

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ МІЖНАРОДНИХ ВІДНОСИН
КАФЕДРА КОМП'ЮТЕРНИХ МУЛЬТИМЕДІЙНИХ ТЕХНОЛОГІЙ

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач випускової кафедри
_____ С. М. Лобода
« _____ » _____ 2022 р.

КВАЛІФІКАЦІЙНА РОБОТА

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВР
ЗА СПЕЦІАЛЬНІСТЮ 186 «ВИДАВНИЦТВО ТА ПОЛІГРАФІЯ»

Тема: «Інформаційна модель проєктування інтерфейсу користувача»

Виконавець _____ студентка групи ВП-415 Головня Ксенія Юріївна
(студент, група, прізвище, ім'я, по батькові)

Керівник: _____ доцент, кандидат технічних наук Мартинова Оксана Петрівна
(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

Нормоконтролер:

_____ (підпис)

_____ С.М. Гальченко
(ПІБ)

Київ 2022

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет міжнародних відносин

Кафедра комп'ютерних мультимедійних технологій

Спеціальність 186 «Видавництво та поліграфія»

ЗАТВЕРДЖУЮ
Завідувач кафедри КММТ
С. М. Лобода
« _____ » _____ 2022 р.

ЗАВДАННЯ

на виконання кваліфікаційної роботи

Головні Ксенії Юрїївни
(П.І.Б. випусника)

1. Тема кваліфікаційної роботи: «Інформаційна модель проектування інтерфейсу користувача»
затверджена наказом ректора від 19 квітня 2022 р. №402/ст.
2. Термін виконання кваліфікаційної роботи: з 16.05.2022 р. по 19.06.2022 р.
3. Вихідні дані: інформаційні джерела з основ створення інтерфейсів електронних ресурсів; модель розробити у формі алгоритму з орієнтацією на проектування інтерфейсних рішень, в т.ч. і в бізнесі, наприклад, в системах електронного управління відносинами з клієнтами.
4. Зміст пояснювальної записки: Основи проектування засобів інтерфейсів людино-машинної взаємодії. Особливості проектування інтерфейсів з урахуванням мультимедійного контенту електронних ресурсів. Дослідження взаємодії користувача з елементами інтерфейсу. Розробка інтерфейсів електронних ресурсів.
5. Перелік обов'язкового ілюстративного матеріалу: презентаційний матеріал.

6. Календарний план-графік

№ пор.	Завдання	Термін виконання	Відмітка про виконання
1.	Обґрунтування вибору теми кваліфікаційної роботи. Визначення об'єкту та предмету дослідження.	16.05.20 - 17.05.22	
2.	Пошук інформаційних джерел за темою кваліфікаційної роботи	18.05.20 - 21.05.22	
3.	Відбір інформації за уявленням належності до тематики кваліфікаційної роботи. Здійснення інтелектуального аналізу тексту та зображень для інтеграції у відповідні тематичні розділи.	22.05.22 - 24.05.22	
4.	Розгляд моделей побудови різних видів інтерфейсів електронних ресурсів.	25.05.22 - 26.05.22	
5.	Розробка структурної схеми алгоритму проектування інформаційної моделі інтерфейсу користувача	27.05.22 - 28.05.22	
6.	Підготовка презентаційного матеріалу	30.05.22 - 10.06.22	

Дата видачі завдання: «16» травня 2022 р.

Керівник кваліфікаційної роботи _____ Мартинова О.П.
(підпис керівника)

Завдання прийняла до виконання _____ Головня К.Ю.
(підпис випускника)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи на тему «Інформаційна модель проектування інтерфейсу користувача»: 7 сторінок, 7 рисунків, 4 таблиць, 18 використаних інформаційних джерел.

ЛЮДИНО-МАШИНА ВЗАЄМОДІЯ, ІНТЕРФЕЙС, КОРИСТУВАЧ, ВІЗУАЛЬНІ АТРИБУТИ, ПРОЕКТУВАННЯ, ВАЛІДАЦІЯ, eCRM.

Об'єкт дослідження: взаємодія користувача з електронними ресурсами.

Предмет дослідження: інформаційна модель проектування інтерфейсу користувача.

Метою дослідження: визначення оптимальних рекомендації при проектуванні користувацьких інтерфейсів електронних ресурсів, які реалізуються при розробці користувацького інтерфейсу електронного видання, а саме — електронної бібліотеки.

Методи дослідження: розробити модель у формі алгоритму з орієнтацією на проектування інтерфейсних рішень, в т.ч. і в бізнесі, наприклад, в системах електронного управління відносинами з клієнтами.

Матеріали кваліфікаційної роботи рекомендується використовувати в навчальному процесі та при проведенні студентських наукових досліджень.

ЗМІСТ

ВСТУП	9
РОЗДІЛ 1. ОСНОВИ ПРОЄКТУВАННЯ ЗАСОБІВ ІНТЕРФЕЙСІВ ЛЮДИНО-МАШИНОЇ ВЗАЄМОДІЇ.	12
1.1. Людино-машинна взаємодія	12
1.2. Методи дослідження зручності інтерфейсів електронних ресурсів.....	155
1.3 Принципи та методології розробки інтерфейсів.....	18
1.4. Вимоги до проєктування користувацького інтерфейсу.....	19
1.5. Основні принципи розробки користувацького інтерфейсу.....	22
Висновки до першого розділу.....	27
РОЗДІЛ 2. ОСОБЛИВОСТІ ПРОЄКТУВАННЯ ІНТЕРФЕЙСІВ З УРАХУВАННЯМ МУЛЬТИМЕДІЙНОГО КОНТЕНТУ ЕЛЕКТРОННИХ РЕСУРСІВ	29
2.1. Вступні зауваження.....	29
2.2. Основні принципи проєктування інтерфейсного рішення	30
2.3. Форма подання інформації.....	31
2.4. Дидактичні принципи проєктування інтерфейсу	32
Висновки до другого розділу.....	41
РОЗДІЛ 3. ДОСЛІДЖЕННЯ ВЗАЄМОДІЇ КОРИСТУВАЧА З ЕЛЕМЕНТАМИ ІНТЕРФЕЙСУ	42
3.1. Взаємодія користувача з ресурсом	42
3.2. Проєктування елементів управління.....	43
3.3. Меню і навігація.....	45
3.4. Перемикачі і прапорці	48
3.5. Списки.....	51
3.6. Вибір візуальних атрибутів відображення інформації.....	53
3.7. Візуалізація виконуваних операцій.....	59

Висновки до третього розділу.....	61
РОЗДІЛ 4. РОЗРОБКА ІНТЕРФЕЙСІВ ЕЛЕКТРОННИХ РЕСУРСІВ	63
4.1. Проектування загальної структури інтерфейсу.....	63
4.2. Конструювання окремих блоків.....	64
4.3. Перевірка схеми за сценарієм.....	65
4.4. Вибір прототипу.....	66
4.5. Тестування і модифікація прототипу.....	677
4.6. Розробка структурної схеми алгоритму проектування інформаційної моделі інтерфейсу користувача.....	68
Висновки до четвертого розділу.....	69
ВИСНОВКИ.....	70
СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ	71
Додаток А.....	73
Додаток Б.....	74
Додаток В.....	75

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

- ANSI - американський Національний інститут стандартів.
- CASE - програми автоматизації процесу аналізу і проектування
- GUI (англ. graphical user interface) — графічний інтерфейс користувача або графічний користувальницький інтерфейс — різновид користувальницького інтерфейсу, у якому елементи інтерфейсу (меню, кнопки, значки, списки і т. п.), представлені користувачеві на дисплеї, виконані у вигляді графічних зображень.
- HCI - (англ. human-computer interaction) — дюдино-комп'ютерна взаємодія.
- UCD - (англ. user-centered design) — розробка, орієнтована на користувача.
- UI - (англ. user interface) — інтерфейс користувача, він же користувальницький інтерфейс — різновид інтерфейсів, в якому одна сторона представлена людиною (користувачем), інша — машиною / пристроєм. Являє собою сукупність засобів і методів, за допомогою яких користувач взаємодіє з різними машинами, пристроями і апаратурою.
- Usability - (англ. usability дослівно «можливість використання», «здатність бути використаним», «корисність») — зручність використання — поняття в мікроергономіці, ергономічна характеристика ступеня зручності предмета для застосування користувачами при досягненні певних цілей в деякому контексті. Термін має зв'язок з поняттям «ергономічність», але на відміну від останнього менше асоціюється з технічною естетикою, із зовнішнім виглядом.

ЕОР	- електронні освітні ресурси.
Когнетика	- когнітивна психологія вивчає когнітивні, тобто пізнавальні процеси людської свідомості. Дослідження в цій області зазвичай пов'язані з питаннями пам'яті, уваги, почуттів, подання інформації, логічного мислення, уяви, здатності до прийняття рішень.
ЛКМ	- ліва клавіша миші.
eCRM	електронні системи управління відносинами з клієнтами.
GOMS	- Goals, Operators, Methods, and Selection Rules (цілі, оператори, методи і правила їх вибору).
Prototyping	- макетування.
CardSorting	- карткове сортування.
CheckBoxes	- прапорці.
Radio	- перемикачі.
Buttons	-
Валідація	- це дії, які відповідно до принципів належної виробничої практики, доводять, що певна методика, процес, обладнання, сировина, діяльність або система дійсно приводять до очікуваних результатів.

ВСТУП

Динамічний розвиток інформаційних технологій, спрямований на підвищення інтерактивності взаємодії людини з віртуальним середовищем. Розвиток інтернету диктує необхідність розробки зручних інтерфейсів електронних видань. При проектуванні інтерфейсів необхідно виходити зі специфіки сприйняття і взаємодії користувачів з електронним ресурсом.

Не менш важливо володіти розумінням дизайну, вмінням відокремлювати значущі питання в проєктній діяльності від незначущих, а головне — відповідати на них. В проєктуванні інтерфейсів не тільки важливо зробити правильно, але не менш важливо не зробити неправильно — і навіть важливо не зробити чого-небудь непотрібного.

Актуальність тематики кваліфікаційної роботи заключається в тому, що незручний інтерфейс електронного видання, а також будь-якої програми чи додатку викликає певні незадоволення користувачів. Користувачеві доводиться відволікатися від своїх завдань, щоб вирішити проблеми, пов'язані з роботою програми. Навпаки, зручна, добре спроектована програма чи електронне видання, а до - зручний інтерфейс, дозволяє користувачеві повністю зосередитися на виконанні відповідного завдання.

Продуманий інтерфейс викликає справжнє задоволення користувачів. Наприклад, продукція компанії Apple, має високоякісний рівень функціонування та зручні інтерфейси., що дає відповідні уподобання користувачів щодо використання на різних етапах фахової діяльності і не тільки.

Щоб інтерне-сервіси були по-справжньому зручними і максимально задовольняли запити користувачів, необхідна дуже серйозна робота по проектуванню інтерфейсу.

Існує добре розроблена теорія інтерфейсів і методики їх проектування. Для розробки зручних рішень, необхідно розуміти критерії ефективності інтерфейсу і

принципи проектування. Тоді створення ефективних електронних ресурсів та програмних додатків стає набагато простішим.

Інтерфейс має бути розробленим таєм чином, щоб користувач не замислювався про проблемність його використання. Все, що користувач робить автоматично, за звичкою, не напружує його. Він буде вважати інтерфейс зручним, якщо зможе повністю зосередитися на розв'язуваній задачі.

При роботі з добре спроектованим інтерфейсом у фокусі уваги користувача буде максимальна зосередженість на робочі процеси виконання завдання.

Якщо натискання на одну і ту ж кнопку приводить до різних результатів в залежності від встановленого режиму, то інтерфейс називається модальним. Очевидно, що модальним інтерфейсом можна користуватися несвідомо. Рано чи пізно забувається, який режим зараз включений, і звична дія призводить до несподіваного результату.

Якщо виконати деяку дію можна тільки одним способом, то такий інтерфейс називається монотонним. Якщо інтерфейс немонотонний, то звична робота переривається свідомим міркуванням, яким чином краще виконати деяку дію.

Варто використовувати стандартні кнопки замість модних анімованих перемикачів. Створювати таблиці і форми за єдиним зразком.

Проектувальник інтерфейсу в процесі розробки задається питаннями: як визначити степінь готовності розробки, який ступінь можна вважати достатнім, яким коефіцієнтом корисної дії може володіти робота з розробки і як цей показник покращити та багато інших.

Таким чином дизайн - це правильна постановка таких питань і пошук правильних відповідей на них. Щоб відповісти на ці питання, доведеться вибрати адекватний процес. А оскільки в кожному конкретному проекті питань небагато, важко буде упустити щось важливе.

Якщо розробник інтерфейсу регулярно ставить питання в ході розробки проекту і максимально фахово знаходить відповідь на них - можна осікувати хорошого результату щодо зручності використання інтерфейсу для користувача. Питання не

дадуть прийняти невдалих рішень і змусять діяти правильно.

Досягнення поставленої мети у кваліфікаційній роботі пов'язується з розв'язанням задач:

- доведення актуальності досліджень у галузі проектування інтерфейсів електронних видань;
- розгляд етапів проектування користувацьких інтерфейсів;
- розробка інтерфейсу електронного видання на прикладі створення інтерфейсу електронної бібліотеки;
- рекомендації щодо побудови якісного сучасного інтерфейсу електронного видання.

Для отримання максимальної інформованості, знань та умінь щодо тематики «Інтерфейси користувача» слід звернути увагу, в першу чергу, на ресурси [1, 2, 3, 4].

РОЗДІЛ 1

ОСНОВИ ПРОЄКТУВАННЯ ЗАСОБІВ ІНТЕРФЕЙСІВ ЛЮДИНО-МАШИННОЇ ВЗАЄМОДІЇ

1.1. Людино-машинна взаємодія

Раніше, як правило, використовувався досить обмежений арсенал засобів взаємодії користувачів зі створюваними ресурсами. Така взаємодія полягала в обміні текстовими повідомленнями або псевдографічними зображеннями. При цьому програмний ресурс в основному відповідав діючим у той час державним стандартам, які передбачали наявність відповідної програмної документації, за допомогою якої користувачі могли усвідомити всі особливості практичного застосування. У силу зазначених причин проблеми, пов'язані з розробкою та освоєнням засобів взаємодії користувачів з програмами, вирішувалися стандартним чином.

За останні роки ситуація докорінно змінилася. По-перше, з поширенням персональних комп'ютерів неймовірно зросло число користувачів, у тому числі тих, які не мають навіть початкових знань у галузі обчислювальної техніки.

По-друге, значно збільшилося і число програмуючих користувачів, які мають відповідну базову підготовку, так і «самоучок»; в їх розпорядженні є потужні засоби розробки, які дозволяють створювати програмні ресурси із практично необмеженими інтерактивними можливостями.

По-третє, наявні державні стандарти на розробку програмних продуктів дещо застаріли і їх вплив на технологію програмування практично не мотивує користувача.

І, нарешті, наявність такого унікального засобу комунікації, як Інтернет, дозволяє всім бажаючим завантажувати результати своєї творчості, незалежно від їх якості і призначення.

Все це призвело до того, що в інтернеті стало з'являтися все більше програмних продуктів недостатньої якості. Навіть корисні ресурси дещо мають невдалий

інтерфейс і залишаються незатребуваними.

Більшість з них забезпечені яскравим і, на перший погляд, вельми вражаючим інтерфейсом у вигляді численних і різноманітних кнопок, піктограм, перемикачів, які заважають побачити, що ж ці програмні продукти все-таки вміють робити. Внаслідок цього активно починають розвиватись дослідження у напрямку проектування, створення на покращення програмних продуктів, в тому числі і взаємодія користувача з ними.

