УДК 004.052

В. С. Яковина, канд. фіз.-мат. наук

КОМПОНЕНТНІ МОДЕЛІ НАДІЙНОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВИЩОГО ПОРЯДКУ

Анотація. *В роботі представлено дві моделі надійності програмного забезпечення на основі ланцюгів Маркова вищого порядку з неперервним часом. Одна з моделей враховує не тільки відмови компонент програмного продукту, але й імовірності відмови інтерфейсів при передачі потоку управління в програмній системі.*

Ключові слова: *програмне забезпечення, надійність, модель, модульна система, ланцюг Маркова вищого порядку, програмний інтерфейс, відмова, інтенсивність відмов, помилка, граф стану системи*

V. Yakovyna, PhD.

COMPONENT BASED HIGH-ORDER SOFTWARE RELIABILITY MODELS

Abstract. *Two software reliability models based on high-order continuous-time Markov chains are presented in the paper. One of the models takes into account not only components failures, but also the probabilities of components interfaces failures during control flow process within software system.*

Keywords: *software, reliability, model, modular system, high-order Markov chain, software interface, failure, failure intensity, defect, system state graph*

В. С. Яковина, канд. физ.-мат. наук

КОМПОНЕНТНЫЕ МОДЕЛИ НАДЕЖНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ВЫСШЕГО ПОРЯДКА

Аннотация. *В работе представлены две модели надежности программного обеспечения с использованием цепей Маркова высших порядков с непрерывным временем. Одна из моделей учитывает не только отказы компонент программного продукта, но и вероятности отказов интерфейсов при передаче потока управления в программной системе.*

Ключевые слова: *программное обеспечение, надежность, модель, модульная система, цепь Маркова высшего порядка, программный интерфейс, отказ, интенсивность отказов, ошибка, граф состояния системы*

Вступ

Розвиток індустрії розробки програмного забезпечення (ПЗ) відбувався в останні десятиліття безпрецедентними темпами. Вартість апаратного забезпечення і, зокрема, пам'яті та обчислювальних потужностей невпинно зменшується; Інтернет та хмарні технології перетворили світ в єдину комп'ютерну мережу, в якій комунікації, інформація та бізнес-процеси постійно тривають он-лайн. Всі ці технологічні зміни обумовили зміни індустрії розробки програмного забезпечення. Сучасне програмне забезпечення вже не може бути створене від початку до кінця талановитими одинаками. Натомість десятки, а іноді тисячі програмістів здійснюють спільну командну розробку проектів за створення програмних продуктів, які складаються з тисяч та мільйонів рядків програмного коду.

Не дивно, що за таких умов контроль якості програмної продукції значно ускладнюється. Забезпечення якості продуктів, створених різними командами розробників з подальшою інтеграцією в єдину програмну систему, є нетривіальною задачею. Одним з методів вирішення такої задачі є використання відповідних методологій проектування ПЗ. Програмна інженерія пропонує різні методики забезпечення та підвищення якості програмних продуктів. Однак, навіть з використанням кращих практик проектування та програмування командами досвідчених розробників, програмний код може містити помилки, які при виконанні цього коду приведуть до відмов системи.

Емпіричний досвід розробки ПЗ свідчить, що на кожну тисячу рядків коду припадає декілька помилок навіть при розробці найдосвідченішими програмістами [1].

Разом з тим, темпи розвитку методів і засобів аналізу та оцінювання надійності програмного забезпечення в останні десятиліття не відповідали темпам розвитку індустрії ПЗ, і до тепер часто використовуються моделі, розроблені в 1970-х роках, які трактують програмну систему як чорну скриньку. Отже розроблення моделей і методів аналізу надійності ПЗ з урахуванням його структури та складності є важливою науково-практичною задачею.

Сучасний стан проблеми

Моделі надійності ПЗ можуть бути розділені на два великі класи – детерміністичні (статичні) та імовірнісні (динамічні) [2 – 4].

Вимірювання продуктивності детерміністичного типу отримується з аналізу тексту вихідних кодів програми. До детерміністичного класу відносять дві моделі [3 – 4]: модель Halstead та модель цикломатичної складності McCabe. Моделі цього класу використовують для дослідження елементів програми шляхом підрахунку кількості операторів, операндів та інструкцій; потоку управління програми шляхом підрахунку розгалужень та трасування шляхів виконання; потоку даних програми шляхом вивчення спільного використання даних та передачі даних.

