

УДК 004.055

Гученко І.В.**Національний авіаційний університет**

ЗАСІБ УПРАВЛІННЯ ЗРУЧНІСТЮ ВИКОРИСТАННЯ ПРОГРАМНИХ ПРОДУКТІВ

Article describes the software system that implements discussed in previous author's works [1] – [4] model and method of managing software products usability considering users' requirements. Short description of the major system components, their purpose and features of program implementation is given.

Ключові слова: програмне забезпечення, зручність використання, відгуки користувачів, автоматизована оцінка зручності використання, управління зручністю використання.

Вступ

У багатьох дослідженнях зроблено спроби надати визначення зручності використання (ЗВ) програмного продукту (ПП), але найчастіше вони є неузгодженими [5]. Тому будемо використовувати визначення, наведене в стандартах ISO 9241-11 [6] та ISO/IEC 25010:2011(оновлений ISO/IEC 9126-1:2001) [7]:

Зручність використання (usability) – ступінь, в якому ПП може бути використаний певними користувачами для досягнення визначених цілей з ефективністю, економічністю та задоволеністю у певному контексті використання.

Зручність використання ПП (надалі – ЗВ), з одного боку, визначає витрати ресурсів (наприклад, часу та зусиль) щодо точності та повноти досягнутих цілей, тобто впливає на продуктивність праці людини – і це вкрай важливо для користувача ПП, а з іншого боку, ЗВ є одним із визначних факторів при виборі ПП, тобто пов'язана з його конкурентоспроможністю – і це важливо для розробника ПП. Тому забезпечення ЗВ актуальне для обох зацікавлених сторін.

З огляду на важливість ЗВ її необхідно досягати та підтримувати, тому в процесі розробки та супроводу ПП виконуються дії, які забезпечують створення зручного у

Стаття присвячена опису програмної системи, яка реалізує розглянуті в попередніх роботах автора [1] – [4] моделі і метод управління зручністю використання програмних продуктів з врахуванням вимог користувачів. Надається характеристика основних компонентів системи, їх призначення та особливостей програмної реалізації.

Статья посвящена описанию программной системы, которая реализует рассмотренные в предыдущих работах автора [1] – [4] модели и метод управления удобством использования программных продуктов с учетом требований пользователей. Дается характеристика основных компонентов системы, их назначения и особенностей программной реализации.

використанні ПП. Виконання цих дій потребує управління. Оскільки ЗВ є характеристикою якості ПП, то управління ЗВ відбувається в рамках управління якістю ПП. При цьому досягнення ЗВ ускладнюється такими чинниками: протиріччям уявлень розробників та користувачів про властивості зручного у використанні ПП; необхідністю оптимізації досягнення ЗВ у процесі розробки ПП; відсутністю у технологіях створення ПП методів управління ЗВ.

Протиріччя уявлень розробників та користувачів про властивості зручного у використанні ПП спричинені їх суб'єктивністю. Попри конкретність задач, які мають вирішуватись ПП, існує значна свобода в тому, у якому вигляді вхідні та вихідні дані будуть надані користувачеві. При цьому важливою є не лише естетика користувацького інтерфейсу, але й забезпечення легкості оперування та доступності ПП, здійснення контролю над ПП та досягнення цілей і задач користувачів з ефективністю, економічністю й задоволеністю. Проте користувачі не завжди розуміють різницю між тим, що вони хочуть, і тим, що буде дійсно зручно у використанні. Саме тому при управлінні ЗВ потрібно виконувати дії, які б забезпечували в процесі розробки ПП прийняття рішень щодо зміни конкретних

показників ЗВ, при цьому враховуючи оцінку ЗВ користувачами.

Необхідність оптимізації досягнення ЗВ пов'язана із запобіганням ризику перевищення вартості трудовитрат на покращення ЗВ над економічним ефектом від виконаних змін. У процесі створення та супроводу ПП можуть виникати декілька варіантів зміни показників ЗВ, реалізація кожного з яких достатньою мірою вплине на збільшення задоволеності користувачів зручністю використання ПП. Тому важливою є задача визначення максимально можливих покращень із мінімальними трудовитратами, які враховуються у загальній вартості ПП.

Відсутність у керівництвах зі створення ПП методів забезпечення ЗВ, особливо для ітераційних моделей життєвого циклу, спричиняє труднощі в ефективній інтеграції процесів управління ЗВ та створення ПП. Як правило, це заважає на ранніх етапах створення ПП діагностувати проблеми ЗВ, а тому збільшує витрати на їх вирішення у майбутньому.

Наявні методи досягнення ЗВ стосуються окремих процесів управління ЗВ – планування та контролю. Тому особливої актуальності

набуває задача розробки методу та засобу управління ЗВ, які б дозволили забезпечувати ЗВ у процесі створення та супроводу ПП, враховуючи вищезгадані чинники.

У попередніх роботах автора [1 – 4] подано метод управління ЗВ і відповідні математичні моделі, розглянуто оптимізаційні та статистичні математичні методи, адаптовані для аналізу та розв'язання згаданих моделей, а також можливість програмної реалізації та оптимізації обчислень. Нижче дається опис архітектури та особливостей прототипу програмної системи (ПС) управління ЗВ, який реалізує вищезгаданий метод.

1. Функціональні вимоги до програмної системи та вибір архітектурного стилю

Програмна система має забезпечувати управління ЗВ в процесі створення ПП шляхом реалізації функціональних вимог, модель яких представлено за допомогою діаграми станів на рис.1.

