

4. 20 прикладів доповненої реальності в освіті [електронний ресурс]. – Режим доступу: <https://arnext.ru/articles/20-ar-eksperimentov-v-obrazovanii-2353>.

5. VR/AR в дитячій освіті: нащо технології потрібні школам? [електронний ресурс]. – Режим доступу: <https://medium.com/modum-lab/vr-ar-v-detskom-obrazovanii-zachem-tehnologii-nuzhny-shkolam-803b06245eeb>.

УДК 004.8

*Гайтан О.М., старший викладач,
Бочкарь В.О., студент групи 402-ТН*

Національний університет «Полтавська політехніка імені Юрія Кондратюка»

МЕТОДИ МОДЕЛЮВАННЯ КОРИСТУВАЦЬКОГО ІНТЕРФЕЙСУ ВЕБ-ДОДАТКІВ

Широке впровадження веб-додатків у всі сфери людської діяльності, за останні десятиліття спричинило стрімке зростання складності користувацьких інтерфейсів. Одночасно зі збільшенням інформації, з якою користувач мав змогу взаємодіяти в системі, збільшилася кількість самих користувачів зі своїми особливостями сприйняття інформації та вимогами щодо зрозумілості та інтуїтивності інтерфейсів. Також зросла кількість апаратних засобів, на яких повинен функціонувати інтерфейс. Зручність використання та користувацький досвід можуть стати вирішальними факторами успіху або невдачі веб-додатків та сайтів. Погано розроблені інтерфейси збільшують кількість помилок користувача, що може дорого коштувати. У результаті гостро постала необхідність моделювання користувацьких інтерфейсів, які б надавали можливість ефективної взаємодії користувача зі складною системою, при цьому не обмежуючи її функціональні можливості [1].

Одним з ефективних підходів до моделювання користувацьких інтерфейсів є орієнтований на використання дизайн (usage-centered design, або UCD).

Usage-centered design – це систематичний процес із використанням абстрактних моделей для проектування мінімальної, найпростішої системи, яка повністю підтримує всі завдання, які необхідно виконати користувачам. Даний метод, розроблений ще на початку 1990-х років, є перевіреним промисловим підходом, який використовується для проектування різноманітних користувацьких інтерфейсів, починаючи від системи промислової автоматизації та побутової електроніки до банківської справи та компаній зі страхування [2]. Оскільки це спрощений процес, керований простими моделями, він легко масштабується і використовується в проектах з невеликими трудовими ресурсами. Дизайн, орієнтований на використання, заснований на трьох простих, тісно зв'язаних абстрактних моделях: роль, завдання та зміст.

Модель ролі відображає головні характеристики ролей, які відіграють користувачі по відношенню до системи. Роль користувача являє собою один із видів відносин, які користувачі можуть мати з веб-додатком. Серед численних можливих аспектів цих відносин є мета і частота взаємодії, обсяг і спрямованість інформаційного обміну тощо.

Модель завдання представляє структуру роботи, яку користувачі повинні виконати в результаті взаємодії з системою.

Модель змісту, або контенту представляє вміст і організацію користувацького інтерфейсу, необхідного для підтримки визначених завдань. Перелік завдань, які ставляться перед системою, формують випадки використання (use cases). Це окремі послідовності дій, які включають в себе послідовності можливих помилок та взаємодіють з системою. Зазвичай варіанти використання виражають конкретні дії та відповіді системи.

Після визначення ролей користувачів майбутнього інтерфейсу та формування випадків використання здійснюється побудова карти випадків використання (task cases map). Карта випадків використання відображає зв'язки між реальними прикладами завдань та представляє собою діаграму, що показує всі випадки завдання та їх взаємозв'язки. Це ефективний інструмент для отримання огляду всіх завдань, які потрібно підтримувати у системі. Крім того, карта варіантів використання та моделі відносин допомагають розробнику інтерфейсу вирішувати, що об'єднати і що розділити в інтерфейсі користувача.