Людино-комп'ютерна взаємодія (HCI, англ. Human-computer interaction) — полідисциплінарний науковий напрямок, існує і розвивається з метою вдосконалення методів розробки, оцінки та впровадження інтерактивних комп'ютерних систем, призначених для використання людиною, а також в цілях дослідження різних аспектів цього використання [4].

Людино-комп'ютерна взаємодія (HCI) — це вивчення, планування та розробка взаємодії між людьми (користувачами) та комп'ютерами. Найчастіше її розглядають як сукупність науки про комп'ютери. Взаємодія між користувачами і комп'ютерами відбувається на рівні інтерфейсу, який включає в себе програмне і апаратне забезпечення: наприклад, образи чи об'єкти, що відображаються на екранах моніторів. Дані, отримані від користувача за допомогою апаратних пристроїв (клавіатури і миші) надають можливість взаємодії користувача з великими автоматизованими системами, комп'ютеризованими видавничими системами тощо.

Фахівці розглядають взаємодію людини і комп'ютера як «дисципліну, що займається проектуванням, оцінкою та здійсненням роботи інтерактивних обчислювальних систем для використання людиною, а також вивченням процесів, що відбуваються». Важливим аспектом людино-комп'ютерної взаємодії є сервісне забезпечення запитів користувачів.

У зв'язку з тим, що людино-комп'ютерна взаємодія вивчається як з людської сторони, так і з комп'ютерної, то знання, отримані в ході дослідження, спираються як на людський фактор, так і на комп'ютерний. З комп'ютерного боку такі технології як комп'ютерна графіка, операційні системи, мови програмування і середовища

розробки. З людського боку, — теорія комунікації, графічне і виробниче проектування, лінгвістика, соціологія, когнітивна психологія і такі людські фактори як задоволення користувачів.

Цілі людино-машинної взаємодії. Основним завданням людино-комп'ютерної взаємодії є поліпшення взаємодії між людиною і комп'ютером, роблячи комп'ютери більш зручними і сприйнятливими до потреб користувачів. Зокрема, людино-комп'ютерна взаємодія займається [4] :

- методологією та розвитком проектування інтерфейсів: виходячи з вимог і класу користувачів, проектування найкращого інтерфейсу в заданих рамках, оптимізація під необхідні властивості, такі як здатність до навчання і ефективність використання;

- методами реалізації інтерфейсів (наприклад, програмні інструментарії, бібліотеки та раціональні алгоритми);

- методами для оцінки та порівняння таких інтерфейсів;

- розробкою нових інтерфейсів і методів взаємодії;

- розвитком описових і прогнозованих моделей;

- теорією взаємодії.

Фахівці людино-комп'ютерної взаємодії — це, як правило, розробники, що займаються практичним застосуванням методик розробки до реальних проблем. Їх робота часто обертається навколо розробки графічних і веб-інтерфейсів.

Дослідники людино-комп'ютерної взаємодії займаються розвитком нових методик проектування, проведенням експериментів з новими апаратними пристроями, створенням прототипів нових систем програмного забезпечення, вивченням нових парадигм для взаємодії і розвитком теорій і моделей взаємодії.

Створення якісного людино-комп'ютерного інтерфейсу, який можна назвати точкою зв'язку між людиною і комп'ютером, є кінцева мета вивчення людино-комп'ютерної взаємодії.

1.2. Методи дослідження зручності інтерфейсів електронних ресурсів

Як визначити, наскільки зручним є електронне видання для більшості користувачів? У кожного є свої переваги, і те, що подобається одному, далеко не завжди подобається іншому. Придатність до використання і зручність веб-сайтів (web-usability) вивчається за допомогою великої кількості спеціальних методів.

Карткове сортування (Card Sorting). *Карткове сортування* — це класифікаційний метод, при якому користувачі сортують різні елементи розроблювального електронного видання по декільком категоріям. Для проведення карткового сортування створюється список параметрів, які передбачається піддати класифікації, після чого кожен із зазначених параметрів випикується на окремій картці. Картки пред'являються користувачам, яких інструктують згрупувати найбільш логічним, на їхню думку, чином. Отриману в результаті карткового сортування інформацію використовують для організації призначеного для користувача інтерфейсу.

Контекстне дослідження (Contextual Inquiry). Контекстне дослідження — це метод структурованого інтерв'ю, яке відрізняється від звичайного, наприклад, журналістського інтерв'ю, тим, що воно завжди побудовано на трьох базових принципах:

- облік контексту, в якому використовується досліджуваний сайт;
- спільна оцінка ресурсу користувачем та розробником;
- у фокусі оцінки ресурсу знаходиться саме його зручність для користувача.

Контекстне дослідження є однією з альтернатив методу еталонного тестування, при якому зручність оцінюється в лабораторних умовах, а не в звичній для користувача робочій обстановці. При контекстному дослідженні робота, час, мотивація і соціальні фактори, що впливають на користувача, залишаються такими ж, як в реальному світі, на відміну від лабораторних досліджень, де ці фактори контролюються експериментатором. Контекстне дослідження найбільш застосовно для того, щоб оцінити ту обстановку, в якій буде використовуватися продукт, тому

воно проводиться на ранніх стадіях його розробки.

Контрольні листи (Checklists). Контрольні листи допомагають упевнитися в тому, що веб-сайт виконаний з урахуванням принципів функціональності дизайну. Зазвичай їх використовують на заключній стадії роботи на додаток до експертних методів для того, щоб структурувати експертні оцінки якихось певних ознак. Існує велика кількість готових контрольних листів, однак рішення про використання того чи іншого списку повинно залежати від завдань дослідження. Найчастіше виникає необхідність у розробці власних критеріїв якості в певній галузі.

Макетування (Prototyping). Макетування — це створення моделі кінцевого продукту (веб-сайту), що дозволяє протестувати його складові на будь-яких стадіях розробки. У процесі макетування будується модель, що включає всі тестовані компоненти (дизайн, елементи управління і т.д.). Можна використовувати різні способи її побудови, від зображення елементів інтерфейсу на папері до створення робочого макета веб-сайту. Розрізняють «горизонтальне» і «вертикальне» макетування. «Горизонтальне» макетування означає представлення в макеті широкого спектру параметрів веб-сайту, але без їх детального опрацювання. Цей вид макетування переважно використовувати для вивчення користувацьких переваг в області інтерфейсу. У процесі дослідження з'ясовують, де, на думку користувачів, повинні знаходитися ті чи інші функції, наскільки вони доступні і т.д. «Горизонтальне» макетування застосовують на ранніх стадіях розробки. При «вертикальному» макетуванні досліджують функціональність окремих компонентів.

Огляди (Surveys). Огляди — це спеціальні інтерв'ю з користувачами, при якому їм задаються спеціально підготовлені питання, а їх відповіді записуються для подальшої обробки. Традиційна методологія проведення оглядів може також відігравати важливу роль у дослідженні веб-сайтів. Питання, що входять в огляд, можуть змінюватись в залежності від цілей дослідження, але як правило групуються за такими категоріями: демографічна інформація про користувача (місце проживання, вік, рід занять і т.д.); оцінка інформаційного наповнення веб-сайту (яку інформацію очікували знайти, чи відповідає наповнення веб-сайту їхнім очікуванням, що хотіли

б додати і т.д.); оцінка дизайну сайту (легкість у використанні, швидкість завантаження, оформлення і т.д.). Огляди використовують як на стадіях концептуалізації та розробки для проведення маркетингових досліджень, ідентифікації потенційних користувачів, встановлення їх інформаційних потреб та компетентності у використанні комп'ютерів, так і після реалізації веб-сайту для оцінки реакцій користувачів на інформаційне наповнення та зручність.

Опитувальники (Questionnaires). Опитувальники для оцінки веб-сайтів складаються за тими ж принципами, що і психодіагностичні тести. Їх основна відмінність від оглядів полягає у формі подання користувачам. Якщо огляди — це структуровані інтерв'ю, то опитувальники пред'являються у вигляді списку запитань із варіантами відповіді. Існує досить багато стандартизованих опитувальників.

Плюралістичне опрацювання (Pluralistic Walkthroughs). Плюралістичне опрацювання проводиться великою за розміром групою, в яку крім експертів в області оцінки веб-продукції входять користувачі та розробники. Група крок за кроком розглядає сценарії поведінки користувача, обговорюючи і опрацьовуючи кожен елемент його взаємодії з веб-сайтом. Залучення в експеримент фахівців різних спеціальностей забезпечує оцінку веб-сайту з найрізноманітніших позицій, а їхні зауваження сприяють ідентифікації наявних недоліків. Метод застосовується на ранніх стадіях розробки в цілях отримання зворотного зв'язку як від фахівців, так і від кінцевих користувачів.

Експертиза компонентів (Feature Inspection). Експертиза компонентів призначена для аналізу конкретного набору ознак веб-сайту, з якими взаємодіє користувач для досягнення кінцевої мети. Наприклад, при роботі з пошуковою машиною користувачеві для досягнення результату необхідно набрати URL цієї системи, скласти запит, керуючись мовою запитів, які використовує система, або скористатися рубрикатором, якщо такий є і т.д. Метод передбачає оцінку доступності і функціональності кожного з цих кроків у контексті виконання завдання. Для цього визначають їх послідовність і відповідають на наступні питання: чи може користувач реалізувати конкретний крок без особливих складнощів; чи логічний перехід від

одного кроку до іншого; чи легко визначити, до якого кроку потрібно переходити на кожному етапі виконання завдання; чи добре позначені і оформлені ті чи інші функції і т.д. Експертиза компонентів застосовується в середині розробки РЕСУРСУ, коли набір функцій і послідовність їх застосування вже визначені [10].

1.3. Принципи та методології розробки інтерфейсів

При оцінці поточного користувацького інтерфейсу або розробки нового слід мати на увазі наступні принципи розробки:

Із самого початку необхідно акцентувати свою увагу на користувачах і завданнях: встановити кількість користувачів, необхідних для виконання завдання і визначити підходящих користувачів: ті, хто ніколи не використовував інтерфейс, або ті, хто ніколи не буде його використовувати в майбутньому є невідповідним користувачем. Крім того, необхідно визначити, які завдання і як часто будуть виконувати користувачі.

Емпіричні вимірювання. На ранній стадії провести тест інтерфейсу з реальними користувачами, які використовують інтерфейс кожен день. Слід мати на увазі, що результати можуть змінитися, якщо рівень продуктивності користувача не є точним відображенням реальної людино-комп'ютерної взаємодії. Встановити кількісні особливості практичності, такі як: кількість користувачів, що виконують завдання, час виконання завдання, і кількість помилок, зроблених в ході виконання завдання.

Ітеративне проєктування. Після визначення кількості користувачів, поставлених завдань, емпіричних вимірювань, виконуються наступні кроки ітеративної розробки:

- розробка користувацького інтерфейсу;
- проводяться тестування;
- аналізуються результати;
- повторюються попередні кроки.

Повторюється ітеративна розробка до тих пір, поки не буде створений

практичний, зручний для користувача інтерфейс.

Різноманітні методики, які викладають техніки проектування людино-комп'ютерної взаємодії, почали з'являтися в часи розвитку даної галузі в 1980-х роках. Більшість методик розробки походять від моделі взаємодії користувачів, розробників та технічних систем. Ранні методики, наприклад, трактували когнітивні процеси користувачів як передбачувані і такі, що піддаються кількісному визначенню, та пропонували розробникам при проектуванні користувальницьких інтерфейсів розглядати результати когнітивних досліджень в таких областях як пам'ять і увага. Сучасні моделі мають тенденцію акцентувати увагу на постійному зворотному зв'язку та діалозі між користувачами, розробниками та інженерами, і докладати максимум зусиль для того, щоб технічні системи крутилися навколо бажань користувачів, а не навпаки, бажання користувачів навколо вже готової системи [5].

Орієнтоване на користувача проектування: розробка, орієнтована на користувача в даний момент є сучасною, широко практикується філософією, суть якої полягає в тому, що користувачі повинні займати центральне місце в розробці будь-якої комп'ютерної системи. Користувачі, розробники і технічні фахівці працюють разом для того, щоб чітко висловити бажання, потреби і кордони, і створити систему, що відповідає цим вимогам. Орієнтовані на користувача проекти часто користуються дослідженнями етнографічного середовища, в якому користувачі будуть працювати з системою.

Принципи розробки користувальницького інтерфейсу (англ. Principles of user interface design): ці сім принципів можуть розглядатися в будь-який час, в будь-якому порядку протягом усього часу розробки, це: звичність, простота, очевидність, допустимість, послідовність, структура і зворотний зв'язок [6].

1.4. Вимоги до проектування користувачького інтерфейсу

У цьому розділі йдеться про поширене нерозуміння сутності таких систем, як інтерфейси, а також методів їх розробки.

Інтерфейс — це щось більше, ніж вікна, піктограми, випадаючі меню і миша. Необхідність проектування інтерфейсу вже на ранніх стадіях розробки продукту іноді упускається з виду. Інший фактор, який часто недооцінюється, полягає в тому, що всі ми наділені пізнавальними апаратами, які мають між собою багато спільного. При розробці інтерфейсів слід спершу врахувати загальні фактори, а потім вже розглядати індивідуальні відмінності. Але, на жаль, існуючі на сьогодні засоби конструювання інтерфейсів не дозволяють підійти до завдання саме таким чином.

В дійсності незалежно від того, наскільки складним є завдання, що виконується тим чи іншим продуктом, складові частини цієї задачі все одно повинні залишатися простими. Ця глава закінчується визначенням людиноорієнтованого інтерфейсу.

Вираз *інтерфейс «людина-машина»* або *інтерфейс «людина-комп'ютер»* буде скорочувати до користувальницького інтерфейсу або просто інтерфейсу. Багато хто вважає, що термін користувальницький інтерфейс відноситься тільки до сучасних графічний користувальницький інтерфейс (GUI), що базується на вікнах і меню, керованих за допомогою миші.

Інтерфейс користувача — це сукупність інформаційної моделі проблемної області, засобів і способів взаємодії користувача з інформаційною моделлю, а також компонентів, що забезпечують формування інформаційної моделі в процесі роботи програмної системи.

Під інформаційною моделлю розуміється умовне уявлення проблемної області, сформоване за допомогою комп'ютерних (візуальних і звукових) об'єктів, що відображають склад і взаємодію реальних компонентів проблемної області.

Засоби та способи взаємодії з інформаційною моделлю визначаються складом апаратного та програмного забезпечення, наявного в розпорядженні користувача, і від характеру розв'язуваної задачі. Наприклад, для користувача, який хоче переписати файл з дискети на жорсткий диск, такими засобами є пристрої введення-виводу (клавіатура, миша і екран монітора) і два дисководи з дисками. А ось для користувача, який намагається встановити власні значення параметрів BIOS, перелік доступних засобів істотно ширший. Причому, більшість наявних обмежень, сприймаються і

розробниками і користувачами як «усвідомлена необхідність», оскільки практично всі вони носять об'єктивний характер.

Зовсім інша справа — та частина інтерфейсу, яка відноситься до програмних засобів. По-перше, для програми значно складніше сформулювати об'єктивні вимоги за складом і компоновкою органів управління; найчастіше не тільки користувач, але і самі розробники не можуть пояснити, чому програма має саме такі елементи керування. По-друге, їх перелік значно ширше, а склад змінюється у багато разів динамічніше, ніж склад апаратних засобів комп'ютера.

Досить поширеною є ситуація, коли програми, рівноцінні за призначенням і функціональним можливостям, виявляються зовсім різними за організацією взаємодії з користувачем. При цьому зовсім не обов'язково інтерфейс якоїсь із програм буде гіршим, він просто буде іншим. І якщо з якоїсь причини знайома програма виявиться недоступною, освоєння нової доведеться починати практично з нуля.

