

АВТОМАТИЗОВАНА ПОБУДОВА UML-ДІАГРАМ У ПРОЦЕСІ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ: МЕТОДИ, ІНСТРУМЕНТИ ТА ПЕРСПЕКТИВИ

Вінницький національний технічний університет

Анотація

У статті розглядається роль автоматизованих систем у побудові UML-діаграм у контексті сучасної розробки програмного забезпечення. Аналізуються основні методи автоматизації, такі як зворотне проєктування, використання текстових мов специфікації та інтеграція в CI/CD-процеси. Оцінюється вплив автоматизації на ефективність розробки, зокрема зменшення часу на створення та оновлення діаграм, покращення комунікації в команді та зниження ймовірності помилок у документації. Крім того, висвітлюються ключові тенденції розвитку технологій автоматизації UML-діаграм, включаючи застосування штучного інтелекту для покращення процесу моделювання. Стаття також окреслює напрями подальших досліджень у цій сфері, зокрема інтеграцію UML-інструментів із новими технологіями.

Ключові слова:

UML-діаграми, автоматизація, зворотне проєктування, текстові специфікації, CI/CD, штучний інтелект, моделювання, програмне забезпечення, ефективність розробки, технології.

Abstract

The article discusses the role of automated systems in building UML diagrams in the context of modern software development. The main automation methods are analyzed, such as reverse engineering, the use of textual specification languages, and integration into CI/CD processes. The impact of automation on development efficiency is assessed, including reducing the time to create and update diagrams, improving team communication, and reducing the likelihood of errors in documentation. In addition, the article highlights key trends in the development of UML diagram automation technologies, including the use of artificial intelligence to improve the modeling process. The article also outlines areas for further research in this area, including the integration of UML tools with new technologies.

Keywords:

UML diagrams, automation, reverse engineering, textual specifications, CI/CD, artificial intelligence, modeling, software, development efficiency, technology.

Вступ

UML-діаграми є важливим інструментом у розробці програмного забезпечення, допомагаючи моделювати архітектуру систем і забезпечуючи спільне розуміння між розробниками. Проте їхня ручна побудова є трудомістким процесом, що може призводити до неточностей, розбіжностей із кодом та зниження продуктивності команди. У великих проєктах оновлення діаграм стає особливо складним завданням.

Метою дослідження є зменшення трудомісткості та помилок при побудові UML-діаграм, підтримка актуальності моделей та узгодженість між різними діаграмами.

Для досягнення цієї мети проводиться аналіз автоматизованих систем генерації UML-діаграм, їхніх можливостей та перспектив розвитку. Розглядаються методи автоматизації, такі як генерація на основі коду, текстові специфікації UML та інтеграція в процеси розробки, а також перспективи застосування штучного інтелекту у цій сфері.

UML-діаграми в програмній інженерії

UML (Unified Modeling Language) — це стандартизована мова моделювання, що використовується для візуального представлення архітектури, компонентів і процесів у програмних системах. Вона сприяє чіткому проєктуванню, покращенню комунікації між розробниками та полегшенню документації.

UML-діаграми поділяються на **структурні** та **поведінкові**. Структурні діаграми (класів, об'єктів, компонентів) відображають архітектуру та зв'язки між елементами. Наприклад, **діаграма класів**

використовується для моделювання об'єктно-орієнтованої системи. Поведінкові діаграми (послідовностей, діяльності, станів) відображають динаміку системи, як-от сценарії взаємодії між об'єктами у **діаграмі послідовностей**.

Ручна побудова UML-діаграм є трудомісткою та схильною до помилок, особливо у великих проєктах із частими змінами. Це ускладнює підтримку актуальності моделей та узгодженість між різними діаграмами. Тому автоматизовані засоби UML-моделювання набувають популярності, оскільки дозволяють швидко генерувати, оновлювати та синхронізувати діаграми з кодом, підвищуючи ефективність розробки.

Методи автоматизованої побудови UML-діаграм

Автоматизація побудови UML-діаграм значно спрощує моделювання програмних систем, усуваючи проблеми ручного оновлення схем та забезпечуючи узгодженість із кодом. Основні методи автоматизованої генерації UML включають зворотне проєктування (reverse engineering), використання текстових мов специфікації та інтеграцію UML-діаграм у CI/CD-процеси.

Генерація UML-діаграм на основі коду (reverse engineering) дає змогу автоматично будувати діаграми класів, послідовностей та компонентів, аналізуючи вихідний код. Це дозволяє швидко отримати актуальну модель системи, особливо у великих проєктах. Серед інструментів для цього підходу можна виділити Visual Paradigm, StarUML та IntelliJ IDEA UML-плагіни.

Використання текстових мов специфікації (PlantUML, Mermaid) дозволяє створювати UML-діаграми на основі текстового опису, що полегшує версіонування, інтеграцію з документацією та спрощує внесення змін. Ці інструменти підтримують автоматичну генерацію діаграм у Markdown-документах та вікі-сторінках.

Інтеграція UML-генерації в CI/CD-процеси забезпечує автоматичне оновлення діаграм під час змін у кодовій базі. Це покращує контроль над архітектурою та знижує ризик невідповідності документації реальному стану проєкту.

Порівняльний аналіз наведено в таблиці 1.