Етап розробки варіантів використання завершується побудовою абстрактних прототипів, тобто прототипів, які дозволяють розробникам проектувати загальну організацію та архітектуру користувацького інтерфейсу веб-додатку без візуалізації її компонентів або деталізації макета. Абстрактні прототипи складаються з моделі вмісту інтерфейсу, що описує вміст різних контекстів, у яких користувач взаємодіє з системою, та містить контекстну навігаційну карту, яка показує, як користувачі переходять від одного контексту до іншого під час виконання випадків задач (task cases).

Використання абстрактних прототипів замість реальних прототипів на даному етапі проектування має ряд переваг. По-перше, вони надають повну картину загальної архітектури інтерфейсу. По-друге, абстрактний характер моделей дозволяє відкласти прийняття детальних рішень. Абстракція заохочує до творчості в дизайні – досвід показав, що використання абстрактного прототипу як керівництво до побудови остаточного візуального дизайну забезпечує більш інноваційний користувацький інтерфейс [3].

На рис. 1 подано схематичний вигляд процесу моделювання користувацького інтерфейсу методом UCD. Системні актори – це інші програмні та апаратні системи, з якими повинен взаємодіяти інтерфейс, вони відокремлені від акторів, що мають ролі користувачів.

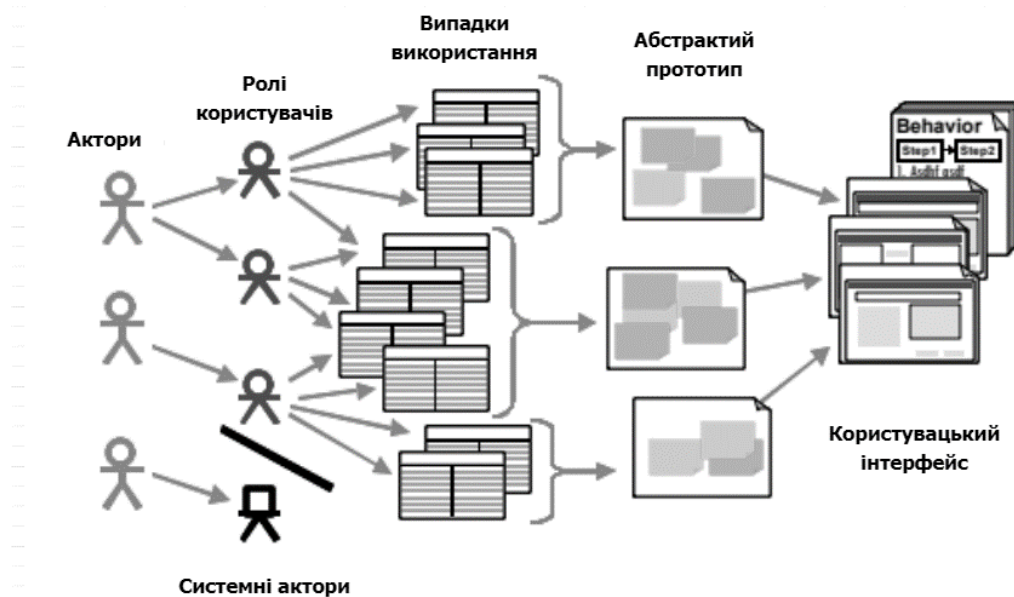


Рис. 1. Схема процесу проектування методом usage-centered design

За таким підходом кожна сторінка, форма чи інший контекст взаємодії відповідає абстрактному прототипу, що підтримує кластер взаємопов'язаних випадків задач. Компоненти користувацького інтерфейсу, такі як кнопки, посилання, таблиці, екрани та інші функції походять безпосередньо з абстрактних компонентів, які реалізують конкретні кроки в рамках підтримуваних випадків завдання. Ці варіанти завдань, у свою чергу, підтримують виконання ролей, які користувачі можуть виконувати щодо сайту чи програми. Такий підхід до проектування користувацького інтерфейсу гарантує, що кінцевий варіант інтерфейсу повністю покриває усі можливі варіанти взаємодії користувача з системою, а також залишає можливість внесення змін у прототип з мінімальними витратами у часі.