Значно більші втрати може понести користувач, якому належить або вибрати одну з незнайомих програм, або перейти на нову версію вже використовуваної програми. У першому випадку вибір може бути зроблений на користь менш функціональної і менш надійної програми, але такої, що володіє більш привабливим (із суб'єктивної точки зору) інтерфейсом.

У другому ж випадку незнайомий інтерфейс нової версії може виявитися психо-логічним бар'єром, не подолавши який користувач так і не зможе скористатися перевагами нової версії. Таким чином, ефективність роботи користувача визначається не тільки функціональними можливостями наявних у його розпорядженні апаратних і програмних засобів, але й доступністю для користувача цих можливостей. У свою чергу, повнота використання потенційних можливостей наявних ресурсів залежить від якості призначеного для користувача інтерфейсу.

Якщо треба надрукувати за допомогою комп'ютера запрошення на ювілей, доведеться скористатися текстовим або графічним редактором. Всі редактори «загального користування» дозволяють виконувати приблизно один і той же перелік операцій, але все питання в тому, як вони це роблять і яким уявляв собі розробник

потенційного користувача свого продукту. Редактор з невдалим інтерфейсом може вимагати від користувача знайомства з абсолютно новими для нього термінами, такими, наприклад, як «лігатура» і «кегель», а після кожного невдалого дії змусить розпочати роботу з самого початку. Робота з таким редактором може закінчитись тим, що ювіляр віддасть перевагу купити запрошення в магазині.

Головний висновок полягає в тому, що якість користувацького інтерфейсу є самостійною характеристикою програмного продукту, в порівнянні за значимістю з такими його показниками, як надійність та ефективність використання обчислювальних ресурсів. Важливий наслідок: розробник додатку повинен знати, що таке хороший інтерфейс, і як його побудувати.

1.5. Основні принципи розробки користувацького інтерфейсу

Створення якісного інтерфейсу вимагає значно більшого, ніж просто дотримання деяких інструкцій. Воно припускає реалізацію принципу «інтереси користувача понад усе» і відповідну методологію розробки всього програмного продукту. В англійській літературі для опису такого підходу використовується термін User-centered Design — UCD. Ця технологія передбачає проектування інтерфейсу з подальшим його розвитком в процесі розробки самого ресурсу.

Основна перевага хорошого інтерфейсу користувача полягає в тому, що користувач завжди відчуває, що він управляє програмним забезпеченням, а не програмне забезпечення управляє ним.

Для створення у користувача такого відчуття «внутрішньої свободи» інтерфейс повинен володіти цілим рядом властивостей, представлених на рис. 1.1.

Природний інтерфейс — такий, який не змушує користувача суттєво змінювати звичні для нього способи вирішення завдання. Це, зокрема, означає, що повідомлення і результати, що видаються додатком, не повинні вимагати додаткових пояснень. Доцільно також зберегти систему позначень і термінологію, використовувані в даній предметній області.



Рис. 1.1. Вимоги до користувацького інтерфейсу

Використання знайомих користувачеві понять і образів (метафор) забезпечує інтуїтивно зрозумілий інтерфейс при виконанні його завдань. Разом з тим не треба обмежувати їх машинну реалізацію повною аналогією з однойменними об'єктами реального світу. Наприклад, папка на Робочому столі Windows може використовуватися для зберігання цілого ряду інших об'єктів.

Метафори є свого роду «містком», що зв'язує образи реального світу з тими діями і об'єктами, якими доводиться маніпулювати користувачеві при його роботі на комп'ютері; вони забезпечують «впізнавання». Користувачі запам'ятовують дію, пов'язану зі знайомим об'єктом легше, ніж вони запам'ятали б ім'я команди, що пов'язана з цією дією.

Узгодженість дозволяє користувачам переносити наявні знання на нові завдання, освоювати нові аспекти швидше, і завдяки цьому фокусувати увагу на розв'язуваній задачі, а не витратити час на з'ясування відмінностей у виконанні тих чи інших елементів управління, команд і т.д. Узгодженість робить інтерфейс впізнаваним і передбачуваним. Узгодженість важлива для всіх аспектів інтерфейсу, включаючи імена команд, візуальне представлення інформації і поведінку інтерактивних елементів. Для реалізації властивості узгодженості в створюваному

програмному забезпеченні, необхідно враховувати його різні аспекти.

Користувачі зазвичай вивчають особливості роботи з новим програмним продуктом методом проб і помилок. Ефективний інтерфейс повинен брати до уваги такий підхід. На кожному етапі роботи він повинен дозволяти тільки відповідний набір дій і попереджати користувачів про ті ситуації, де вони можуть нашкодити системі або даним; ще краще, якщо у користувача існує можливість скасувати або виправити виконані дії. Навіть при наявності добре спроектованого інтерфейсу користувачі можуть робити ті чи інші помилки. Ці помилки можуть бути як «фізичного» типу (випадковий вибір неправильної команди або даних) так і «логічного» (прийняття неправильного рішення на вибір команди або даних). Ефективний інтерфейс повинен дозволяти запобігати ситуації, які, ймовірно закінчатся помилками. Він також повинен вміти адаптуватися до потенційних помилок користувача і полегшувати йому процес усунення наслідків таких помилок.

Інтерфейс завжди забезпечуйте зворотний зв'язок для дій користувача. Кожна дія користувача повинна отримувати візуальне, а іноді і звукове підтвердження того, що програмне забезпечення сприйняло введену команду; при цьому вид реакції, по можливості, повинен враховувати природу виконаної дії. Зворотній зв'язок ефективний в тому випадку, якщо він реалізується вчасно, тобто як можна ближче до точки останньої взаємодії користувача з системою. Коли комп'ютер обробляє завдання, що до нього надійшло, корисно надати користувачу інформацію щодо стану процесу, а також можливість перервати цей процес у разі потреби. Ніщо так не бентежить, як заблокований екран, який ніяк не реагує на його дії. Типовий користувач здатний витерпіти тільки кілька секунд очікування відповідної реакції від свого електронного «співрозмовника».

Інтерфейс повинен бути простим. При цьому мається на увазі не спрощення, а забезпечення легкості в його вивченні і у використанні. Крім того, він повинен надавати доступ до всього переліку функціональних можливостей, передбачених даним додатком. Реалізація доступу до широких функціональних можливостей і забезпечення простоти роботи суперечать один одному. Розробка ефективного

інтерфейсу покликана збалансувати ці цілі. Один з можливих шляхів підтримання простоти — поява на екрані інформації, мінімально необхідної для виконання користувачем чергового кроку завдання. Багатослівні командні імена або повідомлення, непродумані або надлишкові фрази ускладнюють користувачеві сприйняття суттєвої інформації.

Інший шлях до створення простого, але ефективного інтерфейсу — розміщення та представлення елементів на екрані з урахуванням їх смислового значення і логічного взаємозв'язку. Це дозволяє використовувати в процесі роботи асоціативне мислення користувача. Можна допомогти користувачам управляти складністю відображуваної інформації, використовуючи послідовне розкриття (діалогових вікон, розділів меню і т.д.). Послідовне розкриття передбачає таку організацію інформації, при якій в кожен момент часу на екрані знаходиться тільки та її частина, яка необхідна для виконання чергового кроку. Скорочуючи обсяг інформації, наданої користувачеві, зменшують обсяг інформації, що підлягає обробці. Прикладом такої організації є ієрархічне (каскадне) меню, кожен рівень якого відображає тільки ті пункти, які відповідають одному, обраному користувачем, пункту більш високого рівня в каскадній ієрархії.

Гнучкість інтерфейсу — це його здатність враховувати рівень підготовки і продуктивність праці користувача. Гнучкість припускає можливість зміни структури діалогу і/або вхідних даних. Концепція гнучкого (адаптивного) інтерфейсу в даний час є однією з основних областей дослідження взаємодії людини і ЕОР. Основна проблема полягає не в тому, як організувати зміни в діалозі, а в тому, які ознаки потрібно використовувати для визначення необхідності внесення змін і їхньої суті.

Проектування візуальних компонентів є найважливішою складовою частиною розробки програмного інтерфейсу. Коректне візуальне уявлення використовуваних об'єктів забезпечує передачу вельми важливої додаткової інформації про поведінку і взаємодію різних об'єктів. У той же час слід пам'ятати, що кожен візуальний елемент, який з'являється на екрані, потенційно вимагає уваги користувача, яка, як відомо, не безмежна. Слід формувати на екрані середовище, яке не тільки сприяло б розумінню

користувачем представленої інформації, але і дозволяло б зосередитися на найбільш важливих її аспектах. Найбільших успіхів у проектуванні користувальницького інтерфейсу, що володіє перерахованими властивостями, до теперішнього часу домоглися розробники комп'ютерних ігор.

Якість інтерфейсу складно оцінити кількісними характеристиками, однак більш-менш об'єктивну його оцінку можна отримати на основі наведених нижче показників:

1. Час, необхідний певному користувачеві для досягнення заданого рівня знань і навичок по роботі з додатком (наприклад, непрофесійний користувач повинен освоїти команди роботи з файлами не більше ніж за 4 години).

2. Збереження отриманих робочих навичок після закінчення деякого часу (наприклад, після тижневої перерви користувач повинен виконати визначену послідовність операцій за встановлений час).

3. Швидкість вирішення задачі за допомогою даного застосування; при цьому має оцінюватися не швидкодія системи і не швидкість введення даних з клавіатури, а час, необхідний для досягнення мети розв'язуваної задачі. Виходячи з цього, критерій оцінки по даному показнику може бути сформульований, наприклад, так: користувач повинен обробити за годину не менше 20 документів, безпомилково або з помилкою не більше 1%.

4. Суб'єктивна задоволеність користувача при роботі з системою (яка кількісно може бути виражена у відсотках або оцінкою за n-бальною шкалою).

Узагальнюючи викладене вище, можна коротко сформулювати ті основні правила, дотримання яких дозволяє розраховувати на створення ефективного користувацького інтерфейсу:

- інтерфейс користувача необхідно проектувати і розробляти як окремий компонент створюваного додатка;
- необхідно враховувати можливості і особливості апаратно-програмних засобів, на базі яких реалізується інтерфейс;
- доцільно враховувати особливості і традиції тієї предметної області, до якої

належить створюваний додаток;

- процес розробки інтерфейсу повинен носити ітераційний характер, його обов'язковим елементом має бути узгодження отриманих результатів з користувачем.
- засоби та методи реалізації інтерфейсу повинні забезпечувати можливість його адаптації до потреб користувача.

Висновки до першого розділу

Узагальнюючи наведену інформацію, інтерфейс користувача — це сукупність інформаційної моделі проблемної області, засобів і способів взаємодії користувача з інформаційною моделлю, а також компонентів, що забезпечують формування інформаційної моделі (умовне уявлення проблемної області, сформоване за допомогою комп'ютерних (візуальних і звукових) об'єктів) в процесі роботи програмної системи.

Основною перевагою хорошого інтерфейсу користувача є те, що користувач завжди має відчувати процес управління програмним забезпеченням, а не програмне забезпечення управляє ним.

Необхідні властивості, яким має відповідати добре спроектований інтерфейс, це: природність, узгодженість, дружність, зворотний зв'язок, простота, гнучкість, естетична привабливість.

Природність для інтерфейсу, котрий не змушує користувача суттєво змінювати звичні для нього способи вирішення завдання. Це, зокрема, означає, що повідомлення і результати, отримані в процесі зворотного зв'язку, не повинні вимагати додаткових пояснень.

Узгодженість дозволяє користувачам переносити наявні знання на нові завдання, освоювати нові аспекти швидше, і завдяки цьому фокусувати увагу на розв'язуваній задачі. Узгодженість робить інтерфейс впізнаваним і передбачуваним. Узгодженість важлива для всіх аспектів інтерфейсу. Для реалізації властивості узгодженості в створюваному програмному забезпеченні, необхідно враховувати

його різні аспекти.

Дружність. На кожному етапі роботи інтерфейс повинен дозволяти тільки відповідний набір дій і попереджати користувачів про ті ситуації, де вони можуть нашкодити системі або даним. Ефективний інтерфейс повинен дозволяти запобігати ситуації, які, ймовірно закінчатся помилками. Він також повинен вміти адаптуватися до потенційних помилок користувача і полегшувати йому процес усунення наслідків таких помилок.

Зворотний зв'язок. Завжди слід забезпечувати зворотний зв'язок для дій користувача. Кожна дія користувача повинна отримувати візуальне, а іноді і звукове підтвердження того, що програмне забезпечення сприйняло введену команду; при цьому вид реакції, по можливості, повинен враховувати природу виконаної дії. Зворотний зв'язок ефективний в тому випадку, якщо він реалізується вчасно. Коли комп'ютер обробляє завдання, що до нього надійшло, корисно надати користувачу інформацію щодо стану процесу, а також можливість перервати цей процес у разі потреби.

Простота. Інтерфейс повинен бути простим. При цьому мається на увазі не спрощення, а забезпечення легкості в його вивченні і у використанні. Крім того, він повинен надавати доступ до всього переліку функціональних можливостей, передбачених даним додатком.

Гнучкість інтерфейсу — це його здатність враховувати рівень підготовки і продуктивність праці користувача. Гнучкість припускає можливість зміни структури діалогу і/або вхідних даних. Концепція гнучкого (адаптивного) інтерфейсу в даний час є однією з основних областей дослідження взаємодії людини і ЕОМ.

Естетична привабливість. Коректне візуальне уявлення використовуваних об'єктів забезпечує передачу вельми важливої додаткової інформації про поведінку і взаємодію різних об'єктів.

РОЗДІЛ 2

ОСОБЛИВОСТІ ПРОЄКТУВАННЯ ІНТЕРФЕЙСІВ З УРАХУВАННЯМ МУЛЬТИМЕДІЙНОГО КОНТЕНТУ ЕЛЕКТРОННИХ РЕСУРСІВ

2.1. Вступні зауваження

Зміни, що відбуваються у світовому суспільстві, призвели до змін у всіх галузях нашої держави, у тому числі й освітньої. Основним шляхом модернізації й реформування освітньої галузі, відповідно до вимог глобального суспільства, виступає впровадження у навчальний процес інформаційно-комунікаційних, зокрема мультимедійних технологій, створення на їх основі навчально-інформаційного середовища, наповненого якісними електронними освітніми ресурсами (ЕОР), що покликані інтенсифікувати та підвищити ефективність професійної підготовки студентів у ВНЗ. Проте, використання сучасних ЕОР суттєво відрізняється від організації традиційного навчання і потребує специфічних підходів при їх розробці та реалізації, зокрема при поданні навчального матеріалу.

Існує позиція, що подання знань у сучасних ЕОР ідентичне підходам, реалізованим у традиційному навчанні. З цим не можливо не погодитися, адже електронна освіта базується на загальних основах теорії навчання, структуруванні знань, спрямована на розв'язання дидактичних цілей. Проте, і має ряд суттєвих рис, відмінних від традиційної. Насамперед, в ЕОР міститься особливий вид інформації – мультимедійна інформація, що являє собою інтеграцію різномірних даних, характеризується інтерактивністю та нелінійністю. Вона суттєво відрізняється від інформації представленої у традиційних навчальних ресурсах, а відтак спрацьовують інші психофізіологічні механізми при її сприйнятті, перетворенні і відтворенні.

Зважаючи на зазначені фактори, подання навчального контенту мультимедійними засобами в ЕОР є важливою проблемою, яка б забезпечила максимальну ефективність їх засвоєння особою, що навчається, її особистісного

розвитку та сприяла б здійсненню плідної педагогічної взаємодії в інформаційно-навчальному середовищі.

2.2. Основні принципи проєктування інтерфейсного рішення

На рис. 2.1 надана структурна схема основних принципів проєктування інтерфейсу користувача з урахуванням вимог та потреб користувачів.



