

## Література

1. Аблесимов А. К. Роль и перспективы развития современных цифровых промышленных сетей для автоматизированной системы управления технологическим процессом / А. К. Аблесимов, Е. А. Жилияков, А. М. Иванова // *Електроніка та системи управління*. – 2009. – № 4 (22). – С. 47.
2. Майская В. Протокол CANopen. Дальнейшее развитие CAN [Электронный ресурс]: *Электроника: Наука, Технология, Бизнес* – 2006 – № 3. – С. 18–22. – Режим доступа : <http://www.electronics.ru/issue/2006/3/2>.
3. CAN Specification 2.0B. Robert Bosch GmbH, Postfach 30 02 40, D-70442 Stuttgart, 1991. <http://www.semiconductors.bosch.de/pdf/can2spec.pdf>
4. ISO/IEC 7498-1: 1994. Information technology – Open Systems Interconnection – Basic reference model: The basic model. International Organization for Standardization, 1994.
5. Вознесенский А. Н. CAN-технологии для транспортных систем / А. Н. Вознесенский, А. С. Чепурнов, И. В. Грибов. *Электроника и электрооборудование транспорта*. – 2006. – № 5. – С. 1–5.
6. Третьяков С.А. CAN на пороге нового столетия // *Мир компьютерной автоматизации*. – 1999. – № 2. – С. 14–18.
7. CAN Implementation // CAN in Automation. Интернет: <http://www.can-cia.org>.
8. CAN марафон // Введение в протокол CAN. Интернет: <http://can.marathon.ru/can-protocols/canbus/canintro>.

Надійшла 20.8.2011 р.

УДК 681.3.06

Р.П. ГРАФОВ, К.П. ФЕДІВ, М.О. КОЛОСОВСЬКИЙ  
Хмельницький національний університет

## АНАЛІЗ НАДІЙНОСТІ СКЛАДНИХ ПРОГРАМНИХ КОМПЛЕКСІВ

*У роботі наведені моделі та методи для визначення надійності складних програмних комплексів, що містять множини незалежних програмних модулів. Розглядається пряма і обернена задачі, що дозволяють враховувати обмеження на час налагодження та тестування програм і оцінювати якість програмного забезпечення залежно від кваліфікації розробника.*

*This study presents models and methods for determining the reliability of complex software systems that contain a set of independent software modules. The direct and inverse problems, allowing to take into account the time limit for debugging and testing programs and to assess the quality of software, depending on qualifications developer.*

Ключові слова: програмне забезпечення, складні програмні комплекси, стохастична матриця, функція Лагранжа.

## Вступ

Відповідно до міжнародних стандартів ISO 9126: 1991 виділено ряд основних характеристик, які дозволяють оцінювати програмні засоби (ПЗ) з позиції користувача, розробника і керівника проектом. Близьким до зазначеного стандарту за ідеологією, структурою і змістом є ГОСТ 28195-89, на верхньому рівні якого виділені показники якості ПЗ: надійність, коректність, зручність застосування, ефективність, універсальність і супроводжуваність.

Серед наведених показників надійність займає особливе місце, тому що надійна програма повинна забезпечувати мінімальну ймовірність відмови у процесі функціонування, швидке реагування на помилки даних або обчислювального процесу і відбудови працездатності, при якому не фіксується відмова. Як основний показник якості надійність є динамічним параметром, присутнім протягом всього часу функціонування системи.

Розвиток засобів обчислювальної техніки і ускладнення прикладних задач у різних галузях науки і техніки призводить до суттєвого зростання обсягу та складності алгоритмів програмного забезпечення, що пов'язаний з підвищенням вірогідності помилок у використовуваних програмних засобах. Це призводить до зниження продуктивності завдань, для яких розроблено ПЗ. Тому до надійності ПЗ, як основного показника якості програмних продуктів, пред'являються високі вимоги [1, 2].

