

УДК 004.5

## ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПОВЕДІНКИ КОРИСТУВАЧА В ЗАДАЧІ СТВОРЕННЯ ОПТИМАЛЬНОГО ІНТЕРФЕЙСУ

*Гученко М.І., к. т. н., доц., Пшеничний Б.Є., студ., Іванова М. М., ас.*

*Кременчуцький державний політехнічний університет*

*імені Михайла Остроградського*

*39614, м. Кременчук, вул. Першотравнева, 20*

*E-mail: mig@ktp.polytech.poltava.ua*

Розкрито вплив користувацького інтерфейсу на якість роботи користувача. У процесі дослідження створено ігрову навчальну програму, проведено експеримент та отримано дані для охарактеризування дій користувача в задачі створення оптимального інтерфейсу. Користувач виконує безліч дій в процесі роботи, тому всю задачу було розділено на підзадачі такі, як зміна розташування та розміру вікон, для яких можливо створити адекватну математичну модель. Поведінка користувача в цій задачі є нестационарний неергодичний випадковий процес. Визначено, що для побудови математичної моделі необхідно застосовувати апарат нейронних мереж.

**Ключові слова:** користувацький інтерфейс, апарат нейронних мереж.

Influence of the user interface on quality of work of the user is opened. During research it is created the game curriculum, experiment is carried out and it is received the data for the characteristic of actions of the user in a task of creation of the optimum interface. The user performs set of actions during work, therefore all task has been divided into subtasks such, as change of an arrangement and the size of windows for which it is possible to create adequate mathematical model. The behaviour of the user in this task is non-stationary casual process. It is determined, that it is necessary to apply the device of neural networks to construction of mathematical model.

**Key words:** user interface, device of neural networks

**Вступ.** В умовах постійного росту кількості розроблених програм і зміни їх версій, актуальною є проблема побудови ефективного інтерфейсу користувача (ІК), який дозволяє скоротити час на навчання користувачів роботі з новим програмним забезпеченням (ПЗ), підвищити якість і ефективність їх роботи. Зусилля розробників зосереджені на створенні оптимального інтерфейсу в процесі проектування ПЗ. Однак кожний користувач має свої звички, манеру роботи, уподобання, смаки, тощо. Це унеможливає розробку оптимального інтерфейсу на етапі проектування і викликає необхідність в настроюванні ІК в процесі експлуатації програмного продукту.

Сучасне ПЗ передбачає можливість індивідуального налаштування інтерфейсу для кожного користувача, однак таке налаштування може зайняти багато часу. Якщо ж за одним комп'ютером працює декілька користувачів, то постає питання налаштування інтерфейсу прикладних програм ще й після зміни операторів. Оперативних програмних засобів для підстройки ІК в реальному часі під конкретного користувача практично не існує. Одним зі шляхів вирішення проблеми є створення інтерфейсного агента (ІА) для налаштування ІК під визначеного користувача в реальному часі з урахуванням поточного стану інтерфейсу системи [1]. Важливою складовою такого агента має бути математична модель поведінки користувача, яка в процесі налаштування інтерфейсу заміняє реального користувача в обчислювальному середовищі.

Побудова математичної моделі поведінки користувача є складною задачею. Кожен користувач і ко-

жна прикладна програма мають індивідуальні особливості. Побудова загальної моделі поведінки користувача, придатної для всіх ситуацій, практично неможлива. Але, навіть, якби така модель була побудована, то вона була б надто ресурсоємною в обчислювальному плані, що неприпустимо, оскільки агент має налаштовувати інтерфейс значно швидше від реального користувача. Виходом із ситуації може бути декомпозиція загальної моделі на часткові, призначені для вирішення окремих задач. Однак і часткові моделі на теперішній час практично відсутні. Для синтезу вказаних моделей потрібні експериментальні дослідження поведінки користувача.

**Аналіз літературних джерел.** Розробники програмних комплексів схильні розглядати функціональність системи окремо від ІК. Вважається, що ІК є лише доповненням до функціональності системи. Користувачі ж програм, як правило, не розділяють функціональність системи й ІК. Для них ІК і є програмою. Наявність зручного інтерфейсу визначає позитивне відношення до всієї програми.