Рис.2.1. Основні принципи проєктування інтерфейсу користувача

Природність інтерфейсу користувача (ІК) ЕОР – інтерфейс ЕОР не повинен вимагати від особи, що навчається, змінювати звичні способи розв’язування завдань, шляхом збереження системи термінів і позначень предметної області.

Узгодженість ІК ЕОР – елементи ІК мають бути узгоджені з наявними в особи, що навчається, знаннями по їх застосуванню (одні і ті ж команди, елементи управління, метафори повинні позначати ті самі функції, дії).

Простота ІК ЕОР – ІК має забезпечувати легкість його вивчення і використання. Шляхи реалізації – розміщення на екрані мінімально необхідної інформації, розташування елементів у відповідності з їх логічним значенням.

Гнучкість ІК ЕОР – ІК повинен враховувати рівень користувацьких умінь осіб.

Естетичність ІК ЕОР – полягає у візуально привабливому представленні об'єктів. Головна мета – створення на екрані середовища, яке б сприяло особі, що навчається, зрозуміти навчальну інформацію і зосередитися на основному.

Впізнаваність ІК ЕОР – ІК має містити лише стандартні базові елементи.

2.3. Форма подання інформації

Форма подання інформації відіграє суттєву роль при передачі знань, як в традиційних засобах навчання, так і при комп'ютеризованому навчанні. У другому випадку, значення інтерфейсу користувача важко перебільшити.

Від структури інтерфейсу залежить і функціонування ресурсу в цілому, і успіх навчальної діяльності, і психологічний стан осіб, що навчаються.

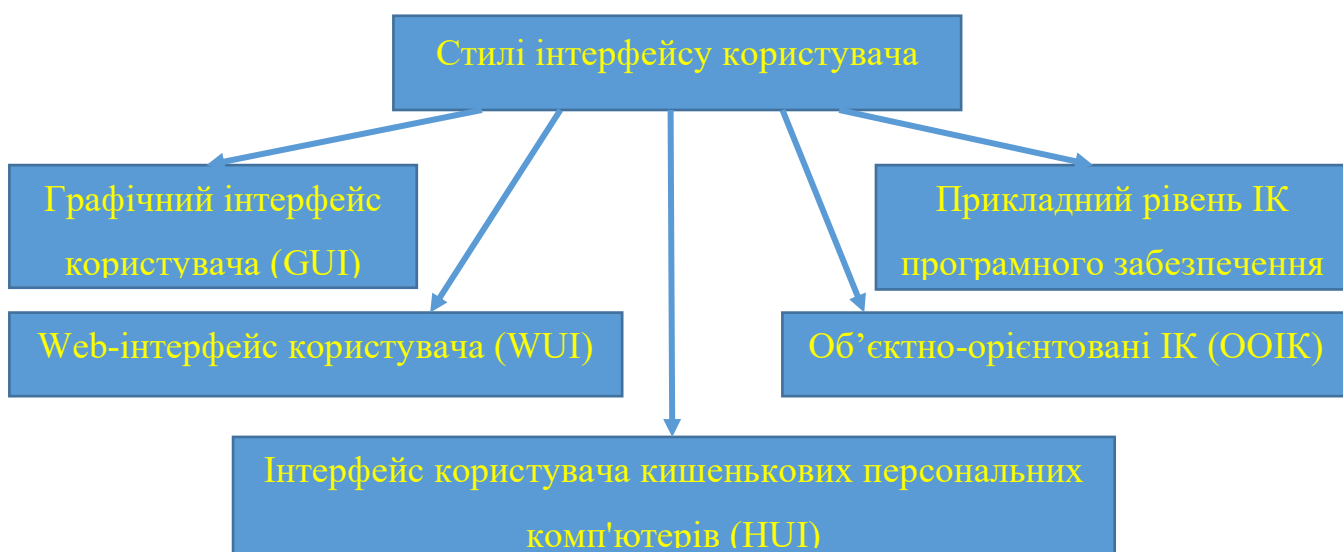


Рис. 2.2. Стили інтерфейсу користувача

2.4. Дидактичні принципи проєктування інтерфейсу

При поданні навчального матеріалу в ЕОР, потрібно створювати умови для максимальної реалізації дидактичних цілей навчання, це можливо, зокрема, за умов дотримання наведених принципів проєктування інтерфейсу користувача (табл. 2.1). Коректна організація інтерфейсу ЕОР повинна максимально сприяти комфортному та якісному сприйняттю поданого мультимедійного контенту, це можливо лише за умови проєктування мультимедійного контенту з урахуванням закономірностей сприймання особи, що навчається.

При поданні навчального матеріалу, значну увагу слід приділяти розташуванню елементів на площині екрану. При цьому варто виходити із закономірностей сприймання екранного зображення:

- *рух погляду* – горизонтально, зліва направо, зверху у низ.
- *найпершою сприймається* інформація у лівому верхньому куті.
- *найточніше сприймається* інформація, розташована у центрі екрану.
- *знижена ефективність сприймання* сигналу – у кутах екрану.

Таблиця 2.1

Принципів проєктування інтерфейсу користувача

Принцип	Реалізація
1	2
<i>Принцип змістовності</i> полягає в адекватному відтворенні наукових знань та представленні способів і алгоритмів дій, для забезпечення ефективності їх засвоєння особою, яка навчається	Проектування мультимедійного контенту має визначатися саме <i>змістом</i> знань
<i>Принцип відповідності цілям навчання</i> полягає у забезпеченні максимальної реалізації поставлених педагогічних цілей.	Ціль є основним визначником дій щодо подання мультимедійного контенту, оскільки спосіб і форма подання навчального матеріалу повинні бути підпорядковані цілі навчальної діяльності

1	2
<p><i>Мотиваційний принцип полягає у такому представленні знань, яке б спонукало до розвитку пізнавальної мотивації в осіб, які навчаються.</i></p>	<p>Новизна та різноманітність Різнманітна наочність Самостійність осіб, які навчаються Проблемність Індивідуальна орієнтованість ресурсу Естетична привабливість Емоційна забарвленість</p>
<p><i>Принцип емоційності полягає у створенні умов задля забезпечення виникнення позитивних емоцій в осіб, які навчаються.</i></p>	<p>Звернення до емоцій користувачів Вияви поваги до користувачів Чуттєве сприйняття Простота і зрозумілість Використання анімаційних персонажів Емоційно насичені зображення Застосування засобів акцентування</p>
<p><i>Принцип активізації навчальної діяльності полягає в активізації та орієнтуванні особи, на здійснення навчальної діяльності і вирішення проблемних ситуацій, у сприянні більш інтенсивному та усвідомленому сприйняттю навчального матеріалу.</i></p>	<p>Забезпечення активної роботи з поданим навчальним контентом. Наявність елементів проблемності. Застосування методів візуалізації. Забезпечення самостійності осіб, які навчаються.</p>
<p><i>Принцип інтерактивності розкриває характер і ступінь взаємодії між складовими навчально-інформаційного середовища</i></p>	<p>Засоби навігації Засоби зворотного зв'язку Засоби впливу на контент Самостійність осіб, які навчаються</p>
<p><i>Принцип адаптованості полягає у такій організації інтерфейсу ЕОР, щоб максимально відповідати особливостям і потребам осіб, які навчаються, шляхом відповідного подання навчального матеріалу.</i></p>	<p>Дозованість навчального матеріалу Покроковість вивчення Контроль рівня засвоєння знань Засоби впливу на контент Механізми налаштування елементів контенту Засоби регулювання контентом</p>
<p><i>Принцип естетичності виражає необхідність створення благоприємних зовнішніх умов реалізації навчальної діяльності.</i></p>	<p>Проектування мультимедійного контенту має сприяти гарному настрою осіб, які навчаються, знижувати втомлюваність та підвищувати працездатність.</p>

Ефективність сприймання залежить від розташування на частинах екрану:

- гірше сприймаються цифри у лівій верхній, та фігури у лівій нижній четвертях екрану;

- цифри по діагоналі від правого нижнього кута екрану до верхнього лівого, та фігури від лівого нижнього кута до правого верхнього;

- при вертикальному розташуванні краще сприймаються елементи, що знаходяться у середині верхньої частини екрану, а при горизонтальному – по середині лівої сторони;

Зменшення ефективності сприймання – у нижній частини екрану.

Колір неодмінна складова сучасних засобів навчання, що чинить безпосередню дію на відчуття, психічні стани людини, створює комфортні умови реалізації навчальної діяльності (або навпаки).

При виборі колірної палітри ЕОР, потрібно зважати на наступне: основним орієнтиром мають слугувати дидактичні цілі, індивідуальні характеристики осіб, що навчаються, та властивості самих кольорів.

У табл. 2. наведено характеристики кольорів та їх вплив на організм людини.

Залежність змісту кольору й інформаційних кадрів:

- хроматичні кольори варто використовувати для емоційно-оцінювальних компонент інформації;

- ахроматичні – для когнітивних.

Залежність кольору і розміру екранного простору – для великих екранних вікон доцільніше використовувати ахроматичні кольори, для малих площин – хроматичні.

Також важливо при оформленні інтерфейсу застосовувати обмежену кількість кольорів (3-4 для фону, основного змісту, виділень та акцентування) і використовувати *єдину колірну гамму*, що викликає відчуття зв'язності, послідовності, комфортності.

Елементи зображення аналізуються у такій послідовності: горизонтальна лінія, вертикальна лінія, величина об'єкту, симетрія.

Характеристики кольорів та їх вплив на організм людини

Характеристика кольору	Вплив на організм людини
Білий Холодний, чистий	Слугує гарним фоном для інших кольорів, посилюючи їх інтенсивність
Сірий Нейтральний	Не викликає ніяких реакцій – не збуджує і не заспокоює. Викликає відчуття стабільності.
Червоний Теплий, збуджуючий, активний, енергійний	Активізує усі функції організму, стимулює мозок, покращує настрій. На початку роботи швидко підвищує працездатність, але швидко втомлює і вона спадає. Спрямовує увагу.
Помаранчевий Тонізуючий, стимулюючий, життєрадісний	Стимулює почуття, загострює сприйняття. Діє як і червоний, але менш інтенсивніше
Жовтий Теплий, тонізуючий, найяскравіший	Фізіологічно оптимальний, найменш втомлює, стимулює зір, нервову діяльність. Налаштовує до активної діяльності. При надмірності може втомлювати.
Зелений Свіжий, спокійний, врівноважений	Діє протилежно червоному: врівноважує, заспокоює нервову систему, сприяє концентрації уваги, підвищує гостроту зору. Незначно, але надовго підвищується працездатність.
Блакитний Холодний, спокійний	Впливає на сприйняття часу: продовжує, а простір розширює удаль. Заспокоює, знижує тривожність, розслаблює. Надмірність може перевтомлювати, викликати відчуття холоду і суму.
Синій Холодний, спокійний, стриманий	Заспокоює, сприяє зосередженню, зменшує тривожність. Впливає на сприйняття часу. При тривалому впливі здійснює гальмівну дію, аж до пригнічення, печалі, втоми.
Фіолетовий Холодний, похмурий	Врівноважує збудження і спокій. Пригнічує раціональне мислення, пробуджує інтуїцію. Підвищує працездатність. Надмірний вплив викликає пригнічення аж до депресії.
Чорний	Чинить сильні пригнічуючі і гальмівні дії. Слугує гарним фоном для інших кольорів, посилюючи їх властивості.

Послідовність розпізнання фігур: трикутник, обернений трикутник, квадрат, коло.

Текст в ЕОР відіграє кілька ролей. Це і візуальне представлення мови, і повноцінний графічний елемент та основна форма подачі навчального матеріалу.

Зовнішній вигляд тексту – це параметр, що значним чином впливає на зручність читання, сприйняття та привабливість. Сприйняття тексту залежить від контексту, характеристик особи, що навчається, та особливостей шрифту.

При поданні текстi, потрібно звернути увагу на вибiр шрифтового оформлення. Усi символи шрифту мають певнi загальнi вiзуальнi характеристики, добре вiзуально поєднуються, формуючи гармонiйну сторiнку, що зручно читається. Вибiр стилю шрифту супроводжується передачею вiдповiдного значення. При виборi шрифту, варто зважати на наступне:

- значним фактором для зручностi читання є iнтервал мiж буквами – не менше 30% вiд висоти знака;

- на зручнiсть читання впливає висота знаку, що має бути не менше 3 мм, та вiдношення ширини знаку до висоти – у межах 0.76–0.80;

- найкраще сприймаються рядковi лiтери, прописнi – значно важче;

- екрану легше сприймаються шрифти iз засiчками, проте для великих фрагментiв тексту доречнiшим є використання шрифтив без засiчок;

- для максимального компактного розмiщення тексту (для помiток, у вузьких колонках) можна застосовувати вузькi шрифти, а подовженi зручнi для заголовкiв та iнших вiдокремлених елементiв тексту;

- жирнi шрифти важкi для сприйняття, нав'язливi, iх доречно застосовувати у заголовках та iнших подiбних елементах, за рахунок iх помiтностi;

- шрифти з нахилом варто використовувати обмежено, оскiльки вони важче сприймаються та читається. Їх можна застосовувати для видiлень у певному фрагментi тексту, для бiльшої актуалiзацiї доцiльно видiлити жирним;

- для видiлення варто застосовувати i пiдкреслення тексту, в основному у заголовках та посиланнях;

- для подачi тексту варто застосовувати шрифти однiєї гарнiтури, або достатньо вiдмiннi, щоб уникнути неприємних вiзуальних вiдчуттiв;

- декоративнi та заголовнi шрифти недоцiльно використовувати для тривалого читання, краще для заголовкiв та з метою привертання уваги.

При поданні тестового фрагменту, потрібно обирати оптимальну довжину речень та об'єм абзаців. Оптимальними для сприйняття є:

- довжина речення до 8 слів;
- довжина рядка, у межах 40-80 знаків;
- довжина абзаців: 50 – 70 слів, або 3 – 4 речення;
- ритм речень у тексті – чергування довгих і коротких речень.

Звук в ЕОР застосовується у вигляді мовного супроводу, музики та звуків.

Параметри сприймання мови:

- тривалість вимови звуків: для голосних – 0,35 с, для приголосних – 0,02–0,3 с;
- темп вимови: оптимально – 120 слів у хвилину;
- інтенсивність мовних звуків: перевищувати інтенсивність шумів на 6 дБ;
- організація мовного повідомлення: точніше сприймаються довгі слова, бо порівняно із односкладовими вони володіють більшою кількістю опізнавальних ознак; найкраще сприйняттю повідомлення сприяє поєднання слів у фрази;
- об'єм повідомлення в межах 11 слів;
- оптимальність інформативного повідомлення: чергувати високоінформативні і низькоінформативні частини
- голос диктора: лаконічний, доступний, якісне звучання;
- мовлення диктора: виразне, чітке, зрозуміле, без дефектів, оптимальної швидкості.

Параметри сприймання музичного супроводу:

- чинники, що впливають на сприймання музики: готовність сприймати музичну композицію, слухацькі очікування, ситуація сприйняття, музична характеристика суб'єкта;
- вплив музики на емоції: відчуття суму, печалі, апатії, радості;
- вплив музики на психічний стан: сприяння швидкій утомі, збудженню, зменшенню тривожності;
- вплив музики на психофізіологічні процеси: заспокоює, активізує та ін.

Осмыслити навчальний матеріал, значить зрозуміти його, зуміти відтворити та застосувати. Осмислення навчального матеріалу нерозривно пов'язане із сприйняттям, увагою, мисленням, індивідуальними особливостями людини.