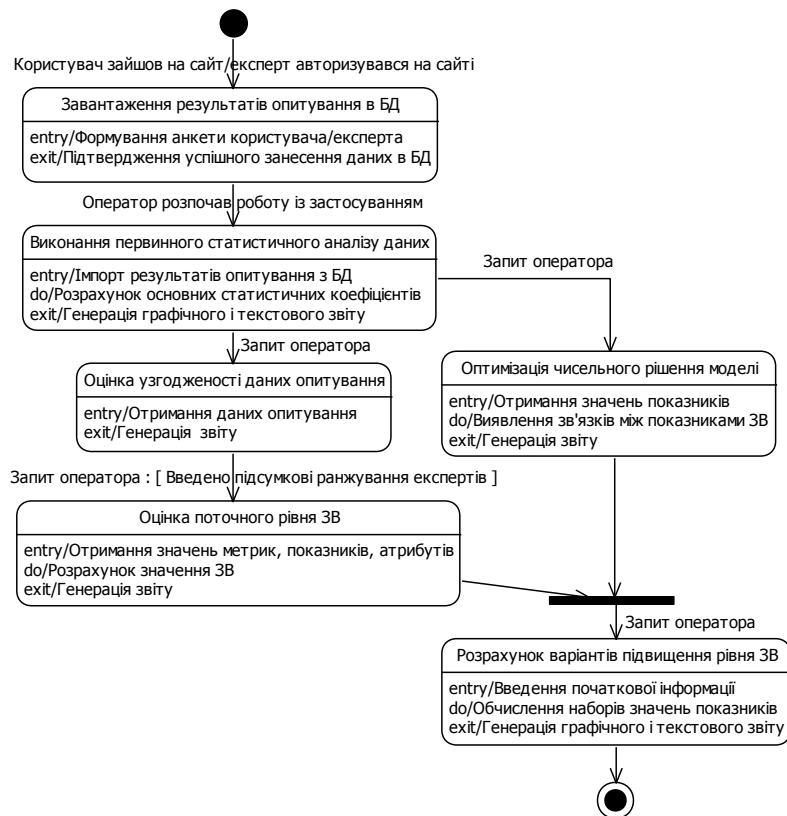


Рис.1 Модель функціональних вимог до програмної системи

Існуючі інструменти, найбільш відомі з яких описано в [8 – 11], не забезпечують реалізацію розробленого методу управління ЗВ. Тому постає задача створення відповідної програмної системи, функціональність якої наведено нижче.

Вимога завантаження результатів опитування в БД забезпечується наступними функціями системи:

- авторизація експерта;
- вибір шаблону форми загрузки даних;
- заповнення шаблону даними з БД;
- пересилання заповненого шаблону анкети користувачу/експерту;
- внесення даних опитування в БД;
- сповіщення користувача/експерта про успішність внесення анкетних даних.

Вимога виконання первинного статистичного аналізу даних реалізується за допомогою функцій системи:

- імпорт з БД оцінок (метрик) та рангів метрик, показників, атрибутів ЗВ;
- розрахунок математичного сподівання, медіани, дисперсії, коефіцієнтів ексцесу та асиметрії;
- побудова двовимірної гістограми розподілу значень оцінок та рангів.

Вимога оцінки узгодженості даних опитування забезпечується функціями системи:

- розрахунок коефіцієнта конкордації окремо для користувачів та експертів;

виконання кластерного аналізу на даних опитування експертів.

Вимога оцінки поточного рівня ЗВ виконується за допомогою функцій:

- введення підсумкових рангів метрик, показників, атрибутів ЗВ;
- розрахунок вагових коефіцієнтів метрик, показників, атрибутів ЗВ;
- розрахунок поточного рівня ЗВ.

Вимога оптимізації чисельного рішення моделі реалізується через наступні функції:

- розрахунок математичного сподівання, медіани, дисперсії, коефіцієнтів ексцесу та асиметрії для значень показників ЗВ;
- зміна закону розподілу згідно обраного математичного перетворення;
- розрахунок коефіцієнтів кореляції для всіх пар показників;
- вибір виду регресії для обраних пар показників;

- розрахунок коефіцієнту детермінації та рівняння регресії;
- побудова графіку лінії регресії.

Вимога розрахунку варіантів підвищення рівня ЗВ забезпечується функціями системи:

- введення формули залежності між показниками;
- введення значення бажаного рівня ЗВ;
- введення кроку перебору значень показників;
- обчислення наборів показників, які задовольняють умовам;
- побудова тривимірного точкового графіка, по осях: сума змін показників в розглядуваному варіанті, максимальна вага показника, початкове значення показника з максимальною вагою;
- при виборі точки на графіку виводиться інформація щодо назви показників, їх початкових значеннях та необхідної величини зміни.

Для реалізації вищевказаних функціональних вимог пропонується використати гібридну архітектуру (рис.2) програмного забезпечення, яка поєднує в собі елементи клієнт-серверної архітектури та архітектури, орієнтованої на БД.

Такий вибір обумовлений необхідністю централізованого збору даних (веб-сайт) та їх довготривалого зберігання (БД). Пропонується застосувати модель «товстого клієнта» для обробки даних на стороні клієнта-застосування, оскільки обробка є досить ресурсоемною та інтерактивною.

2. Основні характеристики системи

Згідно запропонованого в дослідженні методу створена програмна система забезпечує збір інформації щодо зручності використання програмних продуктів у вигляді оцінок користувачів та суджень експертів з наступною її обробкою. Мета – визначення ключових напрямів покращення характеристик ЗВ досліджуваного ПП. Програмна система управління ЗВ вирішує вказану задачу за допомогою автоматизації збору інформації на базі опитування користувачів та формування даних для експертного рішення на основі математичних моделей оцінки та забезпечення ЗВ.