ІК часто розуміють тільки як зовнішній вигляд програми. Однак таке розуміння ІК є занадто вузьким, бо користувач сприймає через ІК систему в цілому. У дійсності ІК містить у собі всі аспекти дизайну, які впливають на взаємодію користувача й системи.

Для більшості систем на розробку ІК йде значна частка бюджету й зусиль програмістів. Наявні статистичні дані [2] стверджують, що:

- код ІК становить від 47 до 60 % коду всієї програми;
- на розробку ІК йде більше 29 % проектного

бюджету й у середньому 40 % усіх зусиль розробників зі створення системи.

Погано розроблений інтерфейс різко обмежує функціональність системи в цілому. Можна виділити наступні причини важливості зручного ІК [2]:

1. Підвищення конкурентоспроможності.

Розробка продукту, більш простого й зручного у використанні, дозволяє одержати значну перевагу в конкурентній боротьбі. Існують приклади, коли перероблення інтерфейсу, проведене з дотриманням вимог ергономіки збільшило доходи виробника на 80 %. При цьому користувачі завжди оцінюють простоту використання продукту як найважливішу його характеристику.

2. Зниження вартості впровадження.

Вартість впровадження та підтримки програмного продукту може становити 80 % від вартості розробки. Це пояснюється, у першу чергу, нерозумінням програмістами цілей і очікувань кінцевих користувачів продукту. Проблеми починають виникати відразу після вводу програмного продукту в експлуатацію. За рахунок локалізації проблем ІК на ранніх етапах розробки майже завжди можна знизити витрати на 60-90 %.

3. Збільшення аудиторії продукту.

Інтернет переповнений ресурсами, на яких доступні програмні розробки, які не відповідають вимогам до якісного інтерфейсу, створені без урахування потреб відвідувачів і тому відштовхують потенційну аудиторію. Такі програми складні у використанні. Використовуючи при розробці інтерфейсу програми підхід орієнтований на потреби користувача, можна підвищити рейтинг програмної розробки серед відвідувачів ресурсу.

4. Збільшення задоволеності користувачів.

Залучення майбутніх користувачів у процес розробки ПЗ та сайтів із застосуванням технік забезпечення якості інтерфейсу підвищує суб'єктивну задоволеність користувача в середньому на 40 %;

5. Зменшення витрат на навчання й підтримку користувачів.

Використання методів підвищення якості ІК при проектуванні ПЗ значно знижує час, необхідний для навчання користувачів і необхідні ресурси технічної підтримки. У середньому використання таких методів при проектуванні продукту знижує час навчання на 25 %, а кількість звернень до служби технічної підтримки – на 60 %.

Системи, розроблені з урахуванням вимог якості, - ергономічні. Вони працюють саме так, як очікують користувачі, і дозволяють їм зосередитися на власних завданнях, а не особливостях взаємодії із системою. Ергономічні програмні продукти простіше вивчити, вони більш ефективні, вони також дозволяють мінімізувати кількість людських помилок і збільшити суб'єктивну задоволеність користувачів. Але це не відбувається саме по собі. Ефективний інтерфейс є результатом усвідомлення розробником необхідності приділити значну увагу не тільки даним, з якими буде працювати користувач, але й діям, властивим користувачеві.

Основний внесок у дослідження проблеми побудови якісного ІК внесли російські веб-дизайнери й розробники В. Головач, М. Покровський, А. Лебедев, а також американські вчені J. Nielsen, T. Landauer, D. Norman.

**Мета роботи.** Отримання експериментальних даних, необхідних для теоретичного узагальнення поведінки користувача і синтезу математичної моделі користувача в задачі побудови оптимального інтерфейсу.

**Матеріал та результати дослідження.** Типовими операціями користувача з налаштування інтерфейсу прикладної програми в операційній системі Windows є зміна висоти та ширини вікна, зміна положення вікон, додавання та видалення панелей інструментів та їх розташування.

Задача настройки інтерфейсу є складною. Модель, що описує таку поведінку, очевидно буде дуже громіздкою. Тому в даній роботі була виконана декомпозиція задачі і як часткову задачу настроювання інтерфейсу обрана задача розташування вікон провідника Windows на екрані.