Важлива роль в осмисленні знань при електронному навчанні відводиться поданні мультимедійного контенту. Щоб створити умови для максимально ефективного осмислення, варто при поданні навчального матеріалу зважати на наступні рекомендації:

- створити установку на сприймання матеріалу;
- матеріал переробити, зробивши структурованим і зв'язним;
- надати матеріалу доступності і ясності, відповідно до змісту досвіду осіб, що навчаються;
- включати різні види наочності;
- в ілюстраціях акцентувати увагу на важливих зв'язках і основних поняттях об'єкту вивчення, включаючи їх аналоги у текстове або мовне супроводження;
- використовувати схеми-опорні сигнали;
- застосовувати кольорове оформлення інтерфейсу;
- забезпечувати операційну сторону діяльності; використовувати виділення, застосовуючи колір, шрифт, піктограми, символи, для виділення головних думок, основних даних тощо.

Пам'ять – неодмінна складова процесу пізнання, адже, перш ніж перетворитися у стійкі знання суб'єкта, сприйнята та усвідомлена інформація має бути порівняна із попереднім досвідом та вбудована у наявні когнітивні зв'язки. Пам'ять тісно пов'язана із увагою, сприйняттям, мисленням, обумовлена індивідуально-типологічними особливостями індивіда. Це складний психологічний процес, який забезпечує запам'ятовування, збереження, впізнавання і відтворення слідів попереднього досвіду.

Для забезпечення високої ефективності запам'ятовування, при проектуванні мультимедійного контенту, потрібно зважати на наступне.

- важливо враховувати властивості пам'яті;

- надавати установки на запам'ятовування;
- забезпечувати активну взаємодію особи, що навчається, з ресурсом;
- використовувати принцип інтересу – усе цікаве запам'ятовується легко і без зусиль користувача;
- використовувати емоційні особливості навчального матеріалу;
- перше враження від матеріалу: чим воно більше, тим яскравіший образ формується; подавати інформацію по різних каналах;
- навчальний матеріал подавати організовано та чітко структуровано;
- доповнювати поданий матеріал елементами, що викликають відповідні асоціації (ілюструвати текст піктограмами, графікою, фото);
- застосувати при викладі мультимедійного контенту явище синестезії.

Створення екранного образу знань є важливим завданням при поданні навчального матеріалу в ЕОР, що полягає в обґрунтованому виборі та комбінації засобів мультимедіа і їх розташуванню на екрані. При поданні мультимедійного контенту, необхідно знати, за допомогою яких засобів мультимедіа найдоцільніше подавати матеріал, та як його організувати на екрані, щоб сприяти кращому засвоєнню знань.



Рис. 2.3. Параметри вибору, комбінування та розташування засобів мультимедіа

Вибір засобу мультимедіа залежить від характеристик засобу. У таблиці 2.3 наведено можливості вибору мультимедійних засобів від дидактичного потенціалу мультимедійних ресурсів. Вибір засобу мультимедіа залежить від дидактичних цілей навчального процесу. У табл. 4. наведено можливості вибору мультимедійних засобів від цілей, що ставиться на певному етапі навчання. Важливо зважати на потенційні переваги поєднання засобів мультимедіа. Наведемо окремі комбінації мультимедійних засобів: *текст + аудіо* краще ніж просто *текст*; *фото + аудіо* краще ніж просто *фото*; *аудіо + текст* краще ніж *зображення*; *аудіо + зображення* краще ніж просто *аудіо*; *анімація + зображення* краще ніж *анімація + текст*; *аудіо + фото* краще ніж *аудіо + текст*; *текст + зображення* краще ніж сам текст чи саме *зображення*.

Таблиця 2.3

Вибору мультимедійних засобів від дидактики мультимедійних ресурсів

Медіа	Дидактичні можливості
Текст	Подання процедурної інформації, презентація нової інформації, розвиток навичок розуміння, засвоєння абстрактної інформації, розвитку прийомів логічного мислення.
Графіка Рисунки Карти Графіки, діаграми Таблиці, схеми	Сприяє розумінню інформації, акцентує увагу на головному. Загальний вигляд об'єкту. Просторове розташування і просторові відношення. Подання цифрової інформації, узагальнення, резюмування інформації, показ відношень. Демонстрація концептуальних відношень, порівняння інформації, організація і структурування змісту матеріалу.
Фото	Реалістичне відображення об'єкту, деталізація елементів об'єкту. Для мотивування, зацікавлення, зосередження уваги, позитивних емоцій.
Відео	Відображення конкретних прикладів, демонстрації процедур, подій, руху, змін у часі. Для детального подання предмету вивчення. При початковому ознайомленні з предметом вивчення. З метою мотивування та зацікавленості.
Анімація	Для демонстрації динамічних процесів і об'єктів, що розвиваються у часі і просторі. Для пояснення механізмів, важких для сприйняття
3D-моделі	Демонстрація процесів і явищ, недоступних у реальності. Поєднання навчання з практикою.
Аудіо	Для супроводу відео-аудіо-ряду, пояснення, заохочення.

Залежність вибору засобу мультимедіа від дидактичних цілей навчального ресурсу.

Дидактичні цілі	Засоби мультимедіа
Засвоєння невеликої кількості інформації. Короткотривале запам'ятовування.	Аудіо краще ніж текст.
Довготривале запам'ятовування.	Текст краще ніж аудіо. Зображ+текст+звук сприяє підвищенню запам'ятовування на 35-40%. Медіа+інтерактив підвищує запам'ятовування на 75%. Текст+відео+фото краще ніж подання інформації лише в аудіоформаті.
Відтворення негайне.	Максимальне використання відео, зображень.
Відтворення віддалене.	Подання інформації в ілюстративній формі краще, ніж текст.
Для покращення сприймання.	Максимальне використання графіки, зображень Відеоряд+аудіо збільшує ефективність сприймання на 50%.
Для підвищення засвоєння	Максимальне застосування зображень, графіки.
Увага. Емоції, інтерес. Мотивація.	Відео, фото, аудіо, анімація.

Висновки до другого розділу

Надані текстові та графічні обґрунтування за темою особливості проектування інтерфейсів з урахуванням мультимедійного контенту електронних ресурсів.

Розглянуті наступні питання:

- вступні зауваження;
- основні принципи проектування інтерфейсного рішення;
- форма подання інформації;
- дидактичні принципи проектування інтерфейсу.

РОЗДІЛ 3

ДОСЛІДЖЕННЯ ВЗАЄМОДІЇ КОРИСТУВАЧА З ЕЛЕМЕНТАМИ ІНТЕРФЕЙСУ

3.1. Взаємодія користувача з ресурсом

Основні операції взаємодії користувача з ресурсом, такі як навігація, вибір, перегляд, редагування, створення нових об'єктів базуються на парадигмі об'єктної обробки, в якій користувач ідентифікує об'єкт і дію, що відноситься до цього об'єкта. Послідовно реалізуючи цю техніку, дається можливість користувачеві переносити власні навички і знання на нові завдання.

Більшість створюваних додатків підтримують основні операції взаємодії для миші, клавіатури і пера. Доповнюючи або розширюючи основний набір операцій, потрібно враховувати можливість їх реалізації за допомогою вказаних пристроїв введення. Техніка використання не повинна бути єдиною для всіх пристроїв. Навпаки, вона повинна бути реалізована таким чином, щоб оптимізувати застосування конкретного пристрою з урахуванням його особливостей. Крім того, потрібно намагтися полегшити користувачеві перехід між пристроями, з тим, щоб він міг міняти пристрої в залежності від виконуваних дій.

Переміщуючи мишу, користувач може перемістити покажчик в будь-яку позицію на екрані. Навігація за допомогою пера подібна до навігації за допомогою миші, за виключенням того, що користувач переміщує перо, не торкаючись вхідної поверхні.

Клавіатурна навігація вимагає від користувача натискання спеціальних клавіш та їх комбінацій. Положення позиції (фокусу) введення визначається контекстом (поточною ситуацією); зокрема, при роботі з текстом воно ідентифікується становищем текстового курсора.

Основні клавіші навігації. Клавіші навігації - це чотири клавіші керування

курсором («Вправо», «Вліво», «Вгору», «Вниз»), а також клавіші «Home», «End», «Page Up», «Page Down» і клавіша «Tab». Натискання клавіш навігації в поєднанні з клавішею «Ctrl» дозволяє збільшувати крок переміщення. Наприклад, натискання клавіші «Вправо» переміщує курсор в текстовій області вправо на один символ, а натискання тієї ж клавіші спільно з «Ctrl» забезпечує переміщення курсора на одне слово.

Вибір. Вибір є основним засобом, за допомогою якого користувач ідентифікує об'єкти, які його цікавлять. Отже, реалізація моделі взаємодії, заснованої на використанні вибору — один з найбільш важливих аспектів проектування інтерфейсу. Вибір, як правило, припускає пряму вказівку користувачем ідентифікуємого об'єкта. Цей механізм відомий як явний вибір. Якщо об'єкт вибрано, користувач може визначити дію для нього.

Можливі також ситуації, коли ідентифікація (вибір) об'єкта проводиться «побічно», на основі деякого логічного правила або виходячи з поточного кон-тексту. Непрямий вибір працює найбільш ефективно в тих випадках, коли між об'єктом і дією існує проста й видима асоціація. Наприклад, коли користувач «протягує» смугу прокрутки, він одночасно визначає і вибір об'єкта «смуги прокрутки», і пов'язану з нею дію — «переміщення». Непрямий вибір може бути реалізований і за допомогою наявного зв'язку між об'єктами. Наприклад, вибираючи символ в текстовому документі, ви, тим самим, вибираєте параграф, в якому міститься даний символ.

3.2. Проектування елементів управління

У графічному інтерфейсі елемент управління — це засіб, за допомогою якого користувач взаємодіє з комп'ютерною програмою. Якість цієї взаємодії залежить від двох аспектів:

- відповідності елемента керування виконуваний задачі;
- від послідовності правил, за якими функціонує елемент управління [8].

Під елементами управління зазвичай розуміють компоненти графічного інтерфейсу, які надають користувачеві можливість змінювати зміст або форму представлення відображуваної інформації, а також керувати роботою додатка. До елементів управління належать, зокрема, списки, смуги прокручування, кнопки тощо. Кожен елемент управління має унікальний образ і забезпечує певну форму взаємодії користувача з додатком. Подібно більшості елементів інтерфейсу, елементи управління забезпечують зворотний зв'язок з користувачем, змінюючи певним чином свій зовнішній вигляд, коли вони активізовані. Наприклад, коли користувач взаємодіє з елементами управління, що використовують мишу, кожен з них «реагує» на вибір при натисканні кнопки миші, але не активізується до тих пір, поки користувач не відпустить кнопку. Якщо користувач переміщує покажчик за межі елемента управління при натиснутій кнопці миші, він перестає сприймати вхідний вплив. Якщо користувач повертається на елемент керування, той знову переходить в активний стан. Гаряча зона, яка визначає, чи буде реагувати елемент керування на покажчик, залежить від типу елемента.

Для деяких елементів управління, таких, наприклад, як кнопки, гаряча зона збігається з видимою межею елемента. Для інших гаряча зона може включати графічний символ елемента керування і текстову область, що до нього відноситься (наприклад, це справедливо для прапорців і перемикачів). Для більшості елементів керування система забезпечує виведення текстової підказки. Оскільки підказка допомагає користувачеві визначити призначення даного елемента управління, завжди потрібно ідентифікувати елемент, з котрим в даний момент взаємодіє користувач. Якщо елемент не має підказки, її можна реалізувати у вигляді статичної текстової області або у вигляді спливаючої підказки (tooltip).

Оскільки деякі з елементів управління можуть забезпечувати які-небудь специфічні способи взаємодії користувача з додатком, доцільно створити для них спливаюче меню. Воно може виявитися ефективним засобом у тих випадках, коли елемент управління використовується або для передачі деякої величини, що має кілька можливих значень, або для доступу до контекстно-залежної довідкової

інформації. Для таких меню повинні дотримуватися стандартні правила створення спливаючих меню, наведені нижче, за винятком того, що в даному випадку клацання лівою кнопкою миші (ЛКМ) на вибраному пункті не запускає дія, що зв'язується з елементом управління. Іншими словами, спливаюче меню елемента управління дозволяє користувачеві визначити, які дії він реалізує в поточній ситуації, але не дозволяє безпосередньо виконати ці дії.

Незважаючи на те, що меню за своїм призначенням має повне право входити в число елементів управління, практично у всіх виданнях, присвячених інструментальним засобам створення додатків, його розглядають як самостійний компонент інтерфейсу. Пояснюється це відмінністю програмної реалізації меню від інших елементів управління (які в англійській літературі називаються controls). Тим не менше на етапі проектування користувацького інтерфейсу багато питань, що відносяться до використання меню, повинні вирішуватися спільно з проектуванням інших елементів управління.

3.3. Меню і навігація

Загальний опис щодо вимог до елементів меню користувацького інтерфейсу надається в [8]:

- перша літера в назві пунктів меню — заголовна;
- всі пункти меню першого рівня активізують меню, що розкриваються;
- використовується не більше двох підрівнів меню;
- у випадючих меню не більше 10 елементів.

Навігація. У будь-якій системі, що використовує єдиний пристрій виводу для будь-яких завдань, якість навігації є важливою складовою: оскільки всі фізичні атрибути системи фактично відсутні, їх необхідно передавати за допомогою зображення на екрані. Наявність в системі розвиненої навігації значно полегшує процес користування системою, оскільки при візуальній навігаційній системі не потрібно пам'ятати контекст своїх дій (як я сюди потрапив? Що я хотів зробити?),

Оскільки цей контекст сам по собі показується на екрані. Розробка навігаційної системи є складна і велика робота, що вимагає великої кількості знань та навичок.

Цілі навігації. Будь-яка система навігації зобов'язана виконувати кілька функцій, а саме показувати користувачам:

- де знаходяться зараз;
- куди взагалі можна переміститися;
- з якої саме сторінки прийшли. Точно знати напрямок свого руху корисно,

оскільки це допомагає підтримувати контекст дій.

Дуже важливо зробити навігаційну систему, яка виконує максимум цих завдань, просто тому, що невиконання частини з них робить систему неповноцінною. Кожен раз, створюючи навігаційну систему, потрібно вкладати в неї максимальну кількість відповідей.

Меню. Меню містить перелік команд, наявних у розпорядженні користувача при виконанні певного кроку завдання або завдання в цілому. Меню надає користувачеві можливість вибору необхідного засобу вирішення задачі, не вимагаючи від нього запам'ятовування імен команд і їх синтаксису. Існують різні типи меню, в тому числі випадачі, спливаючі і каскадні (ієрархічні) меню. Застосування кожного з них має свої особливості, які розглядаються нижче.

Головне меню вікна і випадаюче меню. Одна з найбільш поширених форм меню-лінійна послідовність команд (або розділів). У зв'язку з цим головне меню також називають смугою меню. Смуга меню містить назви пунктів меню, кожен з яких надає доступ до випадаючого меню. Зміст головного меню і пов'язаних з ним випадаючого меню визначається функціональним призначенням застосування і контекстом виконуваних користувачем завдань. Можна забезпечити вибір користувачем додаткової конфігурації вікна, при якій головне меню не відображається. У цьому випадку необхідно передбачити елементи управління, що забезпечують доступ до тих же функцій додатка, що і головне меню. Коли користувач вибирає пункт головного меню, відображається пов'язане з ним випадаюче меню. Щоб вказати пункт меню, який потрібно вибрати за допомогою миші, користувач встановлює покажчик на його

назві і клацає ЛКМ. Ця дія призводить до виділення (підсвічування) даного пункту і відкриття пов'язаного з ним випадаючого меню.