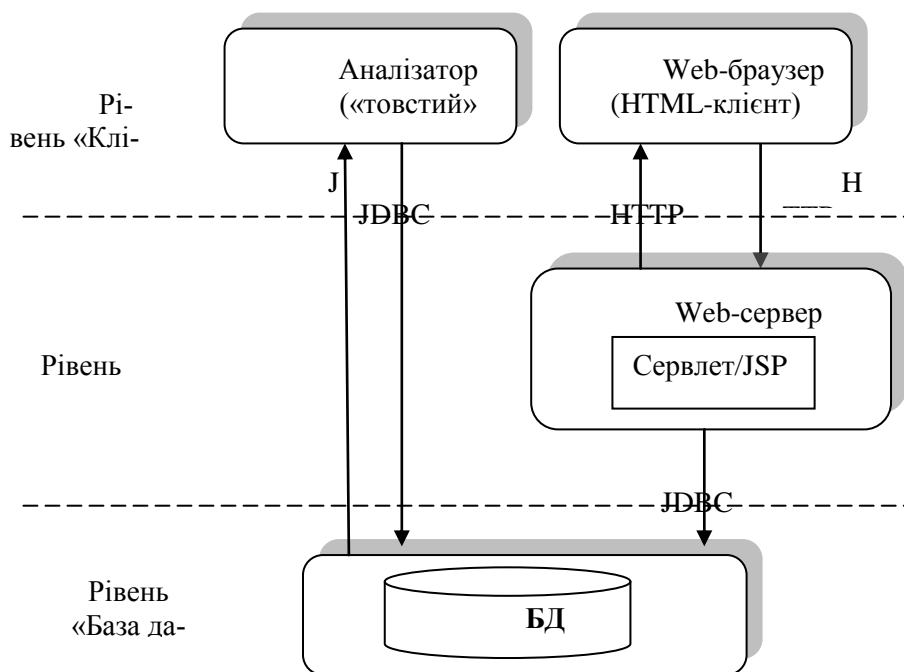


Рис.2 Архітектура програмної системи

Система складається з трьох основних частин:

1. Web-сервер, що забезпечує збір даних.
2. База даних, яка зберігає зібрану інформацію та налаштування системи.
3. Аналізатор, що забезпечує управління системою, аналіз даних та виведення результатів.

Web-сервер. При створенні web-серверу використовувалась технологія Java Server Pages (JSP). Відповідні сервлети, у які перетворюються JSP документи перед використанням, поміщені у web-контейнер Apache Tomcat (рис.3), що забезпечує обмін даними між сервлетом та користувачами, виконує функції створення програмного середовища для функціонування сервлета, ідентифікації та авторизації експертів, організації сесії для кожного з них. Вибір вказаного контейнера пояснюється відкритістю його коду, невеликим розміром проекту і легкістю установки та налаштування. Статичні вихідні дані JSP сторінки оформлені в форматі [HTML](#). JSP сторінки зберігаються в наступних файлах:

- Index.jsp. Це сторінка, яку за замовчанням веб-сервер відсилає браузеру, коли

користувач заходить на сайт, щоб заповнити анкету.

- Login.jsp. Це сторінка вибору режиму опитування «Користувач» або «Експерт». При виборі останнього передбачено введення логіну та паролю.
- Quiz.jsp. Це сторінка, де безпосередньо розміщені запитання анкети.
- Process.jsp. Це сторінка, яка з'являється після відправки даних користувачем та повідомляє про успішність.

Архітектура web-серверу є найпростішим однорівневим рішенням, коли JSP сторінка одночасно містить контролер обробки запитів, шаблон представлення і Java-компонент, який здійснює завантаження даних з сервера БД.

База даних. Як було зазначено, web-сервер забезпечує ефективний і надійний збір даних, їх перевірку і відправку на сервер. На сервері дані зберігаються в базі, що управляється MySQL. Вибір даної БД обумовлений особливостями розв'язуваної задачі управління ЗВ: порівняно малий обсяг збережених даних і невелике число з'єднань таблиць при виконанні запитів дають можливість використовувати цю ефективну й економічну щодо ресурсів систему.

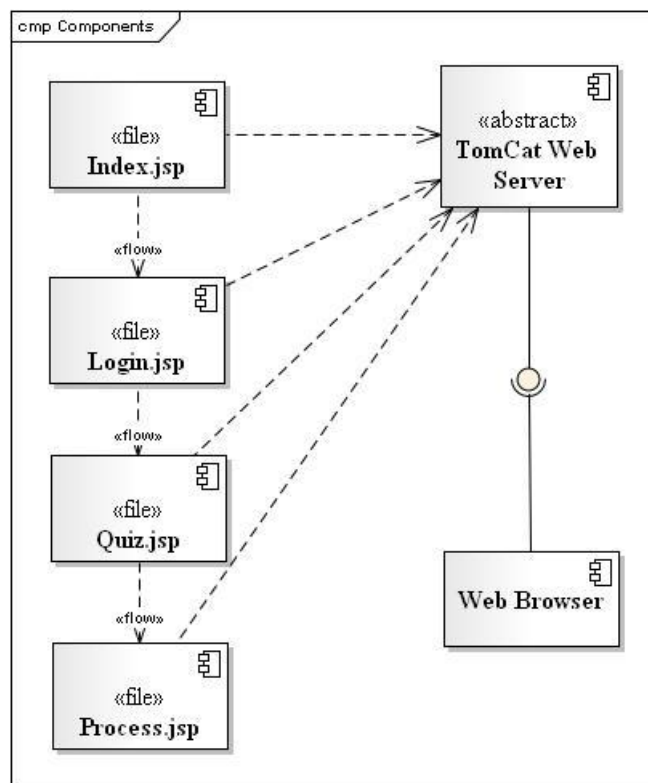


Рис.3 Компоненти web-серверу

БД містить таблиці чотирьох типів:

1. Таблиці зв'язків: `attribute_indicator_map`, `indicator_metric_map`, `metric_question_map`. Містять зв'язки між атрибутами і показниками, показниками і метриками, метриками та оцінками користувачів відповідно до ієрархічної моделі критеріїв ЗВ, обґрунтованої в другому розділі. Конкретизація вказаних зв'язків виконана в рамках практичної реалізації в четвертому розділі.
2. Таблиці ідентифікаторів `attributes_dict`, `indicator_dict`, `metric_dict`, `question_dict`, `experts_dict`, `users_dict` для атрибутів, показників, метрик, оцінок ЗВ, експертів та користувачів відповідно.
3. Таблиці числових оцінок та рангів, які виставляються критеріям ЗВ досліджуваного ПЗ, наприклад, `attributes_answers` (для користувачів) та `attributes_answers_exp` (для експертів).
4. Таблиця із загальними налаштуваннями («settings») програмного комплексу оцінки та управління ЗВ, такими як максимально допустиме число користувачів та експертів, що приймають участь в опитуванні, пароль експертів.

