

Література:

1. Kumar A., Vaisla K. Natural language interface to database: development techniques. *Computer science and engineering*. 2013. Vol. 58. P. 14724–14727.
2. Zhanza A., Khan M., Lee S. Abbas A. Pipeline and deep learning approach for NLIDB: a comparative study. *Journal of theoretical and applied information technology*. 2020. Vol. 98. P. 3667–3681.

УДК 004.45

БРАУЗЕРНИЙ МЕНЕДЖЕР ЗБІРКИ FIRMWARE

Ярмілко А. В., Голик Р. Р.

Черкаський національний університет імені Богдана Хмельницького

Abstract. The user interaction with software development tools using an interactive shell is optimized. The process of obtaining Firmware is modeled, and the possibility of implementing an IDE environment based on this model is proven.

Ефективність розробки програмного забезпечення (ПЗ) залежить від багатьох чинників, і серед них вибір оптимальних засобів розробки є хоч і не визначальним, але значимим [1]. Безпечливе надання переваги комплексним інтегрованим середовища розробки (IDE), в ряді випадків, може не виправдати очікувань, а інколи мати і більш негативні наслідки. До того ж, переважно, такі середовища мають високу вартість, яку можливо відшкодувати лише виконанням великих проектів та при наявності масових замовлень. Безкоштовні версії комплексних IDE, в більшості випадків, функціонально не повні або мають ліцензійні обмеження за характером використання отриманих в них програмних застосунків. До того ж, вони, зазвичай, потребують значних системних ресурсів та не портатбельні. Кардинально іншим підходом створення ПЗ є використання консольних засобів розробки. В мінімалістичному варіанті це може бути інструментальний набір компіляції – Toolchain GNU GCC та утиліта автоматизованої збірки, наприклад, Snake або Ninja. Використання такого набору вимагає від розробника високого професійного рівня та навичок у формуванні набору програмних компонент, розробки кодових модулів, та опису сценарію збірки проекту. Ефективна розробка ПЗ для вбудованих систем має свої особливості [2], причому на всіх стадіях проектування. Актуальність проведених досліджень полягає в оптимізації взаємодії користувача із інструментальними засобами розробки ПЗ шляхом використання інтерактивної оболонки – менеджера автоматизації збірки. Мета дослідження полягає у знаходженні моделей, методів та форм реалізацій засобів ефективною розробки вбудованого ПЗ. Фабула дослідження зводиться до використання браузера як платформи функціонування IDE, а мови Javascript – в якості засобу реалізації бізнес-логіки автоматизації збірки проекту та сервісів встановлення середовища

оточення. Були вирішені такі задачі дослідження: забезпечення доступу браузерного застосунку (BrApp) до ресурсів операційної системи, автоматизація процесу встановлення середовища оточення, генерації скриптів та збірки проекту, завантаження бінарного образу.

Системність вирішення поставлених задач досягнуто шляхом синтезу моделі цільного процесу отримання Firmware, яка на рис. 1 наведена у вигляді діаграми станів. Менеджер збірки повинен підтримувати діяльність програміста у всіх станах процесу проектування. За змістом та характером процес отримання вбудованої програми перебуває у двох станах: формування робочого профілю, розробки та використання результатів збірки проекту. Розмежування станів обумовлене відмінністю способів використання ресурсів ОС та діяльності програміста на цих етапах.