Імовірнісні моделі представляють появу відмов та усунення дефектів як випадкові події. Згідно [3] імовірнісні моделі можуть бути поділені на різні типи: (а) висівання помилок, (б) інтенсивності відмов,

(в) апроксимації залежностей, (г) зростання надійності, (д) архітектурного підходу, (е) області вхідних даних, (ж) шляху виконання програми, (з) неоднорідного пуассонового процесу, (и) марковські, (і) баєсівські та уніфіковані.

Починаючи з 1990-х років все більшого поширення набувають компонентні моделі, які описують надійність програмного продукту з урахуванням його внутрішньої будови. У моделях такого типу надійність системи є функцією надійності її модулів (логічних частин ПЗ, які можна тестувати та реалізовувати незалежно від інших). У свою чергу компонентні моделі поділяються на адитивні моделі, моделі розроблені на основі шляхів виконання ПЗ та на основі архітектурного підходу [5].

Моделі оцінювання надійності ПЗ на основі архітектурного підходу у більшості випадків використовують теорію класичних марковських процесів, припускаючи незалежність виконання компонент програмної системи, що є спрощеним описом реального процесу виконання ПЗ, де імовірність переходу в наступний стан (наприклад передачі потоку управління до іншого модуля програми) може залежати не тільки від поточного стану, а й від передісторії потрапляння в цей стан. У [6] для усунення такого спрощення запропоновано використовувати марковський процес вищого порядку, який дає змогу точніше аналізувати поведінку програмного продукту.

У [7] розглядається модель оцінювання надійності програмного забезпечення, в якій архітектуру ПЗ подано марковським процесом вищих порядків з дискретним часом. Разом з тим, використання моделі надійності на основі ланцюгів Маркова з дискретним часом має ряд обмежень та спрощень. Так, ця модель враховує тільки сумарний час виконання i -ї компоненти та її інтенсивність відмов, не враховуючи розподіл часу виконання компоненти та конкретні моменти часу, в які відбувалась передача потоку управління цьому модулю. Іншим спрощенням є те, що в реальних програмних продуктах передача управління між модулями системи відбувається в довільні моменти часу (причому самі моменти часу, в які відбувається передача управління є випадковими величинами). Тому, для підвищення ступеня адекватності моделі надійності ПЗ, що враховує його архітектуру, слід використовувати ланцюги Маркова з неперервним часом.

Опис розроблених моделей надійності ПЗ

Модель 1

Модель надійності ПЗ на основі ланцюгів Маркова вищого порядку з неперервним часом представляється як (C_i, **A**, **P**, λ_i), де $\{C_i\}$ – напрямлений граф, який відображає структуру програмної системи, вершини графа відповідають компонентам системи (програмним модулям), а ребра відображають передачу потоку управління між модулями; $\mathbf{A} = \{a_{ij}(t)\}$ – матриця інтенсивності переходів між станами системи, які відповідають вершинам графа $\{C_i\}$; $\mathbf{P} = \{p_i(t)\}$ вектор імовірностей перебування системи в стані C_i в момент часу t; $\lambda_i(t)$ – інтенсивність відмов i-ої програмної компоненти.

Основними припущеннями та властивостями цієї моделі є:

- програмний продукт складається зі скінченної кількості модулів, які з точки зору діаграми надійності з'єднані в систему послідовно;
- передача управління між модулями програми відбувається в довільні моменти часу;
- кожен модуль програмної системи може бути протестований незалежно і характеризується інтенсивністю відмов $\lambda_i(t)$;
- операційний профіль програми відповідає протестованим варіантам використання системи і не змінюється з часом;
- інтенсивність переходів a_{ij} між станами i та j залежить від історії переходів між попередніми станами системи, яка закінчується поточним станом i (кількість попередніх станів, що впливають на інтенсивність a_{ij} дорівнює порядку марковського процесу K);
- надійність i-го модуля залежить від конкретних моментів часу, в які було передано управління цьому модулю (у випадку, якщо $\lambda_i = f(t)$);
- в довільний момент часу t відмова програмної системи може бути спричинена виключно відмовою того модуля який виконується в даний момент часу;
- модель є ієрархічною, тобто спочатку обчислюються параметри архітектури ПЗ, а далі враховується поведінка відмов кожного модуля.

