	0	0	0	0	0	0	0
FE3-x1	0	0	0	0	0	0	0
FE3-x2	0	0	0	0	0	0	0
FE3-x3	0	0	0	0	0	0	0
FE3-x4	0	0	0	0	0	0	0
FE3-y1	0	0	1	0	0	0	0
FE3-y2	0	0	0	0	0	0	0
FE4-x1	0	0	0	0	0	0	0
FE4-x2	0	0	0	0	0	0	0
FE4-x3	0	0	0	0	0	0	0
FE4-x4	0	0	0	0	0	0	0
FE4-y1	0	0	0	0	0	0	0
OUT2-x1	0	0	0	0	0	0	0

Рис. 5. Приклад подання графа зв'язків у вигляді квадратної матриці

Ще одним завданням цього модуля є підрахунок ваг для всіх виводів пристрою з використанням базового алгоритму. Визначення ваг відбувається по квадратній матриці.

Модуль пошуку кандидатів на перевірку. Цей модуль проводить аналіз всіх можливих шляхів передачі сигналів у пристрої і обчислює з усієї безлічі тільки ті, які слід перевірити при визначенні несправності на одному з виходів пристрою.

Оскільки у базовому алгоритмі зазначено тільки формальний опис цього етапу без варіантів його реалізації, то (як найбільш ефективний) був обраний метод рекурсивного обходу графа [3] (рис. 6).

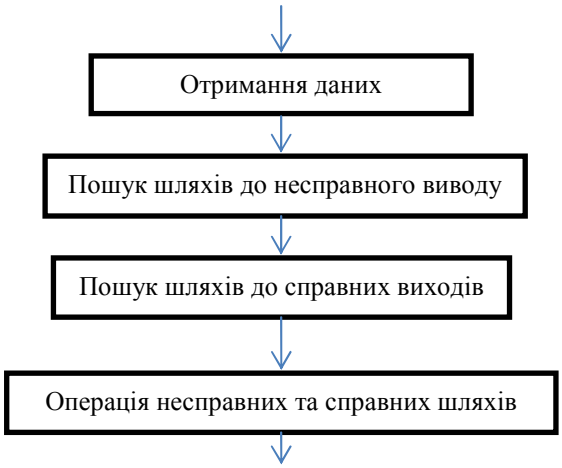

Рис. 6. Модуль пошуку кандидатів на перевірку

Модуль управління логікою системи. Центральним модулем є модуль управління логікою системи. Розглянемо основні операції, які виконує цей модуль. Першим його завданням є отримання даних від модуля графічного інтерфейсу і побудова у пам'яті графа зі зв'язками. Після того, як граф побудований і визначені необхідні дані, він зберігається в базі даних (БД) для того, щоб можна було відновити його в будь-який момент часу. Як сховище даних була використана база даних MongoDB [4].

Вибір зупинений саме на цій БД, тому що вона дуже гнучка і дозволяє зберігати різнорідні дані в одній колекції (аналог таблиці в MySQL). Якщо в майбутньому виникає необхідність у додаванні нових типів вузлів, то це не завдасть великих труднощів.

Наступним кроком роботи модуля є передача побудованого графа модулю роботи з матрицею. Всі необхідні дії, що представляють інтерес, відображаються у вікні логів системи.

Отримавши підраховані на основі графа потенційно несправні виводи, модуль передає дані на графічний інтерфейс для відображення результатів.

Модуль графічного інтерфейсу повертає дані про зовнішні чинники для конкретних виводів, після чого модуль управління логікою пропускає цю інформацію через модуль коригування ваг і повертає остаточний результат алгоритму на графічний інтерфейс для відображення його користувачеві.

Діаграма класів системи, що реалізує алгоритм. На рис. 7 представлена діаграма основних класів системи, що реалізує розроблений алгоритм.