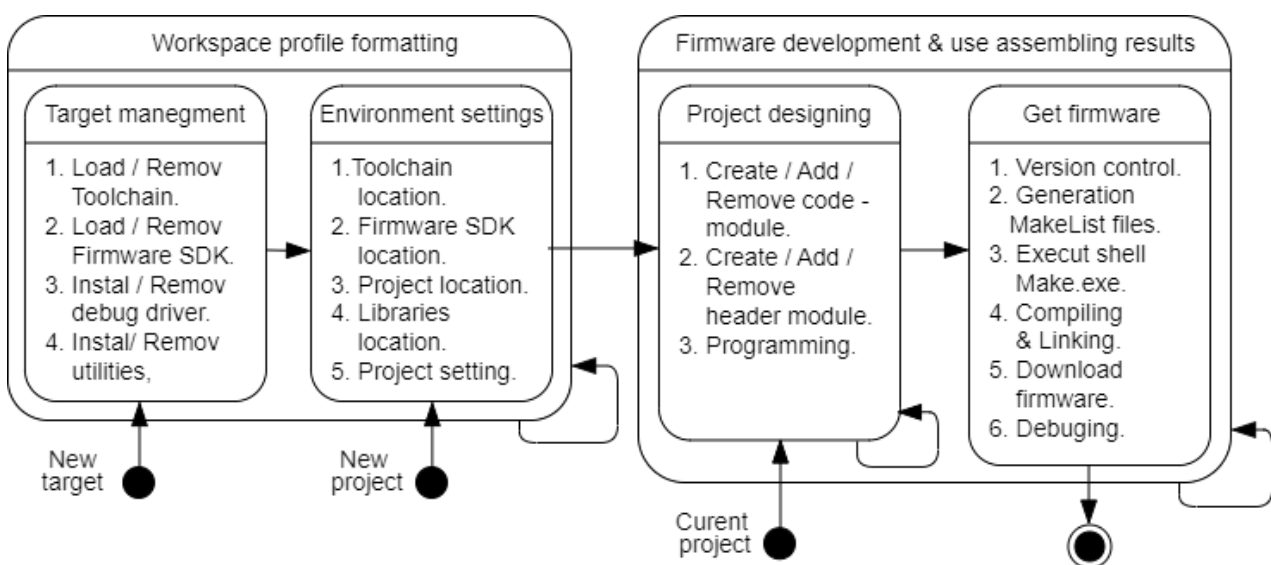


Рисунок 1 – Модель цільного процесу отримання Firmware

Стосовно використання системних ресурсів BrApp варто відзначити, що всі методи є прийнятними, якщо вони не суперечать визначеній політиці безпеки операційних систем. Сучасні браузерни мають достатньо потужні розширення, які надають js-контролеру застосунка доступу до ресурсів, взаємодій та сервісів ОС: через execute-shell оболонку, URL та системний обробник протоколів; через API браузера, який складаються з простору імен і відповідних полів маніфесту дозволів та опису об'єкту дії; через інтегровані webDAV, HTTP (оболонки виконання shell2http або Shellst) та SSH сервера. Так, наприклад, на стадіях автоматизованого налаштування середовища оточення (Environment settings), розробки програмного коду проекту (Project designing) та перегляду результатів збірки проекту (Get firmware) зручно застосовувати вбудований у ОС webDAV-сервер (для ОС Windows служби IIS). Проведення інсталяцій (стадія Target manegment), запуск генератора MakeList файлів, утиліт Smake або Ninja або інших програм пакету Toolchain (стадія Get firmware) зручно реалізовувати через URL доступ до системних обробників протоколів. Відладка firmware проводиться через HTTP-протокол.

За результатами дослідження експериментально доведено можливість реалізації IDE середовища, кросплатформного за видом проектування та браузерного застосунку – за формою виконання, яке підтримує модель наскрізного процесу отримання Firmware. Практична цінність досліджень та розробки полягає в автоматизації процесів створення профілю користувача, генерації сценарію та проведення збірки проекту, завантаження Firmware та відладки вбудованого ПЗ. Виконання IDE у формі браузерного застосунку дозволяє відмовитись від використання незручного інтерпретатора Python, а варіацію автоматизації збірки задавати виключно підключенням оптимального js-контролера. Перенесення всього процесу отримання Firmware на локальний рівень забезпечує кращу захищеність проектів, розвантажує мережу та підвищує ефективність розробки. Інтеграція webDAV-клієнта у браузерний IDE забезпечує можливість управління версіями та кооперативні методи розробки ПЗ за GIT-доступом.

Література:

1. T. Semiawan Analysis of the Effectiveness and Efficiency of Software Development Tools / T. Semiawan, R. Alifi, H. Hayati: веб-сайт. URL: https://www.researchgate.net/publication/357216868_Analysis_of_the_Effectiveness_and_Efficiency_of_Software_Development_Tools (дата звернення: 12.05. 2023).
2. Efficient Embedded System Development: A Workbench for an Integrated Methodology/ N. Hili, Ch. Fabre, S. Dupuy-Chessa: веб-сайт. URL: <https://hal.science/hal-00671966/document> (дата звернення: 12.05. 2023).

УДК 004.4

ПРОГРАМНА ІНЖЕНЕРІЯ НМІ-РОЗШИРЕНЬ ВБУДОВАНИХ СИСТЕМ

Ярмілко А. В., Даценко Д. В.

Черкаський національний університет імені Богдана Хмельницького

Abstract. A generalized requirements list for developing embedded systems' NMI extensions has been compiled. An innovative GUI framework model has been proposed, segmented corresponding to the characteristics of the GUI libraries.

Програмна інженерія застосунків персональних комп'ютерів (PC), включаючи їх ультра-мобільний аналог (UMPC), має відмінності від розробки ПЗ вбудованих систем як за підходами аналізу вимог, проектування, конструювання, відлагодження, тестування, менеджменту конфігурування, супроводу, так і за використанням інструментальних засобів. Виокремлення із галузі інформаційних технологій спеціалізації «програмна інженерія вбудованих систем» дозволяє краще врахувати специфіку підходів, підвищити ефективність отримання такого виду програмного забезпечення, а отже є актуальною задачею досліджень. Особливість поточної стадії дослідження