Випадаюче меню відображається як панель з пунктами меню, розміщеними у вигляді стовпця. Хоча система підтримує можливість виводу пунктів меню, що випадає, в декілька стовпців, слід уникати такої форми подання, оскільки вона ускладнює роботу користувача. У міру того, як користувач переміщає покажчик по меню, кожен його пункт підсвічується, відображаючи поточне положення вказівника. Активуючи ЛКМ на пункті меню, користувач вибирає команду, пов'язану з цим пунктом; при цьому випадаюче меню закривається. Якщо користувач вирішив відмовитися від роботи з випадаючим меню, він може закрити його, повторно натиснувши ЛКМ на тому ж пункті головного меню. Аналогічний результат дає і клацання ЛКМ на іншому пункті головного меню (правда, в цьому випадку буде відкрито випадаюче меню, пов'язане з цим пунктом). Для роботи, з випадаючим меню за допомогою клавіатури використовується клавіша «Alt», яка активізує головне меню. Коли користувач натискає текстову клавішу, утримуючи натиснутою клавішу «Alt», система відображає випадаюче меню, відповідне натиснутій текстовій клавіші (як правило, це клавіша, зіставлена першому символу в назві пункту меню).

Майже будь-яка навігаційна система є меню. Це має безліч переваг, але також і деякі недоліки.

По-перше, меню обмежує можливість вибору тільки певним набором варіантів. Це добре, оскільки дозволяє виключити завідомо неправильні дії.

Але в навігаційних системах це виявляється недобре: замість вільного пересування по системі користувачам доводиться ходити по заздалегідь проробленим для них стежками, тобто ступінь свободи користувачів скорочується.

По-друге, будь-яке багаторівневе меню страждає від каскадних помилок. Способи боротьби з ними існують, але цих помилок краще б взагалі позбутися. Зробити ж це можна, тільки вилучивши або саме меню, або потребу його використовувати.

По-третє, більшість меню не здатне показати користувачам, куди їм розумно піти. Це не означає, що меню погане, але тільки те, що потрібно якось посилювати навігацію крім меню.

Єдиним універсальним (і працездатним) алгоритмом вирішення всіх цих проблем є створення системи з двох не пов'язаних між собою навігаційних систем. Перша (меню) показує користувачеві, де він знаходиться і куди він може перейти. Друга, присутня, можливо, не на кожній сторінці, показує користувачеві, куди він, найімовірніше, хоче перейти в даний момент. Вперше така система набула популярності на сайті Amazon.com (хоча ідея спочатку з'явилася на firefly.com, на жаль, цей сайт припинив існування). Система аналізує читацькі уподобання, простіше кажучи, реєструє, хто що купив і на які товари перед цим дивився. Зібравши ці дані, система пропонує їх користувачам, тобто дозволяє переглядати товари не по абстрактним групам, а по купівельним перевагам [5].

3.4. Перемикачі і прапорці

Прапорці (CheckBoxes) і перемикачі (Radio Buttons) використовуються для однієї мети: для вибору з групи запропонованих варіантів. Різниця між ними, полягає в тому, що прапорці використовуються для вибору одночасно кількох варіантів з групи, а перемикачі дозволяють зробити вибір на користь тільки одного варіанту [8].

Перемикачі (Radio Buttons), іноді звані також радіо-кнопками, надають користувачеві можливість вибрати єдиний варіант із запропонованої безлічі взаємовиключних альтернатив. Іншими словами, в будь-якій групі перемикачів може бути встановлений (обраний) тільки один. Тому якщо вікно містить перемикачі, що відносяться до різних об'єктів або до різних властивостей одного об'єкта, вони обов'язково повинні бути явним чином розділені на відповідні групи.

Перемикачі зображуються у вигляді невеликих кружків. Якщо кнопка вибрана, в середині круга з'являється точка (маркер). Не рекомендується використовувати перемикачі для запуску команд (тобто в якості кнопок управління). Єдиним винятком

є випадок, коли подвійне клацання ЛКМ на перемикачі використовується як засіб швидкого вибору значення деякої величини і виконання вбудованої команди вікна, яка є основною (або єдиною можливою) дією користувача для даного вікна.

Як правило, перемикачі використовуються для вибору одного з можливих значень деякої властивості об'єкта. При цьому бажано обмежити число різних варіантів (звичайно використовують не більше семи), але разом з тим їх повинно бути не менше двох. Якщо необхідно запропонувати користувачеві більшу кількість варіантів, бажано використати інші елементи управління, наприклад меню.

Кожен перемикач забезпечується текстовою міткою. Якщо ж для пояснення призначення кнопок у групі більш зручні графічні символи, краще використовувати кнопки управління. Стандартні перемикачі дозволяють створювати мітки, які складаються з кількох рядків. У таких випадках доцільно використовувати верхнє вирівнювання (якщо ситуація не вимагає якої-небудь іншої орієнтації тексту).

Як і для кнопок управління, для вибору перемикача використовується тиск на ЛКМ — або в кружку, або на мітці кнопки. Фокус введення переміщається на мітку кнопки, якщо покажчик знаходиться над нею або над кружком, коли користувач натискає кнопку миші. Якщо користувач переміщає покажчик за межі перемикача, не відпускаючи кнопку миші, він повертається в початковий стан. Вибір не буде зафіксовано до тих пір, поки користувач не відпустить кнопку миші (при цьому покажчик повинен знаходитися над перемикачем). Повторне клацання ЛКМ на тому ж перемикачі не змінює його стан; користувачеві потрібно явно вибрати альтернативний варіант в групі, щоб змінити або відновити колишній вибір.

Для роботи з перемикачами за допомогою клавіатури можуть бути визначені відповідні клавіші доступу. Вибір перемикача в групі може також виконуватися клавішею «Tab» і клавішами переміщення курсору.

Подібно перемикачам, прапорець може знаходитися в одному з двох станів: «включено» або «вимкнено». Ці стани звичайно іменуються «прапорець встановлений» або «прапорець знятий». Основна відмінність прапорців від перемикачів полягає в тому, що вони використовуються для відображення

незалежних варіантів вибору. Іншими словами, користувач може одночасно вибрати (встановити) кілька прапорців, що входять в одну групу.

Прапорець зображується у вигляді невеликого квадратного віконечка з супутньою міткою. Коли прапорець встановлений, у віконечку з'являється маркер.

Стандартний прапорець має текстову мітку. Якщо ж для пояснення призначення прапорця потрібен графічний символ, слід використовувати замість прапорця кнопку управління. При виборі тексту для мітки слід користуватися все тим же золотим правилом: чим коротше і змістовніше напис, тим краще; якщо текст не вдалося вмістити в один рядок, використовується верхнє вирівнювання рядків.

У деяких випадках прапорець може використовуватися для управління іншими елементами інтерфейсу. Зокрема, від стану прапорця може залежати доступність елементів більш низького рівня ієрархії.

Коли користувач клацає ЛКМ у віконечку прапорця чи на його мітці, стан прапорця перемикається на протилежний. При натисканні кнопки миші фокус введення переходить на прапорець. Як і при роботі з іншими елементами керування, якщо користувач, утримуючи кнопку миші, переміщає покажчик за межі прапорця, елемент повертається в початковий стан. Стан прапорця не зміниться, поки кнопка миші не буде відпущена. Для зміни стану покажчик повинен знаходитися в цей момент над прапорцем або його міткою.

Щоб надати користувачеві клавіатурний інтерфейс для управління прапорцями, потрібно визначити для кожної з них клавішу доступу. Крім того, для навігації між кнопками можуть підтримуватися клавіша «Tab» і клавіші управління курсором. Повторний вибір прапорця приводить до зміни його стану на протилежний (прапорець буде знятий). Якщо ж прапорець буде обраний втретє, то він знову повернеться у вихідне (проміжне) становище; те ж саме відбудеться і з редагованим фрагментом тексту.

3.5. Списки

Елемент керування Список (ListBox) один з найпопулярніших у всій палітрі компонентів для створення інтерфейсу. Він дозволяє легко переглядати великі обсяги інформації і здійснювати виділення потрібних рядків.

Список надає користувачеві зручний засіб управління вибором необхідних об'єктів або їх властивостей. Елементи списків можуть бути представлені як в текстовій, так і в графічній формі. Будь-який список як елемент інтерфейсу повинен забезпечувати не тільки візуальне відображення зробленого користувачем вибору, але й підтримку пов'язаних з вибраними пунктами (або пунктом) дій.

Використання списку є найбільш доцільним у тих випадках, коли число можливих варіантів вибору велике, або коли їх перелік може змінюватися. Порядок розташування елементів у списку визначається, як правило, його змістом, і повинен полегшувати перегляд списку користувачем. Наприклад, якщо список містить імена, то їх доцільно розташувати за алфавітом; список дат краще скласти в хронологічному порядку.

Стандартний список не забезпечується текстовою міткою. Тим не менш, можна створити для списку мітку у вигляді статичної текстової області, яка буде використовуватися в якості посилання для клавіатурного доступу до списку.

Ширина поля списку має бути достатньою, щоб розмістити в списку назву пункту середньої довжини. Якщо заздалегідь неможливо підібрати таку, використовується один з наступних підходів:

- поле списку робиться достатньо широким, щоб розмістити в ньому найдовший елемент;

- використовується еліпсис (...) в середині або в кінці довгих назв, щоб скоротити їх, зберігши при цьому ті символи, за якими їх слід розрізняти. Наприклад, при вказівці маршрутів доступу найбільш важливими є зазвичай початок і кінець маршруту, тому проміжні каталоги можуть бути замінені еліпсисом: пункт 1 \ пункт 2 \ ... \ пункт N;

- включається в панель списку горизонтальна смуга прокручування; правда, цей варіант має деякі обмеження з використання, оскільки наявність смуги прокрутки призводить до скорочення обсягу даних, які користувач може переглянути одночасно, а також псує зовнішній вигляд, тому майже не використовується;

- якщо кількість пунктів в списку перевищує видиму область, то в полі списку включається вертикальна смуга прокрутки. При цьому слід дотримуватися тих же правил використання смуги прокрутки, які були розглянуті раніше.

Списки можуть розрізнятися способом відображення вмісту і типом вибору, який вони підтримують. Нижче розглянуті особливості використання різних видів списків.

Випадаючий список. Подібно списку одиничного вибору, випадаючий список (drop-down list box) передбачає можливість вибору єдиного пункту; відмінність полягає в тому, що випадаючий список відображається на екрані тільки на вимогу користувача. Коли список закритий (точніше, згорнутий), в його вікні відображається тільки вибраний пункт. Щоб змінити вибір, користувачеві необхідно відкрити список. Хоча випадаючий список являє собою ефективний спосіб економного використання простору вікна, він разом з тим вимагає від користувача додаткових дій для перегляду і вибору пункту в порівнянні зі списком одиничного вибору.

Ширина вікна згорнутого списку, що випадає повинна бути на кілька пробілів більше середньої ширини пунктів в списку. У відкритому стані вікно списку має бути достатньо високим для того, щоб показати три-вісім пунктів. Разом з тим, вільний простір у вікні повинен дозволяти не тільки відобразити вибір в списку, а також виконувати користувачеві операції прямого маніпулювання. Взаємодія користувача з випадаючим списком подібна роботі з випадаючим меню. Наприклад, щоб відобразити список, користувач може клацнути ЛКМ на кнопці меню, а вибір пункту призводить до автоматичного закриття списку; вибір пункту, як і в меню, виконується активуванням ЛКМ у відповідному рядку.

Користувач може також працювати з випадаючим списком, використовуючи клавішу доступу і клавіші управління курсором. При натисканні клавіші доступу,

клавiші «Tab» або клавiші «Вгору», «Вниз», а також двох останніх клавiші у поєднанні з клавiшею «Alt», випадаючий список відкривається.

Загальні вимоги до списків:

- При відкритті в списках повинні стояти найбільш імовірні значення.
- Якщо список містить більше 50 елементів, використовується фільтр або режим пошуку.
- Немає часто використовуваних коротких списків (менше п'яти елементів) Такі списки представлені як групи перемикачів або прапорців.
- Елементи списку мають бути відсортовані: або структурно (за загальними ознаками), або за алфавітом.
- Якщо у списку більше 50 відсортованих за алфавітом елементів, першими трьома повинні бути найбільш використовувані елементи. Вони також повторюються на своїх алфавітних місцях.
- Багаторядкові списки мають висоти не менше 4 рядків.

3.6. Вибір візуальних атрибутів відображення інформації

Вдалиий (в сенсі продуманий) вибір візуальних атрибутів відображуваної на екрані інформації — це значно більше, ніж просто гарне зовнішнє оформлення додатка. Якість візуального проектування в значній мірі впливає і на психофізіологічний стан користувача, і на ефективність його роботи в цілому. Внаслідок цього навіть достатньо потужний за своїми можливостями програмний продукт може виявитися недостатньо функціональним, якщо візуальне подання його інтерфейсу не задовольняє відповідним вимогам. Саме тому при реалізації великих і відповідальних проектів програмісти останнім часом все частіше вдаються до послуг професійних дизайнерів. Зокрема, вже отримав практично офіційний статус такий термін, як Web-дизайн.

Зрозуміло, якість візуального проектування найбільшою мірою проявляється при використанні графічного інтерфейсу користувача. Для всіх GUI-додатків справедливі наступні положення:

- всі графічні елементи додатка створюють єдине візуальне середовище, тому обов'язковим етапом візуального проектування є вибір згаданої вище загальної концепції зовнішнього вигляду додатка;

- при використанні графічного інтерфейсу кожен графічний елемент і реалізована ним функція тісно взаємопов'язані, цей зв'язок має бути інтуїтивно зрозумілим користувачеві.

Виходячи з цього, в рамках візуального проектування (дизайну) інтерфейсу ресурси повинні бути вирішені такі проблеми, як ефективне використання простору екрану, вибір адекватної форми подання об'єктів, кольорова палітра і композиція графічних елементів, а також вибір засобів привернення уваги користувача до тих чи інших елементів інформації, яка відображається на екрані.

Дослідження в даній області (як, втім, і практичний досвід) свідчать про те, що людина значно краще сприймає візуальну інформацію, якщо вона відповідним чином організована в просторовому відношенні. Ми перегледаємо інформацію на екрані так само, як і будь-які інші форми інформації. Погляд завжди привертають в першу чергу кольорові елементи, а не чорно-білі, ізольовані (окремо розташовані), а не згруповані, і, нарешті, графічні, а не текстові.

Виходячи з цього, при проектуванні візуальних елементів інтерфейсу додатку доцільно спиратися на певні принципи, основними з яких є наступні:

- ієрархічна організація відображуваної інформації;
- візуальне виділення найбільш важливих елементів;
- збалансованість структури екрану;
- візуальне об'єднання логічно взаємозв'язаних елементів;
- забезпечення зручності читання і логічної узгодженості відображуваної інформації;
- використання єдиних підходів до візуалізації відображуваної інформації не

тільки в рамках додатка, але і робочого середовища в цілому (принцип інтеграції).

Ієрархічна організація інформації. Принцип ієрархічної організації інформації означає розміщення інформації з урахуванням її значення щодо інших візуальних елементів ресурсу. Результат цього впорядкування впливає на реалізацію всіх інших принципів візуального представлення інформації. З точки зору роботи користувача ієрархічна організація інформації визначає доступність тих чи інших даних і послідовність виконання розв'язуваної задачі. Для успішної реалізації цього принципу необхідно отримати відповіді на наступні питання:

- яка інформація найбільш важлива для користувача? Наприклад, найбільш важливими аспектами для користувача можуть бути створення або пошук документа;
- що користувач хоче або повинен робити в першу, у другу, в третю чергу?
- що користувач повинен бачити на екрані в першу, у другу, в третю чергу?