Інформаційні системи функціонують в умовах дії зовнішніх і внутрішніх факторів, здатних порушити нормальну роботу системи. Гарантоздатність як властивість системи, повинна протистояти цим чинникам. Програмне забезпечення є однією з найголовніших складових системи, що забезпечує її гарантоздатність. Під надійністю ПЗ розуміють здатність виконувати всі поставлені функції при заданих умовах протягом заданого періоду функціонування системи [3].

## Постановка задач

Вибір оптимальної моделі для оцінки надійності ПЗ для практичних цілей на всьому протязі життєвого циклу являється не до кінця вирішеним завданням. Отримання такої оцінки є необхідним і

актуальним для уникнення будь-яких ризиків (фінансових) та втрат, пов'язаних з помилками ПЗ, а також для ефективного планування роботи інформаційних систем, які інтенсивно розробляються в наш час. Аналіз надійності функціонування ПЗ здійснюється, як правило, до здачі в експлуатацію на етапі налагодження або тестування. При цьому для розрахунків часто використовується час, що витрачається на налагодження і тестування. Вимоги до значених параметрів можуть бути сформовані з урахуванням специфіки і структури ПЗ. Актуальність вирішення подібних завдань впливає з того, що жоден серйозний проект, пов'язаний з вирішенням комплексу важливих і відповідальних завдань, не може бути зданий в експлуатацію, без оцінки надійності ПЗ відповідно до запропонованих вимог.

Під складним програмним комплексом СПК будемо розуміти складну систему, що містить множини незалежних програмних модулів ПМ, що функціонують відповідно з принципом побудови СПК і своїх власних алгоритмів. У роботі розглядаються наступні завдання, що мають практичне значення:

1. Оцінка надійності СПК, що полягає в знаходженні ймовірності безпомилкового функціонування СПК, при відомій структурній схемі СПК та ймовірності безпомилкового рішення завдань усіх ПМ;
2. Мінімізація часу на налагодження і тестування при досягненні заданого рівня надійності СВК.
3. Знаходження екстремальних значень показника надійності СВК за обмежений проміжок часу налагодження і тестування.

#### Моделі і методи розв'язання задачі

Програмний комплекс СПК це складна система, яка містить сукупність менш складних складових ПМ, пов'язаних між собою таким чином, який визначається структурою і специфікою СПК. Для дослідження надійності СПК в цілому попередньо розглянемо модель надійності окремих програмних модулів ПМ. Розрахункові та прогнозові характеристики програмних засобів гарантоздатних інформаційних систем базуються на використанні множини математичних моделей, розробка яких почалася в 70-х роках минулого століття в умовах інтенсивного впровадження економіко-математичних методів у сферу виробництва та управління. Кожна з розроблених моделей оцінки надійності ПМ ґрунтується на системі припущень залежно від умов функціонування, середовища експлуатації, процесів коригування програмних помилок. Із множини існуючих моделей найбільш прийнятна для вирішення поставлених завдань, є експоненціальна модель [4]. Дана модель враховує час простою і виправлення помилок. У процесі усунення помилок дана модель враховує реальний час роботи ПМ.

Згідно з даною моделлю ймовірність безпомилкової роботи  $i$ -го ПМ можна представити наступною формулою:

$$P_i(t_i) = e^{-I_i t_i} e^{-n_i t_i} \quad (1)$$

де  $I_i$  – інтенсивність появи помилки при роботі ПМ;  $n_i$  – інтенсивність налагодження;  $t_i$  і  $\bar{t}_i$  – час обчислень і налагодження модуля;  $I = 1/T_i$ ;  $T_i$  – початковий середній час безпомилкової роботи модуля;  $n_i = K_i / (N_{oui} T_i)$ ;  $K_i$  – коефіцієнт стиснення часу налагодження (тестування) у порівнянні з часом обчислень;  $N_{oui}$  – початкове (передбачуване) число помилок в ПМ. У виразі (1) для оцінки надійності ПМ присутні також налагоджувальні параметри і коефіцієнт  $K_i$ , що дозволяє врахувати рівень кваліфікації розробника.