Таблиця 1 – Порівняльна характеристика інструментів з побудови UML діаграм

Інструмент	Reverse Engineering	Текстові специфікації	Інтеграція з IDE	Підтримка CI/CD	Основні переваги	Недоліки
StarUML	Немає	Немає	Обмежена	Немає	Простий, розширюваний, XMI	Відсутність автоматизації
Visual Paradigm	Так	Немає	Глибока	Немає	Потужні інструменти аналізу	Висока вартість
IntelliJ IDEA UML	Так	Немає	Вбудований	Немає	Автоматичне створення з коду	Обмежена підтримка UML
PlantUML	Немає	Так	Плагіни	Так	Легкість інтеграції, текстовий опис	Вимагає окремої візуалізації
Mermaid	Немає	Так	Плагіни	Так	Підтримка Markdown та CI/CD	Менше можливостей, ніж PlantUML

Порівняльний аналіз інструментів для автоматизованої побудови UML-діаграм показав, що вибір залежить від потреб проєкту. StarUML та IntelliJ IDEA UML добре підходять для локального моделювання та інтеграції в середовище розробки, але мають обмежену підтримку CI/CD і текстових специфікацій. Visual Paradigm є найпотужнішим серед платних рішень, пропонуючи AI-асистоване моделювання та широкий набір функцій. PlantUML і Mermaid орієнтовані на текстові специфікації та автоматизацію, що робить їх ефективними в CI/CD-процесах і командній роботі. Загалом, текстові інструменти краще підходять для автоматизації, тоді як графічні – для інтерактивного моделювання.

Практичне застосування та перспективи розвитку

Автоматизовані системи побудови UML-діаграм демонструють значні переваги в розробці складних програмних систем. Наприклад, використання інструментів, таких як Visual Paradigm та IntelliJ IDEA, дозволяє швидко генерувати моделі на основі вихідного коду, що спрощує процес аналізу архітектури та пришвидшує адаптацію до змін. Автоматизованість процесу у свою чергу дозволяє зекономити критичні обсяги часу, що витрачаються на розробку схем та діаграм.

За даними дослідження, команди, що використовують під час розробки UML діаграми зменшили час на розробку на 15-20% порівняно з командами, які не використовували UML [1].

Автоматизація UML-діаграм також позитивно впливає на комунікацію між членами команди, оскільки забезпечує єдине джерело правди щодо архітектури системи. У дослідженні брали участь 47 проєктів. Проєкти, які використовували UML, показали зниження кількості критичних помилок на 30% порівняно з проєктами, де UML не використовувався [2].

Перспективи розвитку технологій автоматизації UML-діаграм виглядають багатообіцяючими, зокрема у контексті застосування штучного інтелекту. AI-асистоване моделювання може забезпечити автоматичне аналізування коду та пропонувати оптимальні моделі на основі вивчених шаблонів. Це дозволить зменшити людський фактор, підвищити точність моделей і зекономити час на їх створення. У майбутньому можна очікувати інтеграцію AI у CI/CD-процеси, що ще більше спростить управління архітектурою програмних систем, про що свідчать результати дослідження [3].

Висновки

У процесі дослідження автоматизованої побудови UML-діаграм було виявлено, що автоматизація значно підвищує ефективність розробки програмного забезпечення. Використання сучасних інструментів, таких як Visual Paradigm, IntelliJ IDEA та текстові специфікації, дозволяє зменшити час на створення та оновлення діаграм, а також підвищити точність моделювання. Застосування методів зворотного проєктування та інтеграції в CI/CD-процеси сприяє підтримці актуальності моделей у динамічному середовищі розробки.

Ключові тенденції розвитку автоматизованого UML-моделювання включають зростаючу популярність AI-асистованого моделювання, яке обіцяє спростити процес створення UML-діаграм та підвищити їхню точність. Інтеграція AI у процеси моделювання дозволить розробникам автоматично генерувати діаграми на основі аналізу коду, що зменшить ймовірність помилок і полегшить адаптацію до змін у системі.

Подальші дослідження у цій сфері можуть бути спрямовані на розвиток нових алгоритмів для автоматизації UML-моделювання, аналіз ефективності різних підходів до генерації діаграм, а також вивчення впливу автоматизації на якість програмного забезпечення. Важливими є також дослідження в напрямку інтеграції UML-інструментів з новими технологіями, такими як блокчейн і Інтернет речей (IoT), що дозволить розширити можливості моделювання у складних і розподілених системах.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Erickson, J., & Siau, K. (2007). Can UML be simplified? Practitioner use and understanding of UML in software development. *Journal of Database Management*, 18(1), 1-18.
2. Grossman, M., Aronson, J. E., & McCarthy, R. V. (2005). Does UML make the grade? Insights from the software development community. *Information and Software Technology*, 47(6), 383-397.
3. B. S. Puranik, A. Sonawane, J. Jose, S. Chavan and Y. Patil, "Enhancement of Model Driven Software Development using AI," 2024 4th Asian Conference on Innovation in Technology (ASIANCON), Pimari Chinchwad, India, 2024, pp. 1-5, doi: 10.1109/ASIANCON62057.2024.10837847.

Кіпоренко Ілля Євгенович – студент групи ІПП-24м, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, м. Вінниця

Ліщинська Людмила Броніславівна – д-р техн. наук, професор, професор кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця, e-mail: llb@vntu.edu.ua

Kiporenko Illia Yevhenovych – Student of the Department of Software, Vinnytsia National Technical University, Faculty of Information Technology and Computer Engineering, Vinnytsia, e-mail: kiporenkoillia@outlook.com.

Lishchynska Lyudmyla Bronislavivna – Dr. Sc. (Eng.), Full Professor, Professor of Program Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: llb@vntu.edu.ua