Аналізатор. Аналізатор реалізовано у вигляді віконної інтерактивної системи, яка забезпечує налаштування всієї програмної системи, вибірку даних з БД, їх аналіз в напівавтоматичному режимі і виведення результатів. При розробці застосовувалися мови Java і C++. Вибір Java визначається високою ефективністю реалізації графічних користувацьких інтерфейсів і підсистеми доступу до БД. C++ використовується, оскільки, за інших рівних умов, забезпечує вищу швидкодію обчислювальних алгоритмів. Міжмовна взаємодія досягається за допомогою технології JNA.

Фізичні компоненти аналізатора зображено на рис.4. Можна виділити наступні підсистеми аналізатора:

- підсистема завантаження та первинного аналізу даних опитування;
- підсистема оцінки ЗВ та узгодженості відповідей користувачів/експертів;
- підсистема визначення залежностей між показниками ЗВ;
- підсистема розрахунку варіантів забезпечення ЗВ.

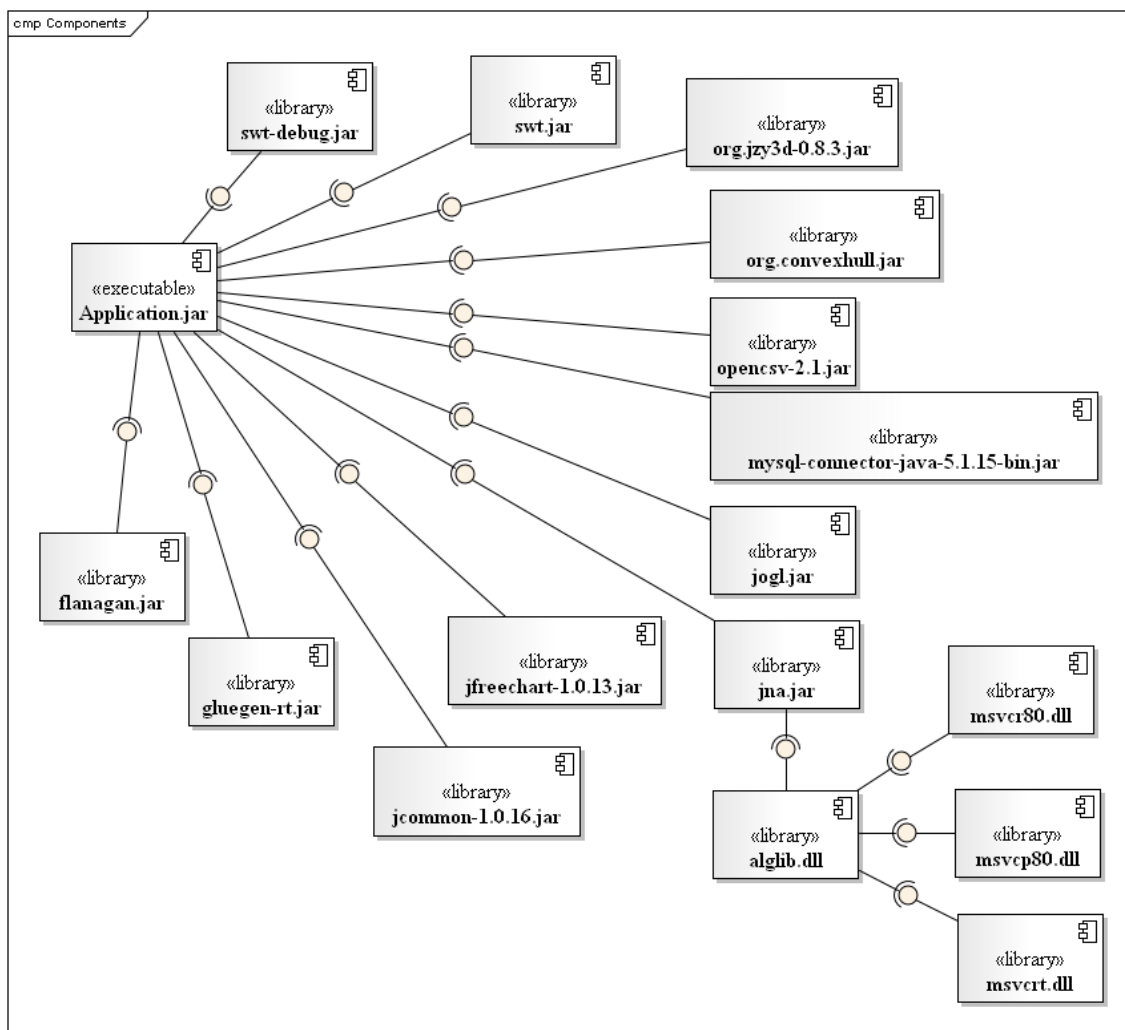


Рис.4 Фізичні компоненти аналізатора

Інтерфейсна частина аналізатора відділена від моделі даних і містить елементи інтегрованого контролера. З метою зручності доступу до класу ModelController, який фактично містить модель даних та забезпечує доступ до неї для всіх компонентів програмної системи, а також для гарантії єдності моделі даних використовується породжуючий шаблон проектування Singleton.

Можливі варіанти використання програмної системи представлено діаграмою прецедентів на рис.5.