В такому випадку інтенсивність відмов програмної системи, яка складається з N модулів може бути записана як

$$\lambda(t) = \sum_{i=1}^{N} p_i(t)\lambda_i(t), \qquad (1)$$

де $\lambda_i(t)$ – інтенсивність відмов i-го модуля; $p_i(t)$ – імовірність виконання i-го модуля в момент часу t.

Як і у випадку моделі з дискретним часом [7] інтенсивність відмов модулів програмної системи може бути отримана на основі результатів модульного тестування з використанням відомих моделей надійності, наприклад на основі неоднорідного пуассонового процесу [2].

Якщо потік управління між модулями системи моделюється як процес Маркова з неперервним часом, то імовірність перебування системи в i-му стані $p_i(t)$ отримують з розв'язку системи рівнянь Колмогорова-Чепмена:

$$\frac{dp_i(t)}{dt} = -\sum_{j\in S} a_{ij}(t)p_i(t) + \sum_{j\in S} a_{ji}(t)p_j(t), \qquad (2)$$

де $a_{ij}(t)$ – інтенсивність переходу зі стану i в стан j; S – множина усіх станів системи.

Вплив процесу вищого порядку виражається в залежності інтенсивності переходу між станами від історії потрапляння в поточний стан, і чим вищий порядок процесу, тим триваліший ланцюжок переходів впливає на цю інтенсивність. В цьому випадку постає задача отримання таких залежностей для урахування їх при обчисленні надійності програмної системи. Для вирішення цієї задачі у [8] був запропонований підхід, що полягає у побудові еквівалентного

процесу першого порядку. Відповідно до [8] марковський процес вищого порядку представляють у вигляді еквівалентного процесу першого порядку з додатковими фіктивними станами. При цьому кожен стан початкового графу розщеплюється на таку кількість фіктивних станів, скільки є різних шляхів до цього стану. Для обчислення кількості ланцюжків порядку K зі стану C_i в стан C_j використовують формулу, що базується на методі Флойда [8]:

$$m_{ij}^{K} = \sum_{j=1}^{N} \left(m_{il}^{K-1} + e_{il} \right) m_{lj}^{1} , \qquad (3)$$

тут N – кількість усіх станів системи, e_{ij} – елементи одиничної матриці.

На основі (3) отримують еквівалентний граф станів системи, складають систему рівнянь Колмогорова-Чепмена (2) для еквівалентного графу та отримують з її розв'язку імовірності виконання i-го програмного модуля в момент часу t. Використавши отримані імовірності в рівнянні (1) разом з отриманими в процесі модульного тестування значеннями інтенсивності відмов кожної компоненти отримують інтенсивність відмов програмної системи в цілому.

Модель 2

У [9] описано модель надійності програмного забезпечення з урахуванням модульної структури. Ця модель представляє передачу управління між модулями програми як марковський процес, при цьому розглядаються два типи відмов системи. Перший тип відмов пов'язаний з відмовами програмних модулів, вважаючи, що ці відмови реалізують пуассонів процес. Другий тип відмов породжується зв'язками між модулями – вважається, що при інтеграції модулів в систему вносяться нові помилки, а відмови такого типу пов'язані з інтерфейсами модулів.

Взявши за основу модель [9], побудуємо модель надійності програмного забезпечення вищого порядку з використанням наступних припущень та структури моделі:

- програмний продукт складається зі скінченної кількості модулів (1, 2, … , N), які можна представити напрямленим графом $\{C_i\}$, і які з точки зору діаграми надійності з'єднані в систему послідовно;
- кожен модуль програмної системи може бути протестований незалежно і характеризується інтенсивністю відмов λ_i, при цьому відмови кожного модуля мають однорідний пуассонів або експоненційний розподіл;
- поведінка програмної системи в часі (потік управління) описується марковською матрицею інтенсивності переходів **A**, де

$$a_{ij}\delta t = P\left\{ X\left(t+\delta t\right) = j \middle| X\left(t\right) = i \right\}, i \neq j \ , \ a_{ii} = -\sum_{j \neq i} a_{ij} \ ;$$

- операційний профіль програми відповідає протестованим варіантам використання системи і не змінюється з часом;
- інтенсивність переходів a_{ij} між станами i та j залежить від історії переходів між попередніми станами системи, яка закінчується поточним станом i;
- інтеграція модулів в програмну систему є неідеальною і на додачу до відмов програмних модулів виникають відмови, пов'язані з переходами між модулями; q_{ij} – імовірність відмови при передачі управління від модуля i до модуля j;
- модель є ієрархічною.