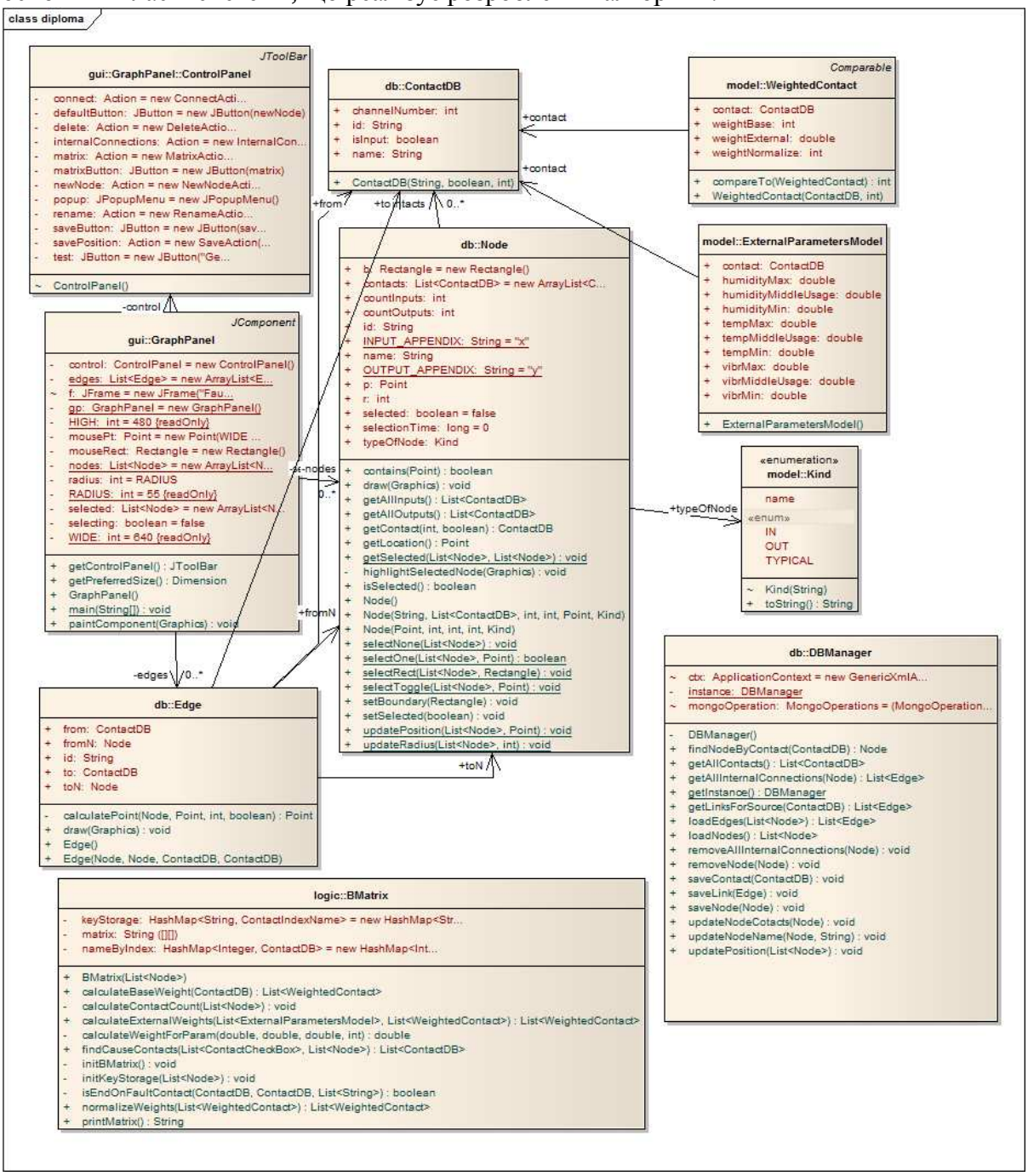

Рис. 7. Діаграма класів системи

Клас GraphPanel є основним і виконує роль модуля управління логікою системи. Його основні завдання – це оброблення подій, що прийшли від графічного інтерфейсу, і виконання відповідних дій, які зумовлені для конкретних подій.

Клас ControlPanel – це панель контролю на графічному інтерфейсі. Користувач його бачить, як панель з кнопками. На ньому розташовані кнопки:

1) New – створення нового блока у робочій області;
2) Save – збереження стану системи в БД;
3) Calculate – викликає ряд послідовних діалогів для введення даних від користувача;
4) Generate – генерація випадкового графа за заданими параметрами.

Клас DBManager – це клас взаємодії з базою даних, він надає операції зчитування, запису і модифікації елементів системи.

Клас BMatrix – це клас роботи з матрицею, а саме перетворення графа в матричний вигляд і підрахунок ваг згідно з базовим методом.

У базі граф представляється як композиція з об’єктів трьох класів: Node, ContactDB, Edge. Клас Node – це функціональний елемент, який зберігає дані про вузол і його контакти. Під контактами розуміємо виводи функціонального елемента, які представлені класом ContactDB. Саме між контактами і будуються всі причинно-наслідкові зв’язки у пристрої. Зв’язки представлені у класі Edge.

Kind – це перерахування будь-яких типів елементів системи: TYPICAL, IN, OUT.

Висновки. У результаті проведеної роботи розроблено алгоритм і програмне забезпечення для пошуку несправностей у складних пристроях з урахуванням зовнішніх факторів, що дозволяє визначати несправний функціональний блок у несправному електронному пристрої. Представлена архітектура модульної системи, що реалізує покращений метод пошуку несправностей. Розглянуто роль кожного модуля у системі та представлена діаграма класів системи.

Список використаних джерел

1. *Конструкторско-технологическое* проектирование электронной аппаратуры : учебник для вузов. – М. : Изд. МГТУ им. Н. Э. Баумана, 2002. – 528 с.

2. *Об одном* варианте решения технического диагностирования радиоэлектронных средств / А. В. Дубов, А. П. Капранов, В. В. Сускин, В. Ф. Шевченко // Управление большими системами: надежность и диагностика средств и систем управления. – 2010. – № 31. – С. 363-377.

3. *Лекции* по теории графов / В. А. Емеличев, О. И. Мельников, В. И. Сарванов, Р. И. Тышкевич. – Изд. 2-е, испр. – М. : УРСС, 2009. – 392 с.

4. *Тим Хоукинс*, Илько Пладж, Питер Мембри. The Definitive Guide to MongoDB: The NoSQL Database for Cloud and Desktop Computing. – М. : Apress, 2010. – С. 350.