Візуальне виділення найбільш важливих елементів. При реалізації принципу залучення уваги користувача повинні вирішуватися два завдання: по-перше, вибір на кожному кроці роботи деякої основної ідеї, найбільш важливої для виконання цього кроку, по-друге, відповідне подання і розміщення елементів реалізують цю ідею. У силу зазначених вище психофізіологічних особливостей люди звертають погляд в першу чергу на верхній лівий кут області перегляду або на ту її частину, яка візуально відрізняється від інших. Виходячи з цього, має сенс розміщувати найбільш важливу інформацію (або вузловий елемент) або у верхньому лівому куті екрану, або у вікні, забезпеченому спеціальними атрибутами.

Монотонність виконання елементів інтерфейсу. Коли потрібно зробити вибір між кількома методами, фокус уваги зміщується з поточної задачі на прийняття рішення про вибір. Це є головним обґрунтуванням монотонності системи. Якщо умови для прийняття рішення залишаються досить простими і ясними, то в кожному разі можна поступити таким собі звичним способом, тим самим, роблячи ситуацію монотонною. Таким чином, перед розробником інтерфейсів стоїть завдання у пошуку монотонного рішення для того, щоб забезпечити такі переваги як легкість вивчення, простоту впровадження, мінімум документації і невеликий розмір витрат на

обслуговування. Монотонність полягає в тому, що для виклику однієї і тієї ж команди не повинно використовуватися безліч жестів. Інтерфейс, який не має режимів і є монотонним, надзвичайно корисним у використанні за умови, що всі інші характеристики мають, принаймні, нормальну якість, прийнятну для сучасних інтерфейсів. Добре розроблений, людиноорієнтований інтерфейс зовсім не вимагає розділення на підсистеми для новачків та експертів.

Збалансованість структури екрану. Принцип збалансованості структури екрану - один з найбільш важливих принципів візуального проектування. Він припускає, з одного боку, раціональне використання простору екрану, а з іншого — таке розміщення інформації, при якому на екрані в кожен момент часу представлена тільки та її частина, яка дійсно необхідна для виконання чергового кроку завдання користувача.

Візуальне об'єднання логічно взаємозв'язаних елементів. Даний принцип можна розглядати як розвиток попереднього. Візуальне об'єднання логічно взаємозв'язаних елементів сприяє з'ясуванню користувачем того, як саме представлена на екрані інформація та елементи управління пов'язані з виконуваним кроком завдання і один з одним. Наприклад, якщо в діалоговій панелі є кнопка, яка впливає на вміст списку, (наприклад, кнопка «читати повністю») доцільно помістити їх поряд того елементу списку, до якого вона відноситься.

Забезпечення зручності читання та логічна узгодженість відображуваної інформації. Мається на увазі, що будь-яка інформація (не тільки текстова) повинна бути виражена в компактній і разом з тим у доступній формі. Крім того, користувач повинен бути здатний усвідомити, як вона пов'язана з попередніми і наступними кроками завдання. При реалізації даного принципу корисно отримати відповіді на наступні питання:

- Чи можна представлену на екрані ідею або поняття висловити більш просто?
- Наскільки легко зможе користувач виконати даний крок?
- Чи дійсно всі відображувані елементи необхідні для виконання цього кроку?

Інтеграція. Якщо інтерфейс додатка візуально узгоджується з інтерфейсом системи і інтерфейсом інших програм, значно легше забезпечити користувачеві послідовне та передбачуване робоче середовище. Для того, щоб здійснити цей принцип, слід продумати наступне:

- наскільки виявляється узгодженим екранне представлення компонентів додатка в процесі роботи користувача?

- як співвідносяться візуальні параметри системного інтерфейсу і інтерфейсу інших програм?

Колір є одним з найважливіших візуальних атрибутів інтерфейсу. Оскільки колір має здатність залучати погляд, його слід використовувати для ідентифікації тих елементів інтерфейсу, на які потрібно звернути увагу користувача (наприклад, для виділення поточного вибору).

Колір також має асоціативний аспект, в багатьох випадках об'єкти одного кольору сприймаються людьми як взаємопов'язані. Колірна гама може також надавати певний емоційний або психологічний вплив; не даремно, наприклад, розділяють кольори на «холодні» і «теплі».

Області, фон яких представлений більш теплими відтінками червоної частини спектра, здаються більшими, ніж області, колір фону яких зміщений до блакитної частини. Об'єкти екрану на білому тлі або на тлі кольору із середньої частини спектра здаються яскравіше і легше сприймаються при різному зовнішньому освітленні. Найбільший контраст між двома областями екрану досягається в тому випадку, якщо колір фону однієї з них - чорний або близький до будь-якої з меж спектра, а колір фону протилежний - білий або взятий із середньої частини спектра.

Дані рекомендації справедливі і для співвідношення між кольорами символів і фону. Слід враховувати, що деякі комбінації кольору, наприклад, блакитний колір символів на червоному тлі, неприємні для очей. Різні психологічні стани людини характеризуються різними кольорами. Крім цього відзначена і зворотна залежність - вплив кольору на настрій і працездатність. У роботах, присвячених дослідженню психологічного впливу різних кольорів, наводяться такі дані: блакитний - заспокоює;

червоний - хвилює і стомлює; зелений - налаштовує на добродушний і безініціативний лад; жовтий - веселий, оптимістичний, викликає легковажний настрій; помаранчевий - розкріпачує фантазію; фіолетовий - згубний для очей, колір задрощів, тривоги, незадоволеності; коричневий - пригнічує розумову активність; чорний - похмурий, сприяє виникненню головного болю, але знижує кількість помилок.

Таким чином, непродумане використання кольору може викликати в користувача негативний емоційний стан або навіть відволікати його від роботи, заважаючи зосередитися на виконуваному завданні. Крім того, при виборі колірної палітри слід враховувати цілий ряд додаткових факторів: хоча можна використовувати колір, щоб відобразити зв'язок між тими чи іншими об'єктами, асоціація між вибраним кольором і конкретним значенням не завжди буде очевидна або легко впізнається користувачем. Колір є дуже суб'єктивною характеристикою.

Тому те, що подобається розробнику, зовсім не обов'язково буде приємно користувачам. Деякі користувачі можуть мати проблеми з кольоровим сприйняттям (близько 9 відсотків дорослого чоловічого населення мають те чи інше відхилення в сприйнятті кольору).

В силу наведених вище обставин колір повинен використовуватися тільки як додаткова форма передачі інформації. Ще один висновок полягає в тому, що доцільно в рамках одного ресурсу використовувати обмежену кількість кольорів, причому перевагу слід віддавати приглушеним, пастельним тонам. Оскільки колір є суб'єктивним фактором, слід дозволити користувачеві самому налаштовувати кольорову гаму, де це можливо. Для стандартних елементів інтерфейсу система забезпечує стандартні колірні схеми. Створюючи власний інтерфейс для зміни кольорів, потрібно враховувати складність виконуваного завдання і рівень підготовленості користувача.

Подібно іншим візуальним елементам, шрифти сприяють організації інформації та створенню певного настрою. Змінюючи розмір і щільність шрифту, можна вказати

користувачеві на ступінь важливості тієї чи іншої інформації і порядок, в якому вона повинна бути прочитана.

На екранах стандартних моніторів шрифти зазвичай менш розбірливі, ніж на друкованій сторінці. Бажано не використовувати курсив (*italic*) та рубаний шрифт (*serif*), оскільки вони важкі для читання, особливо при низькій роздільній здатності монітора. Варто обмежити кількість застосовуваних шрифтів і стилів. По можливості, використовувати менш екстравагантний шрифт для основного тексту. Для заголовків або елементів меню (якщо є така необхідність), а також для невеликого блоку інформації, щоб зробити акцент, можна використати акцендентні шрифти.

3.7. Візуалізація виконуваних операцій

Візуалізація виконуваних операцій є одним зі способів надання користувачу зворотного зв'язку з ресурсом. Продуманий спосіб візуалізації не тільки сприяє кращому з'ясуванню користувачем суті виконуваної операції, але і забезпечує своєчасну і коректну реакцію користувача у разі помилкових чи невдалих дій. Це особливо важливо для процесів в електронних виданнях, що відбуваються в режимі реального часу, наприклад, таких як здійснення завантаження файлу, процес пошуку даних тощо.

Візуалізація операцій вибору. Візуальний зворотний зв'язок при виконанні операцій вибору повинен дозволити користувачеві однозначно ідентифікувати вибраний об'єкт на фоні інших. Спосіб відображення вибору зазвичай залежить від сутності об'єкта і поточної ситуації.

Зображення об'єкта, що обирається повинно змінюватися безпосередньо в процесі виконання операції вибору. Те ж саме відноситься і до обраної області. При цьому стан вибору повинен відображатися тільки для активної області або рівня ієрархії (наприклад, для активного вікна). Це допоможе користувачеві визначити, до якого з наявних виборів відноситься виконувана дія. Тим не менш, в деяких ситуаціях може виявитися корисним одночасне візуальне виділення декількох об'єктів, що

відносяться до різних компонентів ресурсу. Наприклад, коли користувач вибирає об'єкт в первинному вікні і потім вибирає пункт меню, що відноситься до цього об'єкта, вибір повинен відображатися як для об'єкта, так і для пункту меню. У деяких випадках потрібно вказати, що один з існуючих виборів є «первинним», а інший — «вторинним». У таких ситуаціях доцільно використовувати колір: область навколо «вторинного» вибору може бути зафарбована кольором, відмінним від кольору виділення активного вибору.

У вторинних вікнах виділення поточного вибору кольором застосовується, як правило, тільки для текстових полів і елементів списків. Для вказівки вибору інших елементів управління доцільно використовувати візуальне відображення фокусу вводу.

Візуалізація операцій пересилки. При виконанні операцій пересилки засобом візуалізації є переміщення образу об'єкта, що пересилається або у вигляді невеличкої анімації, що ілюструє процес виконання дії. Анімація у багатьох випадках може виявитися вельми ефективним засобом передачі візуальної інформації (наприклад, в якості ілюстрації функціонування якого-небудь пристрою або виконання операції). Іноді її застосування може просто оживити інтерфейс програми та зробити спілкування користувача з ним більш приємним.

Ефективне використання анімації передбачає врахування багатьох факторів з тих, що були розглянуті вище стосовно розробки інших графічних елементів інтерфейсу. Одна з найважливіших вимог до використання анімації полягає в тому, що вона не повинна впливати на інтерактивність інтерфейсу. Якщо анімація не є частиною виконуваного процесу, слід дозволити користувачеві або перервати її, або паралельно продовжувати роботу. Слід уникати необгрунтованого використання анімації. Якщо анімація використовується тільки для створення декоративного ефекту, вона може відволікати або навіть дратувати користувача.

Висновки до третього розділу

У графічному інтерфейсі елемент управління — це засіб, за допомогою якого користувач взаємодіє з комп'ютерним ресурсом.

Якість цієї взаємодії залежить від двох аспектів: відповідності елемента керування виконуваній задачі і від послідовності правил, за якими функціонує елемент управління користувацького інтерфейсу.

Під елементами управління зазвичай розуміють компоненти графічного інтерфейсу, які надають користувачеві можливість змінювати вміст або форму представлення відображуваної інформації.

Наявність в системі розвиненої навігації значно полегшує процес навчання в роботі з системою, оскільки при візуальній навігаційній системі не потрібно пам'ятати контекст своїх дій.

Розробка навігаційної системи вимагає великої кількості знань і навичок. Будь-яка система навігації зобов'язана виконувати кілька функцій, а саме показувати користувачам: де знаходяться зараз, куди взагалі можна переміститися, де вже побували, куди розумно переміститись, з якої саме сторінки прийшли.

Меню містить перелік команд, наявних у розпорядженні користувача при виконанні певного кроку завдання або завдання в цілому.

Меню надає користувачеві можливість вибору необхідного засобу вирішення задачі, не вимагаючи від нього запам'ятовування імен команд і їх синтаксису.

Прапорці і перемикачі використовуються для однієї мети: для вибору дій з групи запропонованих варіантів.

Різниця між ними, полягає в тому, що прапорці використовуються для вибору одночасно кількох варіантів з групи, а перемикачі дозволяють зробити вибір на користь тільки одного варіанту.

Елемент керування «список» один з найпопулярніших у всій палітрі компонентів для створення інтерфейсу.

При проектуванні візуальних елементів інтерфейсу додатку доцільно спиратися на певні принципи, основними з яких є наступні:

- ієрархічна організація відображуваної інформації;
- візуальне виділення найбільш важливих елементів;
- збалансованість структури екрану;
- візуальне об'єднання логічно взаємозв'язаних елементів;
- зручність і узгодженість при читанні (перегляді) відображуваної інформації;
- використання єдиних підходів до візуалізації відображуваної інформації.

До візуальних атрибутів відображення інформації належать колір, шрифти, візуалізація виконуваних операції та процесів, маніпулювання екранним зображенням тощо.

Колір має асоціативний аспект. Колірна гама може також надавати певний емоційний або психологічний вплив.

Подібно іншим візуальним елементам, шрифти сприяють організації інформації та створенню певного комфорту.

РОЗДІЛ 4

РОЗРОБКА ІНТЕРФЕЙСУ ЕЛЕКТРОННОГО РЕСУРСУ

4.1. Проектування загальної структури інтерфейсу

Власне проектування складається з наступних етапів:

- визначення необхідної функціональності системи;
- створення користувальницьких сценаріїв;
- проектування поетапної структури;
- конструювання окремих блоків;
- збірка і початкова перевірка повної схеми системи.

Кожний наступний етап в такій системі залежить від результатів попередніх етапів. Відповідно, пропуск якогось етапу негативно впливає на результати всіх наступних етапів.

Наведену інформація використовуємо для створіння загальної структури системи, необхідно виділити окремі функціональні блоки і визначити, як саме ці блоки зв'язуються між собою.

Проектування загальної структури складається з двох процесів, які відбуваються паралельно: виділення незалежних блоків і визначення зв'язку між ними. Якщо проектується електронне видання, в завершенні необхідно також створити схему навігації.

Виділення незалежних блоків. Для цієї роботи важко дати які-небудь конкретні рекомендації, оскільки дуже багато залежить від системи, що проектується. Тим не менш, можна з упевненістю рекомендувати уникати приміщення в один блок більше трьох функцій, оскільки кожен блок у результуючій системі буде укладено в окрему сторінку або групу керуючих елементів. Перевантажувати інтерфейс небезпечно. Результатом цієї роботи має бути список блоків з необхідними поясненнями, наприклад, визначення смислового зв'язку між блоками. Існує три основних види

зв'язку між блоками. Це логічний зв'язок, зв'язок за поданням користувачів і процесуальний зв'язок. Логічний зв'язок визначає взаємодію між фрагментами системи з точки зору розробника. Користувачі мають свою думку про систему, і ця думка теж є важливим видом зв'язку. Нарешті, процесуальний зв'язок описує нехай не цілком логічну, але природну для наявного процесу взаємодію. Всі три типи взаємозв'язку повинні бути заздалегідь передбачені при конструюванні системи.

4.2. Конструювання окремих блоків

Отже, визначивши, скільки сторінок потрібно і що повинно відбуватися на кожній з них. Можна проєктувати окремі екрани. Це, мабуть, найскладніша частина роботи (не рахуючи використання запитів від користувачів). Є дві речі, які потрібно враховувати: це GOMS (Goals, Operators, Methods, and Selection Rules) — цілі, оператори, методи і правила їх вибору і адаптивна функціональність.