В якості моделі функціонування та оцінки надійності СПК, що містить множини незалежних модулів, використовуємо Марківський процес з дискретним станом і дискретним часом. Для такого процесу моменти  $t_1, t_2, \dots$ , коли система  $S$  змінює свій стан, розглядаються як послідовність кроків, а в якості аргументу, від якого залежить процес, розглядається номер кроку: 1, 2, 3,.... Випадковий процес в цьому випадку характеризується послідовністю станів  $S_0, S_2, \dots, S_n$ , які створюють Марківський ланцюг. При цьому  $S_0$  – початковий стан системи. Стани ланцюга Маркова визначаються ймовірностями  $P_i(k)$  виклику  $i$ -го програмного модуля на  $k$ -му кроці. Очевидно, що для будь-якого  $k$  виконується умова:

$$\sum_{i=1}^n P_i(k) = 1 \quad (2)$$

Зі стану  $i$  в стан  $j$  на  $k$ -му кроці система переходить з вірогідностями  $p_{ij}$ , які утворюють квадратну матрицю  $|p_{ij}|$  розміру  $(n \times n)$ , названу стохастичною матрицею ймовірностей перехідних станів. Сума перехідних ймовірностей в будь-якому рядку матриці дорівнює одиниці.

$$\sum_{i=1}^n p_{ij} = 1 (i = 1, \dots, n) \quad (3)$$

Використовуючи властивості стохастичної матриці (2), (3), можна визначити ймовірність стану системи на  $k$ -му кроці за рекурентною формулою:

$$P_i(k) = \sum_j P_j(k-1) p_{ji}; (i, j = 1, \dots, n)$$

Розглянемо найпростіший СПК, що складається з  $N$  окремих модулів, з'єднаних між собою у відповідності з алгоритмами обробки даних і вирішення завдань. Нехай відомі ймовірності переходів між ПМ. Побудуємо стохастичний граф  $G(t)$ , що містить  $N + 2$  вершин (програмних модулів). Початкова

вершина  $N+1$  означає початковий стан, а вершина  $N$  – кінцевий стан графа. Кожен ПМ вирішує своє завдання у відповідності з алгоритмом досягнення цільової функції. Вершини  $0$ -а і  $(N+1)$ -ий є фіктивними, час знаходження в них дорівнює нулю, а ймовірності безпомилкової роботи – одиниці. Фіктивні вершини вводяться в модель з метою коректного відображення початкового стану і динаміки функціонування у часі всіх ПМ. При цьому передбачається, що модулі статистично незалежні. На рис. 1 показаний стохастичний граф з числом ПМ, рівним  $N=5$ .

Вершини графа – програмні модулі, а значення  $P$  (без аргументу для спрощення) – ймовірності виклику ПМ, перехідні ймовірності  $p_{ij}$  визначають послідовність виклику програмних модулів. Як елементи матриці (див. рис. 1) графа  $G$  використовуємо добутки ймовірностей переходу  $p_{ij}$   $i$ -го ПМ і  $j$ -го і ймовірностей безвідмовного функціонування  $P_i(t_i)$   $i$ -го ПМ протягом часу  $t_i$  ( $i, j = 0, \dots, (N+1)$ ). Якщо система перейшла з  $i$ -го в стану в стан  $i+1$ , то це означає що ПМ, відповідний  $i$ -му стану, відпрацював нормально без помилок, в іншому випадку такий перехід не відбувся б.

Графу  $G(t)$  СПК відповідає стохастична матриця ймовірностей переходів від одного ПМ до іншого, рис. 2.

Використовуючи структурні перетворення над даною матрицею шляхом послідовного піднесення до степеня, можна знайти ймовірність безвідмовної роботи комплексу за певну кількість кроків, включаючи останній крок, на якому завершується процес вирішення завдання. Щоб знайти зазначену ймовірність за два кроки потрібно підсумувати добутки вірогідності по всіх шляхах, що з'єднає дві вершини шляхом зведення матриці в квадрат і т.д. Виконуючи цю операцію над  $G$  при  $k \leq k_{max}$ , можна визначити всі зв'язки між вершинами. Матриця  $G^k$  (макс) називається характеристичною (матрицею всіх шляхів між вершинами), де  $k_{max}$  таке, що подальше піднесення матриці до степені не змінює її вхіджень, тобто  $k_{max} \leq N-1$ , тому що максимальний ранг шляху не може перевищувати значення  $N-1$ .