З урахуванням наведених припущень інтенсивність відмов програмної системи можна записати як:

$$\lambda\left(t\right) = \sum_{i=1}^{N} p_i\left(t\right)\lambda_i + \sum_{i \neq j} a_{ij}\left(t\right) q_{ij} \ , \qquad (4)$$

тут елементи матриці інтенсивності переходів залежать від історії переходів до поточного стану довжиною K (де K – оптимальний порядок ланцюга Маркова) і в загальному випадку можуть бути функціями часу.

Зауважимо, що хоча інтенсивності відмов модулів програмної системи не залежать від часу, інтенсивність відмов системи (4) є функцією часу виконання, що є наслідком складної структури системи до якої входять модулі з різною інтенсивністю відмов.

Як і у випадку моделі (1) ефективна робота з матрицею інтенсивності переходів вищого порядку може бути досягнута шляхом побудови еквівалентного процесу першого порядку [8] та розв'язком рівняння (2) з подальшою агрегацією фіктивних станів.

Важливим практичним питанням використання цієї моделі є визначення матриці імовірності виникнення відмов при передачі управління між модулями системи. Не виключаючи чисто емпіричного підходу до вирішення цієї задачі, можна запропонувати оцінку такої імовірності на основі кількості інформації, що передається при виклику функції та структурної складності програмної системи:

$$q_{ij} = f\left(I\left(i,j\right), M\right), \qquad (5)$$

де $I(i, j)$ – кількість інформації, що передається між модулями, M – цикломатична складність програми (метрика McCabe). Обидва значення можуть бути отримані під час розробки програмної системи шляхом статичного аналізу програмного коду. Зауважимо, що точний вигляд функціональної залежності (5) потребує подальших досліджень.

Висновки

Для підвищення ступеня адекватності реальним об'єктам побудовано дві нових математичних моделі надійності програмного забезпечення на основі ланцюгів Маркова вищого порядку з неперервним часом та з урахуванням відмов інтерфейсів модулів, які можуть використовуватись в залежності від рівня складності програмного продукту. Проведений в [10] порівняльний аналіз класичних моделей та моделей вищого порядку на реальних тестових прикладах показав більший ступінь адекватності моделей вищого порядку.

Список використаної літератури

1. Peled D.A., (2001), Software Reliability Methods, *Springer-Verlag New York Berlin Heidelberg,* 343 p.

2. Бобало Ю. Я. Математичні моделі та методи аналізу надійності радіоелектронних, електротехнічних та програмних систем: монографія / Ю. Я. Боба-

ло, Б. Ю. Волочій, О. Ю. Лозинський, Б. А. Мандзій, Л. Д. Озірковський, Д. В. Федасюк, С. В. Щербовських, В. С. Яковина. – Львів : Видавництво Львівської політехніки, 2013. – 300 с.

3. Pham Hoang, and Pham Michelle, (1991), Software Reliability Models for Critical Applications, *EGG–2663 Technical Report Idaho National Engineering Laboratory*, EG&G Idaho Inc.

4. Rahmani M., and Azadmanesh A., (2008), Exploitation of Quantitative Approaches to Software Reliability, *University of Nebraska at Omaha.*

5. Swapna S. Gokhale, W. Eric Wong, J.R. Horgan, and Kishor S. Trivedi, (2004), An Analytical Approach to Architecture-based Software Performance Reliability Prediction, *Performance Evaluation 58,*. pp. 391 – 412.

6. Bochmann G.V., Jourdan G.-V., and Bo Wan, (2011), Improved Usage Model for Web Application Reliability Testing. In: B. Wolff and F. Zaidi (Eds.): ICTSS 2011, LNCS 7019, pp. 15 – 31.

7. Yakovyna V., Fedasyuk D., Nytrebych O., Parfenyuk I., and Matselyukh V., (2014), Software Reliability Assessment Using High-Order Markov Chains, *International Journal of Engineering Science Invention*, Vol. 3(7), pp. 1 – 6.

