

УДК 004.4'422:004.415.3

## **ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ТА ПРАКТИЧНОСТІ ІНСТРУМЕНТІВ ДЛЯ КОДОГЕНЕРАЦІЇ**

Денисюк В. М.

Науковий керівник – професор Четвериков Г. Г.

Харківський Національний Університет Радіоелектроніки, каф. ПІ  
м. Харків, Україна

email: [vladyslav.denysiuk@nure.ua](mailto:vladyslav.denysiuk@nure.ua)

Research on the effectiveness and practicality of code generation tools is crucial in modern software development. These tools aim to automate the process of writing code, potentially saving time and reducing human error. This study seeks to evaluate the performance and usefulness of such tools in real-world scenarios.

Робота присвячена покращенню взаємодії розробників та людей менш обізнаних у процесах розробки програмного забезпечення та уніфікації методів створення коду на базі візуально представлених сутностей та алгоритмів.

Кодогенерація – це концепт породження та продукції нового коду, виходячи із певного набору правил та джерела даних. Цей процес може відбуватися, коли потрібно оптимізувати використання ресурсів компанії та продукту, адже генерація виходячи із певних наборів правил, може економити час розробки, збірки програмного на етапі компіляції вихідного коду.

Використання UML діаграм для процесу кодогенерації є важливим етапом в розробці програмного забезпечення, оскільки воно дозволяє розробникам визначити архітектуру та структуру програми на високому рівні абстракції.

Одним із найпоширеніших механізмів є T4 (Text Template Transformation Toolkit) [1] – це потужний інструмент для автоматизації генерації текстових файлів, таких як код програм або конфігураційні файли. Цей дозволяє розробникам створювати шаблони, які можуть бути використані для генерації будь-якого тексту, включаючи код на C#, HTML, SQL та інші мови програмування.

Ще однією перевагою T4 є його гнучкість і простота використання. Шаблони T4 можуть бути легко налаштовані для генерації будь-якого типу текстових файлів, від простих конфігураційних файлів до складних класів програм.

Не дивлячись на свою потужність і гнучкість, використання T4 вимагає від розробників певного рівня знань. Важливо ретельно розробляти шаблони, щоб уникнути можливих помилок і забезпечити якість генерованого коду.

Для вирішення проблем сумісності розширень та стандартизації процесу перетворення візуально представлених даних у код, пропонується наступний порядок дій:

1. Створення UML діаграми: Спершу потрібно створити UML діаграму, що відображає структуру вашого програмного забезпечення, включаючи класи, взаємозв'язки, атрибути та методи. Під час цього процесу потрібно притримуватися підходу Model Driven Engineering (MDE) [2].

2. Створення шаблону T4: Наступним кроком є створення шаблону T4, який буде використовуватися для генерації коду на основі UML діаграми. Цей шаблон може містити логіку для розбору XML-структури діаграми та генерації відповідного коду.

3. Налаштування шаблону для генерації коду: В шаблоні T4 можна використовувати вбудований об'єкт XmlReader для розбору XML-структури діаграми. За допомогою цього об'єкту можна отримати необхідну інформацію про класи, атрибути, методи та інші елементи діаграми.

4. Генерація коду на основі UML діаграми: Після налаштування шаблону T4 він може бути використаний для генерації коду на основі інформації з UML діаграми. Наприклад, шаблон може містити цикл, який проходиться по всім класам діаграми і генерує відповідний код для кожного класу.

5. На основі вихідного коду можна доповнити та оновити навчену модель використовуючи рішення навчання на основі повторного використання знань [3].

Список використаних джерел:

1. Code Generation and T4 Text Templates – Visual Studio (Windows). Microsoft Learn: Build skills that open doors in your career.

2. Bashir S. Galadanci M. I. M. Automatic code generation from uml diagrams: the state-of-the-art. Science World Journal Vol 13(No 4).

3. О. Каратаєв, І. Шубін. Проблеми повторного використання знань у процесі проєктування програмних систем. Innovative technologies and scientific solutions for industries. 2023. No. 2, № (24). С. 62. URL: <https://doi.org/10.30837/ITSSI.2023.24.0>.