3. Підсистеми аналізатора

Збір даних та підсистема завантаження і первинного аналізу даних опитування. Для вирішення питання відповідності програмного продукту заданому рівню ЗВ на основі

відгуків користувачів необхідно збирати, зберігати та аналізувати їх оцінки.

У рамках дослідження інформація щодо задоволеності користувачів отримується за допомогою опитування – найбільш розповсюдженого методу збору первинних даних. Опитування зводиться до ряду закритих питань, що передбачають кількісні відповіді в порядковій та інтервальній шкалі.

Особливості програмної реалізації Web-сервісу було описано вище. На рис.6 діаграмою послідовності зображено взаємодію користувача та програмної системи на етапі збору даних. Шаблон анкети, який заповнюється даними з бази та пересилається користувачу, містить набір атрибутів, показників, метрик і питань для оцінки метрик відповідно до використовуваної ієрархічної моделі ЗВ.

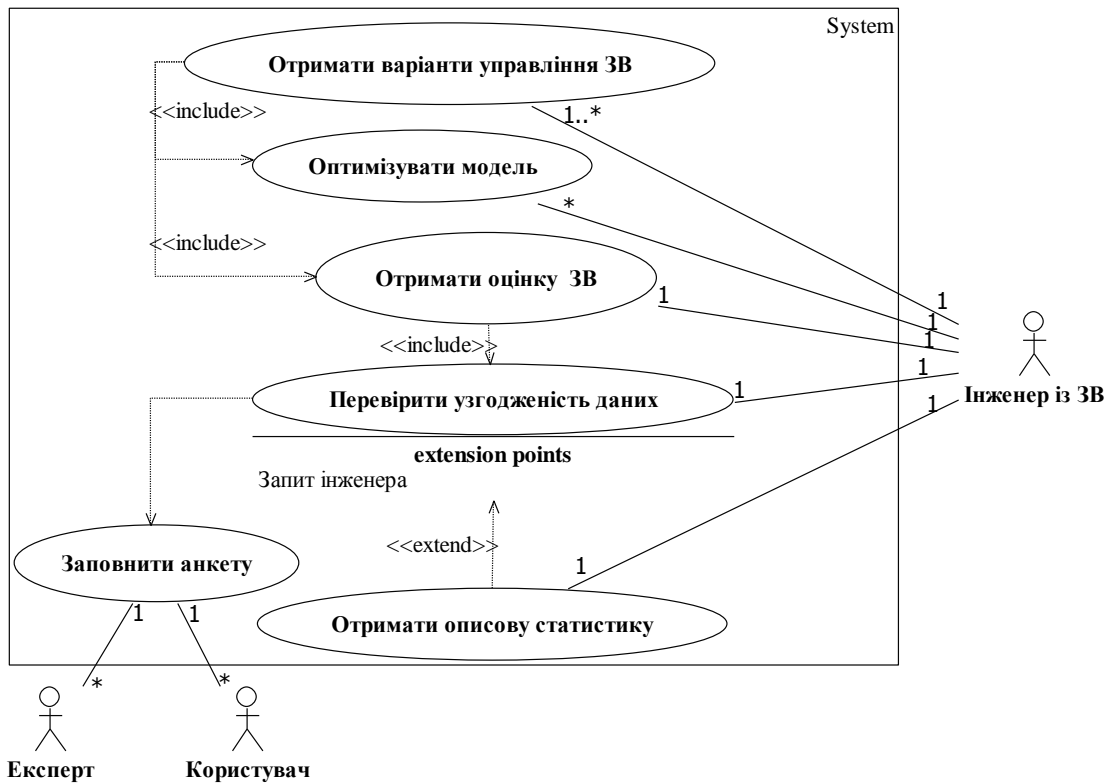


Рис.5 Варіанти використання програмної системи

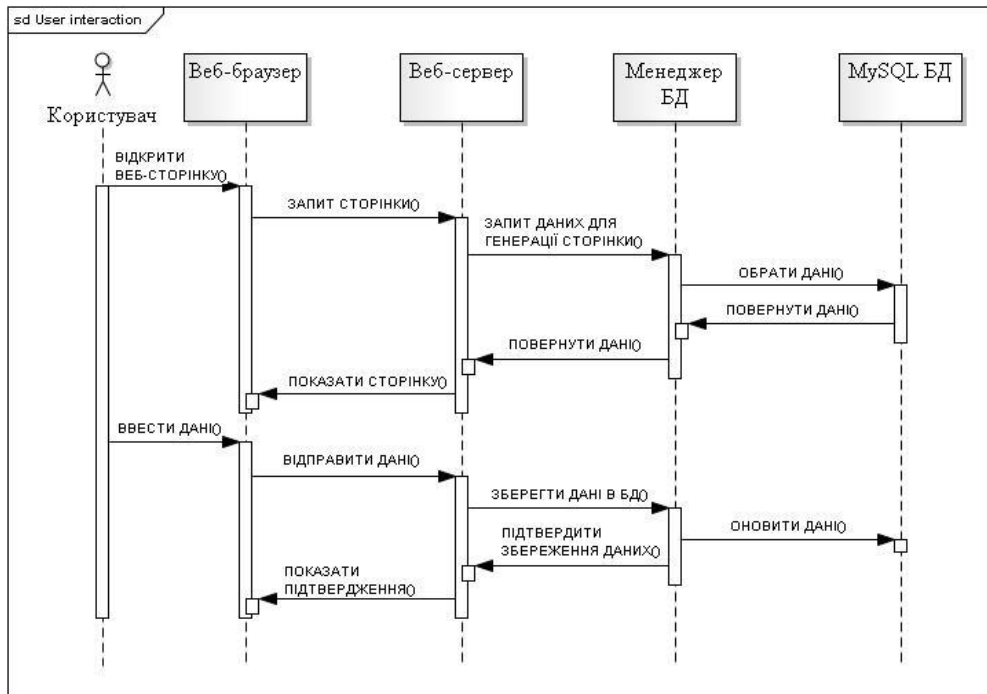


Рис.6 Побудова зв'язків між показниками ЗВ

Коли опитування проведено, інженер зі ЗВ може приступати до первинної обробки даних, використовуючи

підсистему завантаження і первинного аналізу даних.