8. Yakovyna V., Nytrebych O., and Fedasyuk D., (2013), The Representation of High Order Markov Process Through Equivalent First Order Process [Electronic Resource], "*Computer Science and Engineering 2013": Proceedings of the 6th International Academic Conference of Young Scientists (CSE-2013),* Lviv, Ukraine, 21-23 November, pp. 216 – 217, CD-ROM.

9. Littlewood B., (1975), A Reliability Model for Systems with Markov Structure, *Appl. Statist,* Vol. 24, No. 2, pp. 172 – 177.

10. Yakovyna V. and Nytrebych O., (2015), Discrete and Continuous Time High-Order Markov Models for Software Reliability Assessment, *Proc. 11-th Int. Conf. ICTERI 2015,* Lviv, Ukraine, May 14-16, 2015, CEUR-WS, Vol. 1356, pp. 419 – 431.

Отримано 14.05.2015

References

1. Peled D.A., (2001), Software Reliability Methods, *Springer-Verlag New York Berlin Heidelberg*, 343 p. (In English). DOI 10.1007/978-1-4757-3540-6

2. Bobalo Yu.Ya., Bolochiy B.Yu., Lozynskyy O.Yu., Mandziy B.A., Ozirkovskyy L.D., Fedasyuk D.V., Shcherbovskykh S.V., and Yakovyna V.S. Matematychni modeli ta metody analizu nadijnosti radioelektronnykh, elektrotekhnichnykh ta programnych system [Mathematical Models and Methods for Electronic, Electrical and Software Systems Reliability Analysis], (2013), Lviv, Ukraine, *Lviv Polytechnic Publishing House*, 300 c. (In Ukrainian).

3. Pham Hoang, and Pham Michelle, (1991), "Software Reliability Models for Critical Applications", *Idaho National Engineering Laboratory*, EG&G–2663 (In English).

4. Rahmani M., and Azadmanesh A., (2008), "Exploitation of Quantitative Approaches to Software Reliability", *University of Nebraska at Omaha*, Technical Report No. cst-2011-002 (In English).

5. Swapna S. Gokhale, W. Eric Wong, J.R. Horgan, Kishor S. Trivedi, (2004), An Analytical Approach to Architecture-based Software Performance Reliability Prediction, *Performance Evaluation 58,* pp. 391 – 412 (In English). DOI:10.1016/j.peva.2004.04.003

6. Bochmann G. V., Jourdan G.-V., and Bo Wan, (2011), Improved Usage Model for Web Application Reliability Testing. In: B. Wolff and F. Zaidi (Eds.): ICTSS 2011, LNCS 7019, pp. 15 – 31 (In English). DOI 10.1007/978-3-642-24580-0

7. Yakovyna V., Fedasyuk D., Nytrebych O., Parfenyuk I., and Matselyukh V., (2014), Software Reliability Assessment Using High-Order Markov Chains, *International Journal of Engineering Science Invention,* Vol. 3(7), pp. 1 – 6 (In English). Url: http://www.ijesi.org/papers/Vol(3)7/Version-2/A03720106.pdf

8. Yakovyna V., Nytrebych O., and Fedasyuk D., (2013), The Representation of High Order Markov Process Through Equivalent First Order Process, *Proc. of the 6th Int. Acad. Conf. of Young Scientists "Computer Science and Engineering 2013" (CSE-2013),* Lviv, Ukraine, 21-23 November, pp. 216 – 217 (In English).

9. Littlewood B., (1975), A Reliability Model for Systems with Markov Structure, *Appl. Statist*., Vol. 24, No. 2, pp. 172 – 177 (in English). DOI: 10.2307/2346564.

10. Yakovyna V., and Nytrebych O., (2015), Discrete and Continuous Time High-Order Markov Models for Software Reliability Assessment. *In: Batsakis, S. et al. (eds.), Proc. 11-th Int. Conf. ICTERI 2015,* Lviv, Ukraine*,* May 14-16, 2015*,* CEUR-WS.org/Vol. 1356, ISSN 1613-0073, pp. 419 – 431 (In English).

Яковина
Віталій Степанович
канд. фіз.-мат. наук, доцент
каф. програмного забезпечення
Національного університету
«Львівська політехніка».
E-mail: yakovyna@lp.edu.ua