При проведенні експериментальних досліджень поведінки користувача в задачі створення оптимального інтерфейсу необхідно було оцінити якість виконання користувачем поставленого завдання, кількість його помилок та час виконання завдання залежно від якості ІК. Для проведення експериментальних досліджень було вирішено не використовувати якусь конкретну існуючу прикладну програму, а розробити спеціальну тестову програму. Це дає можливість оперативно вносити зміни до методики експерименту та завдання, що виконується. З метою зацікавлення користувача у роботі, яку йому належало виконати в процесі експерименту, тестова програма була побудована у формі гри. Іншою причиною є існування розвинутої теорії ігор, що може стати у пригоді при побудові математичної моделі користувача.

Ігровим середовищем є файлова система комп'ютера. Завданням користувача в розробленій грі є сортування файлів із назвами підвидів у тематичних каталогах, із назвами видів та надвидів загальнозрозумілих лінгвістичних понять. Структура каталогів гри зображена на рис. 1.



Рисунок 1 – Структура каталогів гри

Для формального опису інтерфейсу була вибрана система координат, показана на рис. 2. Кожне вікно описувалось вектором, який складався з чотирьох чисел – координат  $x$ , у верхнього лівого кута, висоти

$h$  та ширини вікна  $w$  у пікселях. Роздільна здатність моніторів під час досліду  $1024 \times 768$  пікселів. Інтерфейс у цілому описувався вектором  $I = \{x_1, y_1, w_1, h_1, x_2, y_2, w_2, h_2, \dots, x_n, y_n, w_n, h_n\}$  де  $x_1, y_1, x_2, y_2, \dots, x_n, y_n$  - координати верхнього лівого кута вікон,  $w_1, h_1, w_2, h_2, \dots, w_n, h_n$  - ширина та висота вікон,  $n$  - кількість вікон.

До вектора  $I$  заносилися тільки вікна, що не були мінімізовані. Під час гри вектор  $I$  фіксувався (через кожні 0,5 с. записувався у файл).

При кожному старті гри вікна інтерфейсу розташовувались на дисплеї у випадковому порядку, який визначався кількістю вікон та вектором  $I$ . Розміри та положення вікон ( $I$ ) було отримано за допомогою генератора випадкових чисел, вони були різних розмірів та перекривали одне одного.

Перед грою проводився інструктаж користувача. Нижче наведено текст завдання:

«Метою гри є правильне розташування файлів по відповідних папках. Імена файлів є загальновідомими видами живого царства або географічними назвами (наприклад, «Ворона.txt» або «Австралія.txt»). Імена папок є узагальнюючими поняттями відповідно до імен файлів, наприклад, «Живе-Неживе», «Суша-Вода» тощо. Папки утворюють ієрархічну структуру (рис. 1). На початку гри файли розташовані в папках у довільному порядку на будь-якому рівні ієрархії. Якщо, наприклад, файл називається «Австралія.txt», то його необхідно перемістити з папки, де він знаходиться на початку гри, в папку C:\Не\_живе\Суша\Материки.

При старті гри вікна розташовані в довільному порядку, їх розміри й розташування також є випадковими. Спочатку Ви повинні створити собі зручний інтерфейс, тобто розташувати вікна в зручному для Вас положенні та налаштувати їх розміри, а вже потім переміщати файли по тематичних папках відповідно до їх назв.

Отримана Вами оцінка буде тим вищою, чим менше Ви зробите помилок і чим швидше виконаєте завдання.

Для старту гри натисніть на кнопку «Старт», а для закінчення - «Готово»».

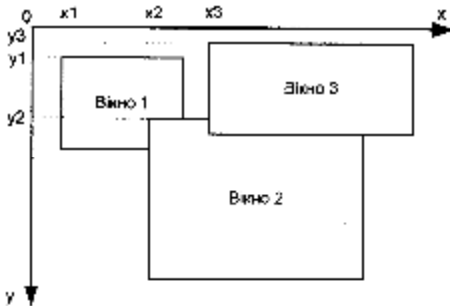


Рисунок 2 – Приклад розміщення вікон при старті гри

Таким чином, користувачеві пропонувалося спочатку налаштувати положення та розміри вікон

(створити оптимальний інтерфейс), а потім вирішити прикладну задачу – розсортувати файли за каталогами.