Завдання 1. Знаходження ймовірності безвідмовної роботи СПК  $P(t)$ . Використовуючи матрицю  $G(t)$  побудуємо матрицю  $A$ , де  $E$ -одична матриця [5].

$$A = E + G(t) + G(t)^2 + G(t)^3 + \dots = E(E - G(t))^{-1}$$

Елемент матриці  $A$  з номером  $(0, N+1)$  використовується для визначення ймовірності безпомилкової роботи всього СПК, тому що враховує всі послідовності викликів окремих ПМ. Використовуючи правила обчислення оберненої матриці, вираз для ймовірності безпомилкової роботи СПК з урахуванням всіх ПМ, можна представити у вигляді [6]:

$$P(t) = D(t) / R(t), \tag{4}$$

де  $D(t)$  – алгебраїчне доповнення елемента з номером  $(N+1, 0)$  матриці  $(E-G(t))$ ;  $R(t)$  – головний визначник матриці  $(E-G(t))$ .

Виконавши перетворення у відповідність з (4) і припустивши, що у виразі (1) для оцінки надійності окремих ПМ час налагодження = 0, отримаємо шукану ймовірність  $P(t)$ .

Завдання 2. Визначення мінімального часу налагодження СПК при заданому значенні  $\geq P$  зад. Мінімальний час визначимо як:

$$t = \sum_{i=0}^{N+1} t_i$$

Для вирішення завдання використаємо метод невизначених множників Лагранжа, . Запишемо функцію Лагранжа:

$$f_1(t, t, x) = \sum_{i=0}^{N+1} t_i + x(P(t, t) - P_{зад}), \tag{5}$$

де  $x$  – множник Лагранжа.

Диференціюючи (5) по  $t_i$  і  $x$  і прирівнявши отримані вирази нулю, отримаємо систему рівнянь:

$$\begin{aligned} \partial f_1(t, t, x) / \partial t_i &= 0, \\ P(t, t) - P_{зад} &= 0 \end{aligned} \tag{6}$$

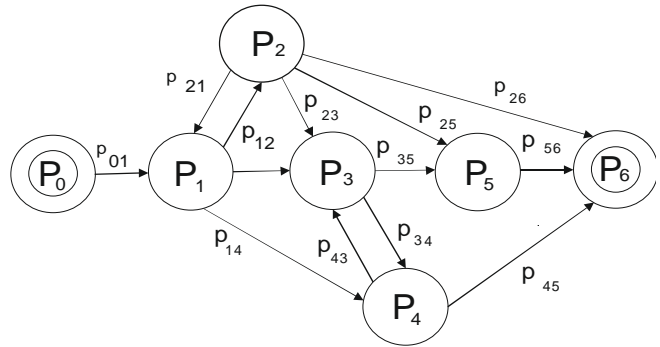


Рис. 1. Стохастичний граф  $G(t)$

$$G(t) = \begin{vmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & p_{12}P_1 & p_{13}P_1 & 0 & 0 & 0 \\ 0 & p_{21}P_2 & 0 & p_{23}P_2 & 0 & p_{25}P_2 & p_{26}P_2 \\ 0 & 0 & 0 & 0 & p_{34}P_3 & p_{35}P_3 & 0 \\ 0 & 0 & 0 & p_{43}P_4 & 0 & 0 & p_{46}P_4 \\ 0 & 0 & 0 & 0 & 0 & 0 & p_{56}P_5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{vmatrix}$$

