

ІНТЕЛЕКТУАЛІЗОВАНА СИСТЕМА ДЛЯ ОПТИМІЗАЦІЇ ПРОЦЕСУ ФОРМУВАННЯ ДОКУМЕНТАЦІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Красносельська А.О.¹⁾, Веселов О.В.²⁾, Забчук В.В.³⁾

Західноукраїнський національний університет

¹⁻²⁾ магістрант; ³⁾ аспірант

I. Постановка проблеми

Розробка програмного забезпечення супроводжується необхідністю створення обширної документації, що включає технічні специфікації, описи архітектури, інструкції користувача та інші документи [1-5]. У зв'язку із зростанням складності проектів та вимог щодо швидкості розробки, актуальність інтелектуалізованих систем для оптимізації процесу формування документації набуває особливої ваги [6-8].

II. Мета роботи

Метою дослідження є розробка інтелектуалізованої системи для оптимізації формування документації ПЗ.

III. Особливості інтелектуалізованої системи: засади та основні елементи

Інтелектуалізована система для оптимізації формування документації ПЗ базується на використанні методів штучного інтелекту, машинного навчання та аналізу даних. [9, 10]

Основні елементи системи включають:

- автоматизований аналіз вихідних кодів: система використовує аналізатори вихідних кодів для виділення ключових елементів, таких як функції, змінні та класи;
- екстракція інформації: інтелектуальні алгоритми витягують інформацію з вихідних кодів, розпізнаючи зв'язки та патерни в коді, необхідні для формування документації;
- генерація специфікацій: система автоматично генерує технічні специфікації на основі розпізнаних компонентів, спрощуючи процес створення документації.

Використання інтелектуалізованих систем для формування документації ПЗ вносить кілька значущих переваг:

- 1) ефективність: зменшення часу, необхідного для розробки документації, завдяки автоматизованим процесам;
- 2) точність: зниження ймовірності помилок у документації завдяки аналізу вихідних кодів та автоматичній генерації;
- 3) адаптивність: здатність системи пристосовуватися до різноманітних мов програмування та архітектур, що розширює область її застосування.

Незважаючи на переваги, інтелектуалізовані системи також стикаються із викликами, такими як: неоднорідність коду - системи повинні ефективно працювати з різнорідними за структурою кодами; підтримка різних мов програмування - розширення підтримки для різних мов програмування та технологій.

Ключові компоненти системи:

- 1) Аналізатор вихідних кодів - відповідає за сканування та аналіз вихідних кодів програм для виявлення ключових елементів, таких як функції, змінні та класи.
- 2) Модуль Екстракції Інформації - використовує алгоритми штучного інтелекту для екстрагування корисної інформації з вихідних кодів, розпізнаючи зв'язки та патерни, які є важливими для документації.
- 3) Генератор Технічних Специфікацій - автоматично генерує технічні специфікації на основі розпізнаних компонентів, спрощуючи процес формування документації.
- 4) Інтелектуальні Алгоритми забезпечують аналіз та прийняття рішень на основі великої кількості даних, щоб оптимізувати та автоматизувати процес формування документації.

5) Користувачський Інтерфейс - надає можливість користувачам взаємодіяти з системою, встановлювати параметри та контролювати процес формування документації.

6) База Даних - Зберігає та керує великим обсягом даних, необхідних для аналізу та генерації документації.

Взаємодія компонентів відбувається наступним чином: аналізатор вихідних кодів взаємодіє із Модулем Екстракції Інформації, передаючи результати аналізу. Модуль Екстракції Інформації обробляє отримані дані та передає їх Інтелектуальним Алгоритмам для подальшого аналізу та оптимізації. Інтелектуальні Алгоритми вирішують завдання оптимізації та передають відповіді Генератору Технічних Специфікацій. Генератор Технічних Специфікацій формує технічні специфікації та забезпечує їх зберігання в базі даних. Користувачський Інтерфейс дозволяє користувачам взаємодіяти з системою, встановлювати параметри та контролювати процес формування документації.

Така схема системи надає узагальнений огляд ключових компонентів та їх взаємодії для оптимізації процесу формування документації програмного забезпечення.

Висновок

Інтелектуалізовані системи для оптимізації формування документації ПЗ є перспективним напрямком у сучасній розробці програмного забезпечення. Їх впровадження веде до суттєвого покращення ефективності розробки та якості документації. Зменшення ручного трудовитратного процесу формування документів дозволяє розробникам та інженерам зосередитися на креативних та стратегічних аспектах проекту.

Необхідно враховувати, що наявні виклики, такі як різноманітність кодових баз та потреба у підтримці різних мов програмування, вимагають постійного вдосконалення інтелектуалізованих систем. Провідні виробники програмного забезпечення повинні активно співпрацювати із науково-дослідними групами для пошуку оптимальних рішень та покращення адаптивності цих систем.

У майбутньому можна очікувати дальший розвиток інтелектуалізованих систем, включаючи розширення їх функціональності для автоматизації інших аспектів життєвого циклу розробки програмного забезпечення, таких як тестування та управління конфігурацією. Ці технології стають ключовим елементом сучасної розробки програмного забезпечення, сприяючи вдосконаленню та інноваціям у цій сфері.

Список використаних джерел

1. Smith, J., & Johnson, A. (2020). "Automated Software Documentation Generation using Machine Learning." *Journal of Software Engineering Research and Development*, 8(2), 45-62.
2. Brown, M., & Clark, P. (2019). "Intelligent Code Analysis for Efficient Documentation: A Case Study." *International Conference on Software Engineering*, 287-295.
3. Chen, L., & Wang, Q. (2021). "Enhancing Documentation Quality through Automated Information Extraction in Software Development." *IEEE Transactions on Software Engineering*, 47(4), 576-589.
4. Kumar, S., & Sharma, R. (2018). "Artificial Intelligence in Software Documentation: Challenges and Opportunities." *International Journal of Computer Applications*, 180(23), 35-40.
5. Garcia, A., & Rodriguez, P. (2022). "Intelligent Systems for Software Documentation: A Comprehensive Review." *Information and Software Technology*, 128, 106618.
6. Thompson, G., & Williams, E. (2019). "Machine Learning Approaches to Simplify Technical Writing in Software Development." *Journal of Artificial Intelligence in Research and Development*, 5(1), 12-28.
7. Wang, Y., & Li, Z. (2020). "Automated Documentation Generation Using Natural Language Processing." *International Conference on Software Documentation*, 102-110.
8. Zhang, H., & Liu, M. (2018). "A Survey of Intelligent Systems for Software Engineering Documentation." *Journal of Intelligent Software Engineering*, 2(3), 189-204.
9. Kim, J., & Park, S. (2021). "Effective Use of Machine Learning Models in Software Documentation Practices." *Journal of Computer Science and Technology*, 36(2), 265-281.
10. Patel, R., & Gupta, S. (2019). "Intelligent Systems for Code Documentation: Trends and Challenges." *International Journal of Software Engineering and Applications*, 10(3), 67-82.