Література

1. Речинський А. В., Сергєєв С. Ф. Розробка інтерфейсів користувача. Юзабіліті-тестування інтерфейсів інформаційних систем: навч. посібник. – СПб.: Вид-во Політехн. ун-ту, 2012. - 145 с.
2. Anderson J., Fleek F., etc. Integrating Usability Techniques into Software Development. – IEEE Software, 18 (1), January/February, 2001.
3. Cloyd M.H. Designing User-Centered Web Applications in Web Time. – IEEE Software, 18 (1), January/February, 2001.

СУЧАСНІ ТЕНДЕНЦІЇ В РОЗРОБЦІ ТЕЛЕГРАМ-БОТІВ

Telegram бот – це програма, яка зв'язується з серверами компанії Telegram за допомогою API від розробників. Основна робота бота опитувати сервери із заданою періодичністю на наявність нових повідомлень від користувачів. У разі підтвердження наявності нового повідомлення, з сервера компанії відправляється файл типу JSON з детальною інформацією про відправника та зміст повідомлення.

Telegram створив офіційний сайт де розмістив детальну інструкцію для полегшення роботи розробників. Сайт постійно оновлюється та модернізується, як і сам API [1].

Телеграм досить популярний месенджер, який впевнено набирає популярність. Його використовує велика кількість користувачів. Це приваблює певних покупців, безліч організації як комерційних, так і некомерційних. Для вирішення великої частини їх потреб можуть знадобитися роботи які будуть вирішувати певні завдання для економії часу і грошей людей/організацій. Це і визначає напрямки розвитку чат-ботів.

Після появи такого інтелектуального програмного забезпечення, воно знайшло досить швидко широке застосування. Сьогодні його активно використовують у різних сферах (рис. 1). В електронній комерції чат-бот став незамінним помічником, так як дає можливість не «роздувати» штат операторів для обробки кожного звернення, а надавати на початковому етапі онлайн консультації за стандартними запитам користувачів. На інформаційних порталах і та онлайн довідниках такі програми допомагають знайти необхідну інформацію за лічені секунди. Активно використовується для навчання іноземним мовам. Допомагають самостійно вивчати англійську та інші мови, і навіть оцінюють знання своїх учнів. Чат-боти можуть грати з користувачем в ігри, надавати необхідну інформацію про правила гри [3].

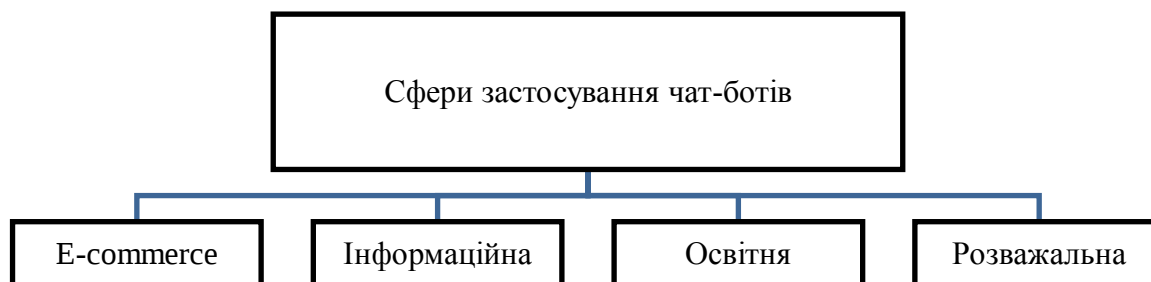


Рис. 1. Сфери застосування чат-ботів

Широкого розповсюдження набуло створення телеграмів ботів мовою програмування Python. Найпопулярніші технології створення телеграм