В експерименті приймали участь 7 операторів (студенти першого курсу Кременчуцького державного політехнічного університету за напрямком «Комп'ютерні системи та мережі»). Рішення про закінчення гри приймав користувач за власною суб'єктивною оцінкою. Після того, як користувач закінчував гру, проводився аналіз відповідності розташування файлів за каталогами. Помилкою вважалося розміщення файлу в невірному каталозі.

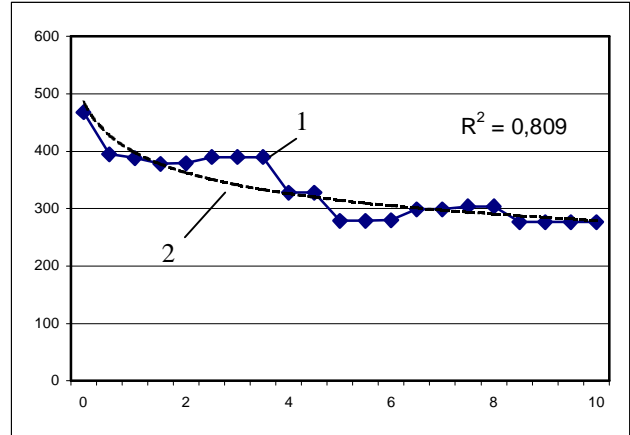


Рисунок 3 – Зміна висоти вікна з плином часу (1- залежність зміни висоти вікна від часу, 2- степенева лінія тренду, R – коефіцієнт достовірності апроксимації)

Після закінчення гри кількість помилок та час виконання завдання автоматично записувалися до файлу.

За отриманими даними було побудовано графік зміни висоти вікна з плином часу (рис. 3).

На рис. 3 зображено приклад зміни лише однієї характеристики вікна, бо вона є найбільш типовою. Графіки зміни інших характеристик досить схожі. Аналіз отриманих реалізацій дозволяє зробити висновок, що дії користувача при налаштуванні інтерфейсу за деяким окремим параметром є безперервним, нестационарним кусково-гладким випадковим процесом, причому довжина гладких ділянок змінюється випадковим чином. Виходячи з того, що асимптотична лінія тренду з плином часу прагне до константи можна зробити висновок, що для середнього розміру вікна існує деякий оптимальний середній розмір вікна. При відкритті відразу 4-х вікон усі користувачі намагалися розташувати їх таким чином, щоб усі вони були видимими на екрані одночасно, мали по можливості максимальний розмір та не перекривали одне одного. Це дає підстави вважати, що такий інтерфейс є оптимальним для даної задачі.

На рис. 4 та 5 показано залежність часу виконання завдання гри та кількості помилок від числа спроб відповідно. Графіки побудовані за результатами роботи двох користувачів, які показали найгірший та найкращий результати.

Можна помітити, що професійні навички та індивідуальні особливості користувачів мають вплив

на результати виконання завдання. З рис.4 можна зробити висновок, що, хоча різниця в часі досягнення кінцевого результату гри досить велика, але динаміка навчання у обох операторів досить подібна – лінії регресії однотипні, коефіцієнти достовірності апроксимації одного порядку.

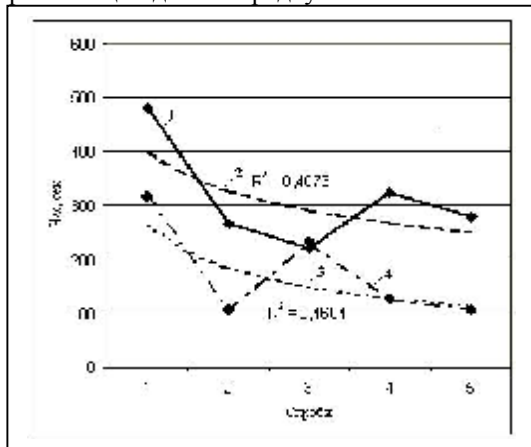


Рисунок 4 – Залежність часу проходження гри від числа спроб для першого (1) та другого (4) користувачів, 2 та 3 – степеневі лінії тренду, R – коефіцієнт достовірності апроксимації

Це дає надію на можливість створення загальної математичної моделі користувача в даній задачі. У деяких користувачів на початку була велика кількість помилок, у інших – багато часу затрачувалося на виконання завдання гри при меншій кількості помилок. Але при збільшенні числа спроб користувачі за кількістю помилок виходили на приблизно однакові кінцеві показники (рис. 5).