Вкладка «Описова статистика» (рис.7) призначена для перегляду закону розподілу оцінок та рангів, а також обрахунку основних статистичних коефіцієнтів: математичного сподівання, медіани, дисперсії, коефіцієнтів ексцесу та асиметрії. Інженер із ЗВ повинен натиснути кнопку «Аналіз». Підсистема через компоненти інтегрованого контролеру, які містить інтерфейс, отримує з моделі (клас ModelController) ранги метрик, показників, атрибутів та оцінки критеріїв ЗВ і, використовуючи клас LowLevelStatAlgorithmsWrapper та бібліотеки jfreechart-1.0.13 [12] і alglib [13], виконує розрахунки статистичних коефіцієнтів і будує двовимірну гістограму розподілу значень оцінок та рангів. Результати виводяться у числовій і графічній формі відповідно. Міри центральної тенденції характеризують центральне значення, навколо якого розподілені значення відповідей користувачів. Якщо закон розподілу величин близький до нормального, то для його опису добре підходить математичне

сподівання (перший момент розподілу). Якщо ж розподіл має досить довгі або широкі хвости, то для оцінки «центрального» значення бажано використовувати медіану, оскільки математичне сподівання вибірки може дуже повільно збігатися до «істинного» математичного сподівання розподілу. Дисперсія є мірою розсіювання і характеризує розкид, з яким випадкова величина розподіляється навколо свого центрального значення.

Міри форми розподілу дозволяють проаналізувати основні риси «зовнішнього вигляду» розподілу: ступінь асиметрії відносно центрального значення (коефіцієнт асиметрії) та ступінь крутизни (коефіцієнт ексцесу) порівняно з нормальним розподілом.

Переглянувши вищезгадану статистику, інженер зі ЗВ може зробити первинний висновок щодо якості зібраних даних і в разі незадовільних результатів сповістити про необхідність повторного збору та корекції даних.

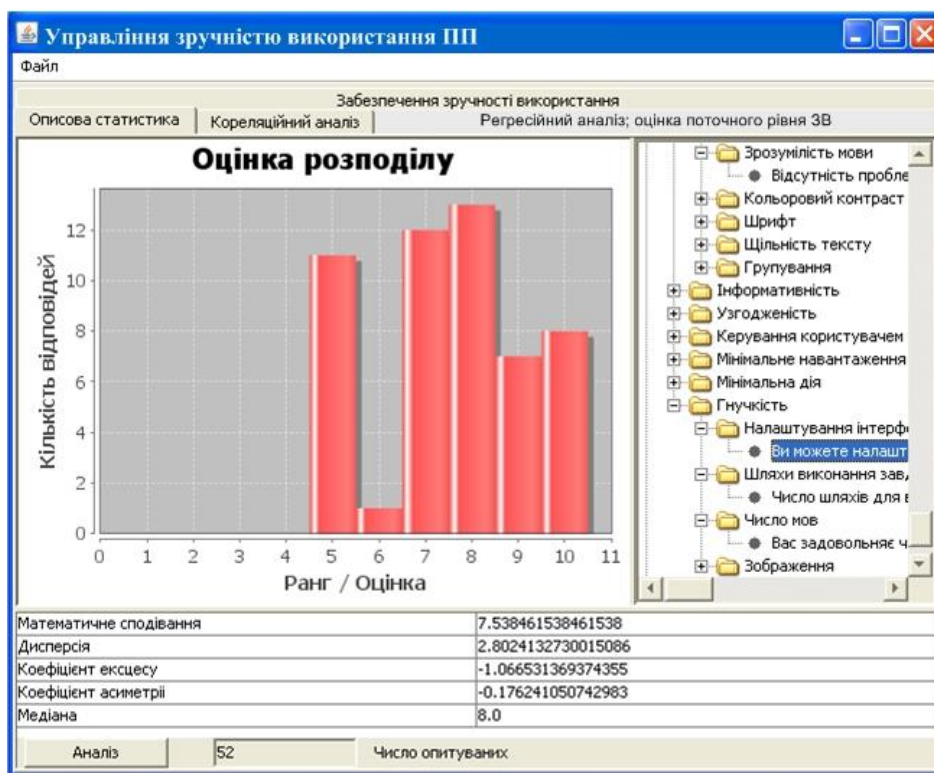


Рис.7 Інтерфейс описової статистики

Підсистема оцінки ЗВ та узгодженості відповідей користувачів/експертів.

У роботі [2] було показано, що при формальній постановці задачі забезпечення ЗВ використовуються експертні ранжування для обчислення ваги метрик, показників та атрибутів ЗВ. У рамках програмної реалізації вирішується задача перевірки узгодженості ранжувань експертів, обчислення ваги критеріїв ЗВ, а також поточного рівня ЗВ згідно запропонованого методу. Виконання відповідних функцій системою ініціюється інженером зі ЗВ через користувацький інтерфейс, а саме: інженер має вказати очікуване число кластерів та натиснути кнопку «Аналіз». Підсистема через компоненти інтегрованого контролеру, які містить інтерфейс, отримує з моделі ранги метрик, показників і атрибутів та, використовуючи клас `LowLevelStatAlgorithmsWrapper`, виконує розрахунки. Результати виводяться у формі звіту, у якому відображається коефіцієнт конкордації та розподіл

експертних ранжувань по кластерах (рис.8). При достатній узгодженості думок експертів передбачено введення результуючих ранжувань по атрибутах, показниках та метриках окремо (рис.9).

