Войтюк О. В., аспірант
Державний університет «Житомирська політехніка»

ОПТИМІЗАЦІЯ ПРОМАЛЬОВУВАННЯ ВЕБЗАСТОСУНКУ НА ОСНОВІ ОБ'ЄКТІВ З ГЛИБОКОЮ ВКЛАДЕНІСТЮ ТА БАГАТОЗАЛЕЖНИМИ ЗВ'ЯЗКАМИ

Сучасний світ інформаційних технологій стрімко набирає швидкість свого розвитку, адже кожний веб застосунок прагне стати найкращим у низці категорій, зокрема швидкість роботи, якість обслуговування, доступність та дизайн. Кожен з цих аспектів напряму впливає на втриманні власного користувача. Одним з визначних факторів впливу є швидкість роботи веб застосунку, а точніше швидкість промальовування веб елементів. Веб додатки з малим набором елементів взаємодії користувача та інтерфейсу не потребують прийняття складних архітектурних рішень. Вони більшою мірою розраховані на ознайомлення користувача з інформаційним наповненням додатку або на швидке досягнення бажаної мети користувача, зокрема онлайн замовлення, запис на прийом, бесіда в чаті, тощо. Рідше зустрічаються додатки з великими об'ємами інформації, яка має бути конфігурована користувачем. Наприклад: система налаштування 3D принтеру або структурована схема налаштувань будь-якої сутності в системі.

Приклади таких складних інтерфейсів потребують динамічної зміни наповнення додатку. Зі свого боку це збільшує час на обробку та відмальовування користувацького інтерфейсу. Розробка таких інтерфейсів може бути ускладнена глибокою вкладеністю елементів та можливістю їх залежності один від одного.

Кожен з фреймворків або бібліотек, де відбувається розробка продукту, пропонує низку інструментів які допомагають вирішити проблему динамічного відмальовування. Однак не кожен розробник розуміє тонкощі правильного використання цих інструментів та може навпаки уповільнити процес оптимізації додатку.

Результатом аналізу проблеми відмальовування веб застосунку є розробка методу оптимізації промальовування та створення програмного продукту, який можна впровадити у веб застосунок. Зі свого боку це дозволить розробнику отримати низку переваг, серед яких можна виділити:

- зменшення часу на оновлення даних у існуючих компонентів системи;
- зменшення часу на оброку та калькуляцію вхідних параметрів;
- зменшення візуальних дефектів при взаємодії користувача та інтерфейсу;
- зменшення часу на налагодження системи;
- підвищення продуктивності додатку;
- єдине джерело структури даних;
- зменшення кількості дублюючого коду;
- можливість використовувати інструменти перегляду стану компонентів.

Сам метод оптимізації полягає у створенні обгортки для вже існуючих компонентів, який бере на себе усю логіку по прийняттю рішень з відмальовування компонента та її розміщення у структурі додатку.

Сутність головної обгортки полягає у створенні одного джерела даних, яка буде отримувати на вхід об'єкт з невпорядкованою схемою даних, а саме: початкове значення, межі можливих значень, набір опцій, назви елементів, тощо. Мета головної обгортки – обробити цей набір даних та перетворити його структуру, яка, у підсумку, буде мінімально необхідна для подальшої роботи. Наступним етапом оброки цих даних буде встановлення нових значень, які будуть допомагати системі будувати правильну позицію елементів в інтерфейсі, а також значень, завдяки яким система зможе орієнтуватися серед усіх інших компонентів. Наявність такої обгортки дасть можливість мати семантичну та ієрархічну структуру даних.

Сутність другорядних обгорток буде полягати в обробці вже існуючих компонентів, але з додаванням зовнішньої логіки. Вона буде перевіряти вхідні данні на їх тотожність або відмінність, щоб завчасно знати чи потрібна додаткове промальовування того чи іншого компоненту. До цієї перевірки також буде підключатися перевірка на зв'язки, що дасть змогу мати непрямий зв'язок між різними елементами, водночас не передаючи напряму їх значення. Наприклад можна розглянути залежність вибору столиці від країни – якщо користувач змінив країну то столиця має змінитись автоматично. Цей тип обгорток буде мати можливість рекурсивного створення, тобто обгортка буде створювати свої дублікати всередині самої себе, до того моменту, поки не закінчяться усі дочірні елементи.

Таким чином, система оптимізації промальовування веб застосунку дає розробнику можливість гнучко впроваджувати великі обсяги вхідних даних, водночас не піклуючись про результуючу продуктивність додатку.