

УДК 004.4'2:004.5

ПРОЄКТУВАННЯ АРХІТЕКТУРИ ТА АЛГОРИТМІЧНОГО ЗАБЕЗПЕЧЕННЯ ВІЗУАЛЬНОГО КОНСТРУКТОРА ВЕБ-ІНТЕРФЕЙСІВ

Недужко В.Р., студент

Київський національний університет технологій та дизайну

Демківська Т.І., канд. техн. наук, доцент

Київський національний університет технологій та дизайну

Ключові слова: візуальне моделювання, генерація коду, абстрактне синтаксичне дерево, DOM-модель, патерни проєктування, користувацький інтерфейс.

Сучасний етап розвитку індустрії програмного забезпечення характеризується стрімким переходом до парадигми Low-code та No-code розробки [1]. Традиційний процес створення користувацьких веб-інтерфейсів (UI), який передбачає ізольовану роботу графічних дизайнерів та подальшу ручну верстку макетів фронтенд-інженерами, є економічно ресурсозатратним. Актуальним науково-прикладним завданням є розробка спеціалізованих візуальних конструкторів, здатних забезпечити детерміновану трансляцію графічних примітивів у семантично валідний та оптимізований програмний код.

Аналіз існуючих рішень (Webflow, Framer) виявляє ряд концептуальних недоліків їхньої архітектури. Більшість систем базується на використанні жорсткого абсолютного позиціонування елементів за координатами (X,Y), що порушує принципи адаптивності веб-документів та призводить до надмірної генерації надлишкового коду (явище «div-soup»). Крім того, наявні платформи створюють штучні обмеження (vendorlock-in) щодо експорту вихідного коду для його розгортання на незалежних серверах.

Метою даного дослідження є проєктування архітектури та програмна реалізація візуального конструктора інтерфейсів, який оперує абстрактним синтаксичним деревом (AST) для динамічної генерації розмітки на базі алгоритмів Flexbox та CSSGrid[2].

Основою розробленої системи є математична модель представлення візуального компонента. Будь-який елемент на робочому полотні конструктора (контейнер, текст, зображення) формалізується у вигляді кортежу стану, де ID – унікальний ідентифікатор вузла; T – тип DOM-елемента; P – множина статичних атрибутів; S – множина динамічних стилістичних властивостей (CSS-правил); Ch – впорядкований масив дочірніх вузлів, що забезпечує рекурсивну вкладеність компонентів.

Сукупність таких кортежів формує глобальний стан застосунку у вигляді AST-дерева, яке серіалізується у формат JSON. На рис. 1 наведено структурну схему алгоритму функціонування розробленого програмного комплексу.

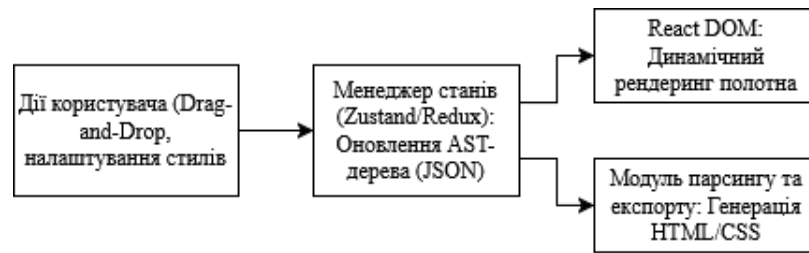


Рисунок 1 – Схема бізнес-процесу

Для забезпечення високої продуктивності системи та уникнення затримок при рендерингу складних макетів, архітектуру конструктора побудовано на базі бібліотеки React. Управління глобальним станом (AST-деревом) реалізовано за допомогою спеціалізованих патернів управління станом (Zustand/Redux), що дозволяє ізолювати логіку обчислень координат від логіки відображення інтерфейсу. Реалізація механіки вільного переміщення об'єктів (Drag-and-Drop) спирається на розрахунок векторів перетину меж елементів (boundingclientrect) у реальному часі.

Ключовим алгоритмічним модулем системи є підсистема експорту чистого коду. Розроблений рекурсивний алгоритм парсингу виконує обхід дерева S у глибину (Depth-FirstSearch). На кожному кроці ітерації алгоритм генерує відповідні HTML-теги, спираючись на параметр T , та екстрагує множину стилів S у відокремлений віртуальний файл styles.css із присвоєнням унікальних класів-хешів. Це унеможливує конфлікт каскадних стилів та гарантує семантичну чистоту згенерованого результату.

Додатковим модулем розширення функціональності є допоміжна інтеграція методів генеративного штучного інтелекту для етапу швидкого прототипування. Оперуючи API-запитами до сервісів класу Pollinations, система дозволяє асинхронно генерувати графічні плейсхолдери безпосередньо в середовищі редактора[3]. Крім того, реалізовано алгоритм автоматичного розрахунку комплементарних колірних палітр на основі зсуву фази в колірному просторі HSL.

Запропонована архітектура візуального конструктора вирішує проблему оптимізації процесу фронтенд-розробки. Використання абстрактних синтаксичних дерев замість прямого маніпулювання DOM-моделлю дозволило реалізувати стабільний механізм генерації чистого, адаптивного HTML/CSS коду без жорсткої прив'язки до платформи розробника.

Список використаних джерел

1. Algorithms in Low-Code-No-Code for Research Applications: A Practical Review. URL:<https://www.mdpi.com/1999-4893/16/2/108>
2. World Wide Web Consortium (W3C). CSS Flexible Box Layout Module Level 1. W3C Candidate Recommendation Draft. 2023. URL: <https://www.w3.org/TR/css-flexbox-1/>
3. Pollinations.ai API Documentation. Generative AI for Visual Content Integration. 2024. URL:<https://pollinations.ai/docs>