Після введення ранжування, підсистема виконує розрахунок вагових коефіцієнтів і поточного рівня ЗВ програмного продукту, що аналізується. Результати в числовому вигляді відображаються у формі звіту, після чого управління повертається до користувацького інтерфейсу. Якщо поточний рівень ЗВ співпадає або більше, ніж заданий, інженер із ЗВ може завершити роботу із аналізатором.

Підсистема визначення залежності між показниками ЗВ. Одним з важливих питань при вирішенні задачі оптимізації ЗВ є врахування залежних змінних, тобто показників. Підсистема побудови зв'язків всередині математичної моделі забезпечення ЗВ дозволяє вирішити відповідні задачі визначення залежних показників та побудови лінії регресії, видачі емпіричної залежності.

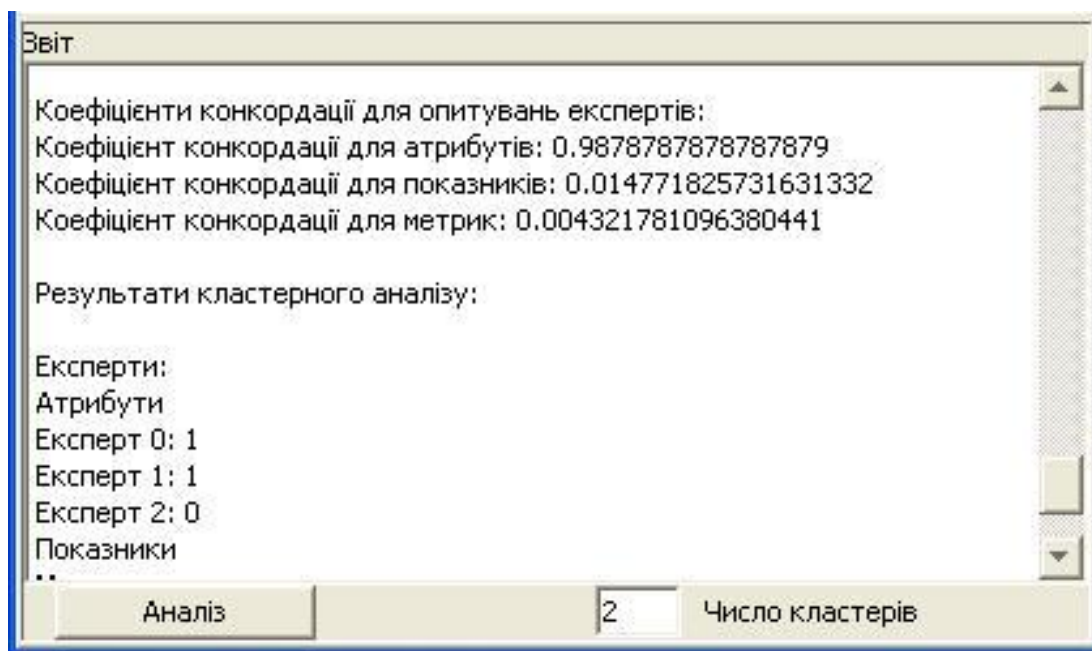


Рис.8 Інтерфейс оцінки узгодженості ранжувань експертів

Розрахунок вагових коефіцієнтів	
Атрибути	
Назва	Підсумковий ранг
Можливість навчання	2
Доступність	5
Операбельність	3
Показники	
Назва	Підсумковий ранг
Правильність	3
Відгук	4
Узгодженість	5
Метрики	
Назва	Підсумковий ранг
Візуальна когерентність	4
Видимість завдання	2
Число зображень	3

Рис.9 Інтерфейс введення експертних ранжувань

Як і в попередніх випадках виконання відповідних функцій системою ініціюється інженером зі ЗВ через користувацький інтерфейс. Інженер може виконувати наступні дії:

1. Перегляд закону розподілу значень показників ЗВ.
2. Вибір та застосування відповідних математичних перетворень при необхідності зведення закону розподілу до нормального.
3. Аналіз результатів розрахунку коефіцієнтів кореляції.
4. Введення номерів залежних показників та вибір форми регресії. При необхідності – введення початкових значень коефіцієнтів та кроку.
5. Аналіз отриманої емпіричної залежності та вибір іншої форми регресії (при незадовільній точності наближення).

Підсистема через компоненти інтегрованого контролеру, які містить інтерфейс, отримує значення оцінок користувачів та виконує розрахунки. Результати виводяться у формі звітів, що містять коефіцієнти кореляції та статистику, яка характеризує точність наближення, також відбувається побудова графіку лінії регресії. Зауважимо, що після виконання інженером із ЗВ дії зведення закону розподілу до нормального (фактично змінює значення показників ЗВ),

підсистема виконує збереження змін в базі даних.

Підсистема розрахунку варіантів забезпечення ЗВ. Дана підсистема виконує основне функціональне призначення після підсистеми оцінки поточного рівня ЗВ. За коректність та ефективність роботи підсистеми розрахунку варіантів забезпечення ЗВ відповідають усі інші підсистеми. Послідовність кроків винесення рішення та алгоритми, на якому воно ґрунтується, детально описані в другому розділі.

Підсистема вирішує задачі перебору простору показників ЗВ із заданим кроком та отримання варіантів зміни k показників ($k=5$) ЗВ, які забезпечують досягнення бажаного рівня ЗВ згідно розробленого методу. Також реалізовано функцію введення формули залежності між показниками, яка враховується при виборі оптимального варіанту забезпечення ЗВ. Програмно реалізовано наступні можливості роботи з підсистемою:

1. Введення формули залежності одного показника від іншого.
2. Введення кроку перебору та бажаного значення ЗВ.
3. Перегляд варіантів зміни показників (рис.10).

