

ВИКОРИСТАННЯ МАРКОВСЬКИХ ЛАНЦЮГІВ ПРИ МОДЕЛЮВАННІ ПРОЦЕСУ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Вінницький національний технічний університет

Анотація

У даній роботі обґрунтовано доцільність використання Марковських ланцюгів при моделюванні процесу розробки програмного забезпечення, що дозволить описати даний процес на декількох рівнях деталізації за допомогою відповідного математичного апарату.

Ключові слова: Марковські процеси; Марковські ланцюги; розробка програмного забезпечення; моделювання процесів.

Abstract

The feasibility of using Markov chains in modeling software development process, which will describe the process at several levels of detail by using the appropriate mathematical apparatus is substantiated in this paper.

Keywords: Markov processes; Markov chain; software development; process modeling.

Дослідження математичних моделей, за допомогою яких можна описати процес розробки програмного забезпечення (ПЗ), є актуальною задачею оскільки на даний момент переважають моделі, що базуються на типах даних (функціональні, реляційні або аксіоматичні), і яких недостатньо для опису динамічної поведінки систем.

Метою роботи є обґрунтування доцільності використання Марковських ланцюгів при моделюванні процесу розробки програмного забезпечення.

В роботі [1] здійснюється формалізація варіантів використання ПЗ користувачем програмного забезпечення через модель Кріпке [2], яка є одним із варіантів недетермінованого закінченого автомата. Автори [1] пропонують застосовувати шаблон, який перетворюватиме опис варіантів використання в структуру Кріпке, що в подальшому може бути використана для формальної верифікації системи або для автоматичної генерації тестових прикладів.

В роботі [3] здійснюється моделювання процесу розробки програмного забезпечення за допомогою Х-машини. Перший крок моделювання полягає в створенні двох частин системи: групи розробників, що має два стани: «зайнята» і «вільна» та черги із завдань, які необхідно передати в групу розробників. Автор [3] припускає реалізацію наступного механізму описаних Х-машин: коли стан групи змінюється на «вільна», відправляється повідомлення черзі, яка видає групі наступні завдання, якщо такі у ній є. Таким чином, повідомлення від Х-машини – «Групи» посилаються на вхід Х-машини – «Черга», що і моделює процес розробки ПЗ.

Наведені вище методи дозволяють розглянути загальний процес розробки ПЗ в динаміці, проте не розглядають його на різних етапах деталізації, що може бути реалізовано з використанням Марковських процесів та відповідного математичного апарату.

Перший рівень деталізації процесу розробки ПЗ відображає програмний продукт P , як скінчену множину компонентів (програм, що розглядаються як єдине ціле та виконують певну функцію), що описується виразом:

$$P = \{p_1, p_2, \dots, p_i, \dots, p_I\}, \quad (1)$$

де $i = \overline{1, I}$, $I \in N$.

Процес розробки ПЗ на даному етапі можна розглядати як випадковий процес $X(t)$, у якого область визначення T – це дискретна множина точок $t_0 < t_1 < \dots$, а простір станів являє собою дискретну множину P . Процес p_i в момент часу t_n (де $n=0,1,2,3,\dots$) може перебувати в одному із цих станів, а в

час t_{n+1} перейти в деякий інший стан чи залишитися в тому ж. Таке відображення процесу розробки ПЗ відповідає математичному опису Марковського ланцюга як випадкового процесу, що задовольняє властивість Маркова і приймає скінченну чи зліченну кількість станів [4]. Відповідний Марковський процес описується за допомогою однорідної умовної функції розподілу:

$$F_1 = (x_N | x_0, \dots, x_{N-1}) = F(x_N | x_{N-1}), N = 1, 2, \dots \quad (2)$$

Другий рівень деталізації процесу розробки ПЗ відображає детальний опис етапів розробки конкретного компонента, що розробляється за алгоритмом, який складається з таких кроків:

1. Аналіз поставленої задачі.
2. Проектування вимоги до ПЗ.
3. Реалізація ПЗ.
4. Тестування ПЗ.
5. Документування ПЗ.

На даному рівні деталізації, процес розробки окремого компонента можна розглядати як випадковий процес $X(t)$, у якого область визначення D – це неперервна множина точок $d \in D$, а простір станів S являє собою дискретну множину точок $\theta_l \in S$, де $l = \overline{1, L}$. В будь-які випадкові моменти часу $d_0 < d_1 < \dots$, можливі зміни стану. Такий процес являє собою дискретний випадковий Марковський процес, для якого одновимірна функція розподілу відображена у виразі:

$$F_1 = (x_N; d_N | x_0, \dots, x_{N-1}; d_0, \dots, d_{N-1}) = F(x_N; d_N | x_{N-1}; d_{N-1}). \quad (3)$$

Третій рівень деталізації процесу розробки ПЗ відображає детальний опис окремого етапу розробки конкретного компонента. Одним із таких етапів є тестування – процес виконання програми з метою виявлення помилки чи дефекту [5], основними кроками якого є: підготовка до тестування; розробка тестів; виконання тестів; оцінка результатів тестування.

Процес розробки ПЗ на даному етапі можна розглядати як випадковий процес $X(t)$, у якого область визначення T – це дискретна множина точок $t_0 < t_1 < \dots$, а простір станів являє собою дискретну множину $S = \{\theta_l, l = \overline{1, L}\}$. В момент часу t_n (де $n=0,1,2,3,\dots$) можлива зміна стану і такий випадковий процес можна розглядати як дискретну випадкову послідовність дискретних випадкових величин $X_N = X(t_N), N = 0,1,\dots$, а, отже, як Марковський ланцюг, функція розподілу якого описана виразом (2).

Отже, використання Марковських ланцюгів для моделювання процесу розробки програмного забезпечення на різних рівнях деталізації даного процесу є доцільним.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Zaman Q. Formalizing a Use Case to a Kripke Structure / Q. uz Zaman, A. Nadeem, M. A. Sindhu. – USA. Proceedings of the IASTED International Symposium Software Engineering and Applications, 2015. – 232-239 pp.
2. Stirling C. Modal and temporal logics / Colin Stirling. – GB. University of Edinburgh, Department of Computer Science, 1991. – 265 p.
3. Соболев М.С. Описание систем при помощи X-машин / М.С. Соболев. – М. Информационные технологии и вычислительные системы (4), 2009. – 22-27 с.
4. Тихонов В. И. Марковские процессы / В. И. Тихонов, М. А. Миронов. – М. Советское радио, 1977. – 488 с.
5. Синицын С. В. Верификация программного обеспечения / С. В. Синицын, Н. Ю. Налютин. – М. Московский Инженерно-физический институт (государственный университет), 2006. – 158 с.

Савчук Тамара Олександрівна — к.т.н, доцент кафедри комп'ютерних наук, професор кафедри комп'ютерних наук ВНТУ, Вінницький національний технічний університет, м. Вінниця

Приймак Наталія Василівна — аспірант кафедри комп'ютерних наук ВНТУ, Вінницький національний технічний університет, м. Вінниця, e-mail: nata.pryimak@gmail.com

Tamara O. Savchuk — Cand. Sc. (Eng), Assistant Professor of Computer Sciences Chair, Professor of the Computer Sciences Chair, Vinnytsia National Technical University, Vinnytsia

Natalia V. Pryimak — postgraduate of the Computer Sciences Chair, Vinnytsia National Technical University, Vinnytsia, e-mail: nata.pryimak@gmail.com