Рис. 2. Матриця ймовірностей перехідних станів

Прийнявши  $t_i = t_i^0$  і вирішивши друге рівняння системи (6), визначаємо необхідний час:  $t^0$ :

$$t^0 = \sum_{i=0}^{N+1} t_i^0$$

Завдання 3. Визначення максимуму ймовірності безвідмовної роботи комплексу  $P(t, t)$  при заданому часі налагодження. Запишемо для цього випадку функцію Лагранжа:

$$f_2(t, t, w) = P(t, t) + w \left( \sum_{i=0}^{N+1} t_i - t_{зад} \right) \quad (7)$$

Диференціюючи (7) по  $t_i$  і  $w$  і прирівнявши отримані вирази нулю, отримуємо систему рівнянь:

$$\partial f_2(t, t, w) / \partial t_i = 0,$$

$$\sum_{i=0}^{N+1} t_i - t_{зад} = 0$$

Вирішивши друге рівняння системи, знаходяться значення  $t_i^0$ , що підставляються в  $P(t, t)$  для визначення максимальної надійності СПК.

#### Висновки

Розглянуті в роботі моделі дозволяють робити оцінку надійності складних програмних комплексів при відомих показниках надійності складових модулів і експериментально отриманої стохастичної матриці перехідних станів. Вирішення обернених задач дозволяє проводити розрахунок надійності з урахуванням обмежень на ресурси, що залежать від кваліфікації розробника програмного забезпечення. Наведений метод може бути використаний для аналізу та оцінки надійності програмних комплексів в цілому, а також для дослідження надійності комплексу на початковому етапі розробки, що включає етапи налагодження і тестування.

#### Література

1. Липаев В.В. Качество программного обеспечения / В.В. Липаев. – М.: Финансы и статистика, 1983. – 261 с.
2. Карповский Е.Я. Надежность программной продукции / Е.Я. Карповский, С.А. Чижов. – Киев: Техника, 1990. – 160с.
3. Авижение А.Н. Гарантоспособные вычисления: от идей до реализации в проектах / А.Н. Авижение. – ТИИЭР, 1986. – Т.74. – № 5. – с. 8-21.
4. Musa J.D. Validity of Execution time theory of software reliability // IEEE Trans. On reliability. – 1979. – v 3. – P.199-205.
5. Баглюк С.И. Надежность функционирования программного обеспечения / Баглюк С.И., Мальцев М.Г., Смагин В.А., Филимоныхин Г.В. – С. – Пб.: -1991. – 78с.
6. Гантмахер Ф.Р. Теория матриц / Ф.Р. Гантмахер. – М.: Наука, 1988. – 548 с.

Надійшла 9.8.2011 р.

УДК 681.327

Д.Н. МОАМАР, В.Г. РЯБЦЕВ, Т.Ю. УТКИНА

Черкасский государственный технологический университет

### МЕТОД ОЦЕНКИ ДИАГНОСТИЧЕСКИХ СВОЙСТВ ТЕСТОВ МИКРОСХЕМ ПАМ'ЯТИ

*В статті розглянуті питання діагностування несправностей мікросхем пам'яті. Наведені результати моделювання несправностей мікросхем пам'яті за допомогою моделей, представлених у вигляді графів цифрових автоматів зі скінченим числом станів. Проведено аналіз особливостей алгоритмів тестів мікросхем пам'яті, визначені діагностичні властивості тестів. На основі отриманих результатів розроблено метод оцінки діагностичних властивостей тестів мікросхем пам'яті.*

*The questions of fault diagnosis of memory chips are considered in the article. The results of fault simulation of memory chips using the models presented in the form of graphs of digital machines with a finite number of states. The analysis of the features of algorithms tests of memory chips is carried out; the diagnostic properties of tests are defined. On the basis of the received results the method of an estimation of the diagnostic properties of tests of memory chips is developed.*

Ключові слова: мікросхеми пам'яті, несправності, діагностичні властивості тестів.

#### Введение

Имеется несколько очень веских причин, которые объясняют, почему методы диагностирования микросхем памяти заслуживают особого внимания. Во-первых, микросхемы памяти являются жизненно