Для графічного відображення використано бібліотеку візуалізації

тривимірних розподілів на основі OpenGL - Jzy3D [14], яка, в свою чергу, використовує JOGL. По осях відкладено: максимальна вага показника в поточному варіанті; сума змін показників в розглядуваному варіанті, яка забезпечує досягнення заданого рівня зручності використання; початкове значення показника (до змін), який має максимальну вагу в даному варіанті.

Висновки

Для реалізації запропонованого в попередніх роботах автора методу визначено функціональні вимоги та розроблені архітектура й прототип програмної системи, яка дозволяє автоматизувати управління ЗВ в процесі створення та супроводу ПП. Архітектура поєднує в собі елементи клієнт-серверної архітектури та архітектури, орієнтованої на БД.

Розглянуто складові програмної системи та їх компоненти, описано задіяні технологічні рішення. Система складається з трьох основних частин: web-сервер, що забезпечує збір даних; база даних, яка зберігає зібрану інформацію та налаштування системи; аналізатор, що забезпечує управління

системою, аналіз даних та виведення результатів.

Розроблено й описано підсистеми аналізатора, який входить в програмну систему, їх призначення та особливості програмної реалізації.

Список літератури

1. Сидоров М. О. Оцінка зручності застосування програмного забезпечення в контексті Agile-розробки / М. О. Сидоров, І. В. Гученко // Наукоємні технології. – 2009. – № 4(4). – С.64-68.
2. Гученко І. В. Аналіз експертної бази на узгодженість методами Data Mining / І. В. Гученко // Вісник Київського Національного університету ім. Тараса Шевченка. Серія: Фізико-математичні науки. – 2011. – № 1. – С. 119-122.
3. Гученко І. В. Математична модель забезпечення та оцінки практичності програмного забезпечення / І. В. Гученко // Вісник НАУ: Науковий журнал. – 2011. – № 4. – С. 83-90.
4. Гученко І. В. Зручність застосування програмного забезпечення: атрибути та метрики / І. В. Гученко // Інженерія програмного забезпечення. – 2010. – №2. – С. 5-13.
5. Dubey S. K. Analytical Roadmap to Usability Definitions and Decompositions / Sanjay Kumar Dubey, Ajay Rana // International Journal of Engineering Science and Technology. – 2010. – № 2(9). – P. 4723-4729.

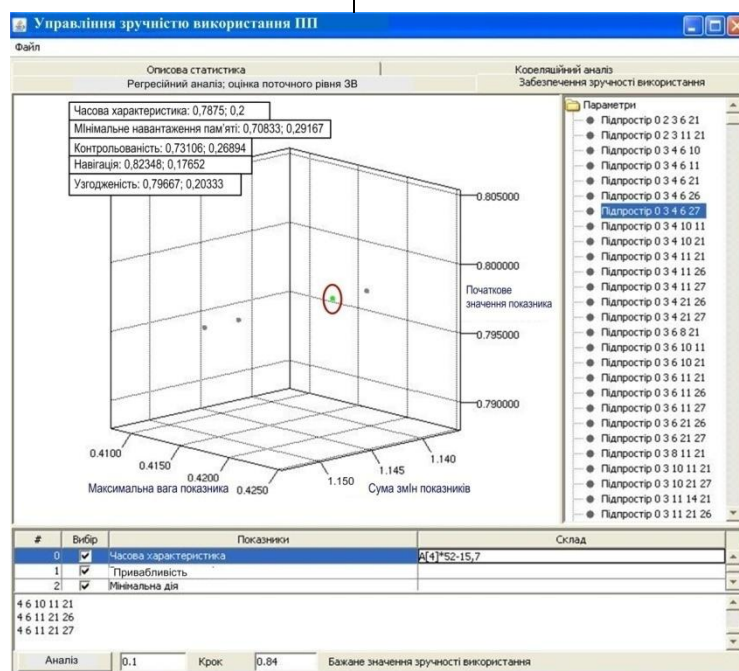


Рис.10 Графічне зображення варіантів зміни показників ЗВ

6. Ergonomic Requirements for Office Work with Visual Display Terminals, Part 11: Guidance on Usability: ISO 9241-11. – Geneva: International Organization for Standardization, 1998. – 22p.
7. Systems and software Quality Requirements and Evaluation, System and software quality models: ISO/IEC 25010:2011. – Geneva, 2011. – 34p.
8. Gray W. Project Ernestine: Validating a GOMS Analysis for Predicting and Explaining Real-World Task Performance / W. Gray, B. John, M. Atwood // Human-Computer Interaction. – 1993. – Vol. 8. – P. 237-309.
9. Kieras D. An overview of the EPIC architecture for cognition and performance with application to human-computer interaction / D. Kieras, D. Meyer // Human-Computer Interaction. – 1997. – Vol. 12. – P. 391-438.
10. Padda Harkirat K. QUIM: A Model for Usability/Quality in use Measurement / Harkirat K. Padda. – Colne, 2010. – 124 p.
11. Ivory M.Y. The State of the Art in Automating Usability Evaluation of User Interfaces / M. Y. Ivory, M. A. Hearst // ACM Computing Surveys. – 2001. – Vol. 33(4). – P. 470-516.
12. Java chart library. [Електронний ресурс]. – Режим доступу: <http://www.jfree.org/>
13. Alglib. Алгоритмы описательной статистики. [Електронний ресурс]. – Режим доступу: <http://alglib.sources.ru/statistics/descriptive.php>
14. A java library for scientific 3d plotting. [Електронний ресурс]. – Режим доступу: <http://code.google.com/p/jzy3d/>

Сведения об авторе



Гученко Инна Владимировна - асистент кафедри інженерії програмного забезпечення Національного авіаційного університету. Наукові інтереси: оцінка зручності використання ПЗ, UML і проектування ПЗ, системи систем ПЗ.

E- mail: inna.zapozhets@livenau.net

Стаття надійшла до редакції 01.12.2011р.