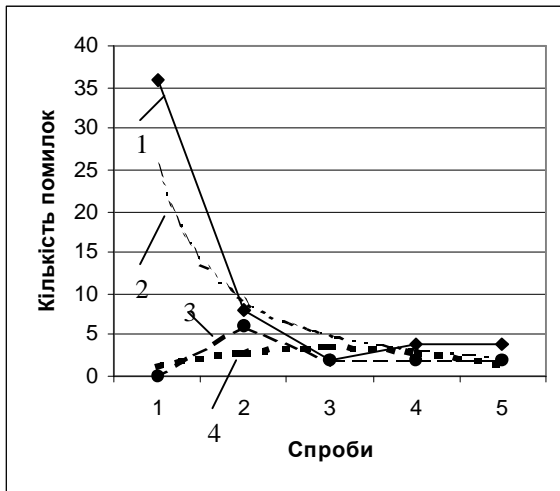


Рисунок 5 – Залежність кількості помилок від спроб для першого (1) та другого (3) користувачів та лінії тренду (2,4)

Таким чином, для оцінки якості інтерфейсу, очевидно, потрібен змішаний критерій, який буде враховувати кількість зроблених помилок та час досягнення кінцевого результату з деякими ваговими коефіцієнтами.

На рис. 6 показана залежність середнього часу виконання завдання від середнього розміру вікон, також відзначені середні розміри вікон до того, як прийшли користувачі, результати яких показані на попередніх графіках. Аналізуючи цей графік можна зробити висновок, що існує деякий оптимальний інтерфейс, при наближенні до якого кількість помилок та час виконання завдання значно знижуються.

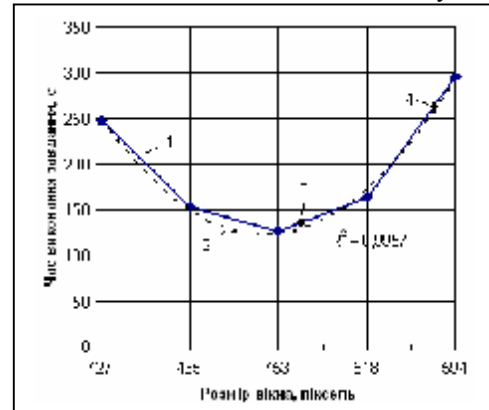


Рисунок 6 – Залежність середнього часу виконання завдання від середнього розміру вікон (1), 2 – степенева лінія тренду, 3,4 – показники першого та другого користувачів, R – коефіцієнт достовірності апроксимації

**Висновки.** В задачі налаштування оптимального інтерфейсу в смислі розмірів та положень окремих вікон дії користувача при налаштуванні інтерфейсу за деяким окремим параметром є безперервним, нестационарним кусково-гладким випадковим процесом, причому довжина гладких ділянок теж змінюється випадковим чином. Є підстави вважати, що в указаній задачі існує оптимальний інтерфейс. Індивідуальні особливості користувачів мають значний вплив на результати виконання завдання, але динаміка навчання всіх операторів досить подібна, що дає надію на можливість створення загальної моделі користувача в задачі налаштування оптимального інтерфейсу. Для оцінки якості інтерфейсу, очевидно, потрібен змішаний критерій, який буде враховувати кількість зроблених помилок та час досягнення кінцевого результату з деякими ваговими коефіцієнтами.

Для налаштування інтерфейсу в реальному часі перспективним видається шлях створення інтелектуальних інтерфейсних агентів [3].

#### ЛІТЕРАТУРА

1. Рассел С., Норвиг П. Искусственный интеллект: современный подход // Пер. с англ. – М.: Издательский дом «Вильямс», 2006.
2. Nielsen, Jakob, and Landauer, Thomas K.: A mathematical model of the finding of usability problems // Proceedings of ACM INTERCHI'93 Conference (Amsterdam, The Netherlands, 24-29 April 1993).
3. Растиргин Л. А. Адаптация сложных систем. Методы и приложения. – Рига: Зинатне, 1981.

Стаття надійшла 1.10.2007

Рекомендовано до друку к. фіз.-мат. н., доц. Ляшенко В. П.