Передбачення швидкості. Часто доводиться вибирати між різними варіантами реалізації інтерфейсу, тому їх потрібно детально аналізувати з метою вибору найбільш коректного. Можна, звичайно, зробити кілька прототипів і протестувати їх на користувачах, але це досить тривалий і трудомісткий процес. Є метод оцінки інтерфейсу, що дозволяє швидко вибрати кращий варіант. Це метод оцінки швидкості роботи з системою, названий аббревіатурою GOMS. Ідея методу дуже проста: всі дії користувача можна розкласти на складові (наприклад, взяти мишу або пересунути курсор). Обмеживши номенклатуру цих складових, можна заміряти час їх виконання на масі користувачів, після чого отримати статистично вірні значення тривалості цих складових. Після чого прогнозування швидкості виконання якої-небудь задачі, або, вірніше, вибір найбільш ефективного вирішення, стає досить простою справою — потрібно тільки розкласти цю задачу на складові, після чого, знаючи тривалість кожної складової, все скласти і дізнатися тривалість всього процесу. Зазвичай той інтерфейс кращий, при якому час на виконання завдання менше.

Є розробки більш складних (але точних) варіантів цього методу.

Найпоширенішим, все одно, є початковий, так званий Keystroke-level Model (KLM). На жаль, цей варіант методу має певні недоліки (що, втім, врівноважується його простотою): застосовується в основному для пророкування дій досвідчених користувачів; ніяк не враховує ні прогресу в навчанні, ні можливих помилок, ні ступеня задоволення користувачів; погано застосовується при проектуванні електронних видань через непередбачуваний часу реакції системи.

4.3. Перевірка схеми за сценарієм

Останнім завданням перед побудовою прототипу є перевірка внутрішньої логіки системи. Справа в тому, що завжди існує ймовірність того, що щось забули або спланували неправильно. Як вже було сказано, виправити ці помилки краще всього до побудови прототипу (навіть першої його версії). Звичайно, багато структурних помилок не можна знайти ніякими методами, крім тривалого логічного аналізу. З іншого боку, практика показує, що майже всі знайдені помилки будуть істотними. Так що зайва перевірка не зашкодить. Для фінальної перевірки схеми знадобляться розроблені користувальницькі сценарії. Не дивлячись на схему, необхідно докладно описати, як користувачі будуть взаємодіяти з системою, не пропускаючи жодного елемента керування. Після чого звірити отриманий текст зі схемою.

Експертна оцінка. Досить ефективним засобом оцінки отриманого інтерфейсу є експертна оцінка. Хоча експертна оцінка не може бути повноцінною заміною тестуванню, вона володіє однією істотною перевагою — для її проведення не потрібен прототип. Це означає, що експерт може бути запрошений на ранніх стадіях роботи.

Для проведення експертної оцінки потрібно знати наступне: метод працює краще, коли кількість експертів більше одиниці; краще залучати кілька експертів не одночасно, але послідовно;

- чим більше інформації про систему буде надано експерту, тим ефективніше він надасть предмет вирішення проблем [6].

4.4. Вибір прототипу

При створенні прототипу найбільш частою помилкою є надмірне наведення глянцею і взагалі прагнення зробити прототип можливо більш схожим на результуючу систему. У самому такому підході немає нічого поганого (все одно певні частини прототипу доводиться робити максимально досконалыми), проблема в тому, що в більшості випадків прототип після тестування виявляється неправильним. Його доводиться переробляти. Тому завжди правильно робити прототип настільки схожим на результуючу систему, наскільки версія прототипу пізня. Перший прототип варто робити максимально примітивним. Тільки після того, як тестування підтверджує його правильність, варто робити більш деталізований прототип.

Перша версія. Паперова. Необхідно намалювати на папері всі екрани і діалогові вікна (роздрукувати відповідні частини схеми). Потрібно тільки переконатися, що всі інтерфейсні елементи виглядають одноманітно і скільки схоже на реальні. Ця роздруківка і є першим прототипом. На ньому цілком можна тестувати сприйняття системи користувачем і її основну логіку. Користь початкового прототипування на папері полягає, по-перше, у виключній простоті модифікації за результатами тестування, а по-друге, у можливості безболісно відловлювати представників цільової аудиторії. Зрозуміло, значення слова «версія», досить умовно. Після виявлення кожної помилки схема і прототип виправляються, а тестування триває вже на новому прототипі. Так що на цьому етапі прототип може пережити безліч виправлень і, відповідно, багато версій.

Друга версія. Презентація. Після вичерпання можливостей паперовій версії прототипу варто створити нову версію (виправивши, зрозуміло, вже виявлені проблеми). Для цього так само малюється інтерфейс, але вже не на папері, а в будь-якій презентаційній програмі (Adobe Photoshop або MS Power Point, наприклад). Кожна сторінка отримує окремий файл або слайд, а результат натискання кнопок імітується переходами між файлами або слайдами. З цією версією прототипу можна тестувати значно більш складну взаємодію людини з системою, ніж з паперовою.

4.5. Тестування і модифікація прототипу

Якими б не були досконалыми логічні міркування, що призвели до створення інтерфейсу, завжди залишається ймовірність того, що інтерфейс вийшов поганий, або, що більш ймовірно, не такий хороший, яким би він міг бути. Необхідно мати які-небудь підтвердження його працездатності. На щастя, перевірка якості інтерфейсу зазвичай неproblemатична. Все, що для цього потрібно, це кілька користувачів середньої кваліфікації, що ніколи не бачили тестованої системи, плюс прототип.

У літературі часто зустрічається думка, що тестуванням можна вирішити чи не всі проблеми інтерфейсу. Твердження це сумнівне. Тестуванням, швидше, можна визначити слабкі місця інтерфейсу, але майже неможливо виявити сильні, оскільки вони користувачами просто не помічаються, і зовсім вже неможливо визначити нові способи поліпшення. Відбувається це через те, що суб'єкти тестування: не володіють всією необхідною інформацією про систему; нічого не знають про проектування інтерфейсів; їх мотивація суттєво відрізняється від необхідної - замість того, щоб прагнути зробити хороший інтерфейс, вони прагнуть позначити в цьому свою участь.

Взагалі, слухати споживачів зазвичай неправильно. У той же час, навіть не слухаючи користувачів, обов'язково потрібно брати до уваги їхні потреби, здібності й уподобання. Наприклад, нерідко дизайнер інтерфейсу знає про предметну область менше, ніж майбутні користувачі. У таких умовах втрата контакту з користувачами загрожує крахом продукту просто тому, що система виявляється нездатною вирішувати завдання, про які дизайнер нічого не знав. Частіше, однак, трапляється так, що рівень «комп'ютерної грамотності» дизайнера виявляється вище рівня аудиторії. У цьому випадку все виходить ще гірше: дизайнер вибирає рішення, які забезпечують ефективність роботи, а споживачам потрібні рішення, які вони можуть зрозуміти, в результаті вони виявляються нездатними скористатися системою (це абсолютно нормальна ситуація, особливо в інтернеті). Наприклад, помічено, що досвідчені користувачі (до яких відносяться дизайнери) створюють значно менш працездатні ієрархії меню, ніж користувачі початківці. Зрозуміло, якщо переоцінка

здібностей реальних користувачів і незнання предметної області збігаються, результат буває ще гірше.

Постановка завдання. Однією з найважливіших передумов успішного тестування є правильна постановка задачі. Завжди є шанси витратити кілька годин у пошуках відповіді на непотрібні питання. Гірше того — трапляється, що після закінчення тривалого і стомлюючого сеансу приходить розуміння того, що тих же результатів можна було б домогтися з меншими трудовитратами. Правильна постановка завдання дозволяє цих проблем уникнути [7].

Власне тестування. Технічно сеанс тестування досить простий. Потрібно мати кілька користувачів, які жодного разу не бачили поточного стану системи. За винятком рідкісних випадків, коли система розрахована на більш досвідчених користувачів. Ідея тестування проста, тим не менш, з цього приводу написано досить багато літератури. Втім, у більшості випадків достатньо пам'ятати наступне: тестування на одному користувачеві дозволяє знайти приблизно 60% помилок. Відповідно слід вирішити, скільки користувачів необхідно для одного сеансу.

4.6. Розробка структурної схеми алгоритму проектування інформаційної моделі інтерфейсу користувача

За основу розробки обираємо результати аналізу інформаційних джерел [12-18]. При формуванні структурної схеми алгоритму проектування інформаційної моделі інтерфейсу користувача максимально враховуємо інформацію щодо класифікації моделей користувацьких інтерфейсів, зображену в Додатку на рис. А.1, адаптованого з ресурсу [14].

Для виконання подальших проектних рішень обираємо в якості прототипа наукову проблематику створення електронних систем управління відносинами з клієнтами. По даній тематиці у автора кваліфікаційної роботи є наукова публікація, яку і покладаємо в основу подальших дій [10]. Адаптуємо з даного ресурсу зображення (Рис. Б.1. Комплексна модель електронної системи управління

відносинами з клієнтами), на якому представлені чотири основні бізнес-процеси, інтегровані в систему eCRM, кожний із яких потребує розробки інтерфейсних рішень. Модель показує, як вихідні дані одного етапу представляють собою входи для наступного етапу. Отримані поточні дані на закінчення процесу замикає цикл і перезапускає процес для кращого налаштування діяльності компанії до потреб клієнтів. Вся система eCRM має використовувати програмні продукти (додатки) взаємодії з Web-процесами для кращого налаштування взаємодії до потреб клієнтів.

До моделі можуть бути включені додаткові складові структури інтерфейсу.

Отже, перший етап пройдено. Є варіанти 2-х моделей, рис. 4.1, адаптований з [14] та рис. 4.2, який є результатом авторських досліджень (Головня К.Ю. Електронні системи управління відносинами з клієнтами. // Науково-практична конференція „Мультимедійні технології в освіті та інших сферах діяльності”: Тези доповідей. – К.: НАУ, 2021), на основі яких розробляємо структурну схему алгоритму проектування інформаційної моделі інтерфейсу користувача.

Після кожного етапу передбачуємо валідацію, що дозволяє створити більш якісний інтерфейс з наданням розширених сервісних послуг користувачу.

Висновки до четвертого розділу

Якщо на запит - розробляти інтерфейс простим у використанні з урахуванням людських можливостей і обмежень, слід виконати наступні дві дії. Перше – зрозуміти, що можна робити і чого не можна, вивчити психологію людського мислення на основі когнітивних технологій. Дослідження показують, що наші здібності розділяються між когнітивним свідомим і когнітивним несвідомим; що формування звичок відіграє центральну роль в тому, як ми реагуємо на ті чи інші види функціонування інтерфейсів. Індивідуальні відмінності при формуванні звичок невеликі в порівнянні з іншими відносинами. Все це закладено в алгоритм проектування моделі, який представлений у Додатку В.1.

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи отримані наступні результати:

- доведена актуальність досліджень у галузі проектування інтерфейсів електронних ресурсів;
- визначені основи проектування засобів інтерфейсів людино-машинної взаємодії та етапи проектування користувацьких інтерфейсів;
- розглянуті особливості проектування інтерфейсів з урахуванням мультимедійного контенту електронних ресурсів;
- досліджена можлива уявна взаємодія користувача з елементами інтерфейсу;
- для створення алгоритму проектування інтерфейсів використані варіанти 2-х моделей, рис. А.1, адаптований з [14], та рис. В.1, який є результатом авторських досліджень здобувача [3] (Головня К.Ю. Електронні системи управління відносинами з клієнтами. // Науково-практична конференція «Мультимедійні технології в освіті та інших сферах діяльності»: Тези доповідей. – К.: НАУ, 2021.).
- розроблена структурна схема алгоритму проектування інформаційної моделі інтерфейсу користувача з урахуванням визначеної кількості складових (етапів) з валідацією після кожного етапу. Алгоритм представлений у Додатку (рис. В.1).

Проведені дослідження щодо методичних основ комплексного проектування користувацьких інтерфейсів орієнтують на використання eCRM для моделювання, здатного представити як організаційну, так і функціональну архітектуру на рівні системи, процесу, програмних додатків і ресурсів з урахуванням потреб клієнтів.

Для практичних розробок інтерфейсних рішень рекомендується комплексний підхід використання результатів досліджень наступних моделей, представлених в Додатках на рис. А.1, рис. Б.1, рис. В.1.

Сукупність одержаних результатів досліджень забезпечує досягнення мети в кваліфікаційній роботі щодо проектування користувацьких інтерфейсів.

СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Сучасні технології електронних мультимедійних видань: монографія / Під ред. О. І. Пушкаря. – Харків: ВД «ІНЖЕК», 2011. – 296 с.
2. Поморова, О. В. Проектування інтерфейсів користувача [Текст] : навч. посіб. / О. В. Поморова, Т. О. Говорущенко. – Хмельницький : ХНУ, 2011. – 206 с.
3. Головня К.Ю. Електронні системи управління відносинами з клієнтами. // Всеукраїнська науково-практична конференція з міжнародною участю «Мультимедійні технології в освіті та інших сферах діяльності». kmmmt/nau.edu.ua/xi – всеукраїнська науково-практична он-лайн конференція / м. Київ, НАУ. Дата проведення 12.11.2021.
4. BS EN ISO 14915-1:2002. Software ergonomics for multimedia user interfaces. Design principles and framework. – British Standard/European Standard/International Organization for Standardization/10-Dec-2002/24 pages. – ISBN: 0580409236.
5. Гультьяев А. К. Проектирование и дизайн пользовательского интерфейса / Гультьяев А. К., Машин В. А. – Санкт-Петербург, Корона-Принт, 2007. – 239 с.
6. Головач В. В. Дизайн пользовательского интерфейса / В. В. Головач; Искусство мыть слона. – 2010. – 94с.
7. Раскин Дж. Интерфейс: новые направления в проектировании компьютерных систем / Дж. Раскин. – Санкт-Петербург, Символ-Плюс, 2005. – 272 с.
8. Тидвелл Д. Разработка пользовательских интерфейсов / Д. Тидвелл. – Санкт-Петербург, Питер – 2008. – 416 с.
9. Будаева А.А. Учебное пособие по курсу «Интерфейсы АСОИУ» для студентов направления «Информатика и вычислительная техника» / А. А. Будаева. – Владикавказ, 2008. – 150 с.
10. Алан Купер. Об интерфейсе. Основы проектирования взаимодействия / Купер А. – Санкт-Петербург, Символ-Плюс, 2009. – 688 с.
11. Usability-методы исследования Web-сайта. [Електронний ресурс]. – Режим

доступу: <http://usability.ru/Articles/um.htm>. (дата звернення: 05.04.2022) – Назва з екрана.

12. Двадцять принципів проектування інтерфейсів. [Електронний ресурс]. – Режим доступу: <http://actualweb.ru/2012/05/22/20-принципов-проектирования-интерфейс/> (дата звернення: 07.04.2022) – Назва з екрана.

13. Користувальницька і програмна моделі інтерфейсу [Електронний ресурс]. – Режим доступу: http://bricks.narod.ru/tp/33_pol.htm. (дата звернення: 09.04.2022) – Назва з екрана.

14. Типи користувацьких інтерфейсів та етапи їх розробки [Електронний ресурс]. – Режим доступу: <https://studfile.net/preview/7436131/page:21/> (дата звернення: 15.04.2022) – Назва з екрана.

15. Моделі користувацького інтерфейсу [Електронний ресурс]. – Режим доступу: <https://ppt-online.org/256375> (дата звернення: 17.04.2022) – Назва з екрана.

16. Проектування інтерфейсів [Електронний ресурс]. – Режим доступу: https://studopedia.su/12_88956_proektirovanie-interfeysov.html (дата звернення: 18.04.2022) – Назва з екрана.

17. Генератор кода пользовательского интерфейса, управляемый онтологиями [Електронний ресурс]. – Режим доступу: <https://pandia.ru/text/77/376/25131.php> (дата звернення: 21.04.2022) – Назва з екрана.

18. Дизайн і проектування користувацького інтерфейсу [Електронний ресурс]. – Режим доступу: <https://dlogic.com.ua/service/proektirovanie-interfeisov/> (дата звернення: 25.04.2022) – Назва з екрана.

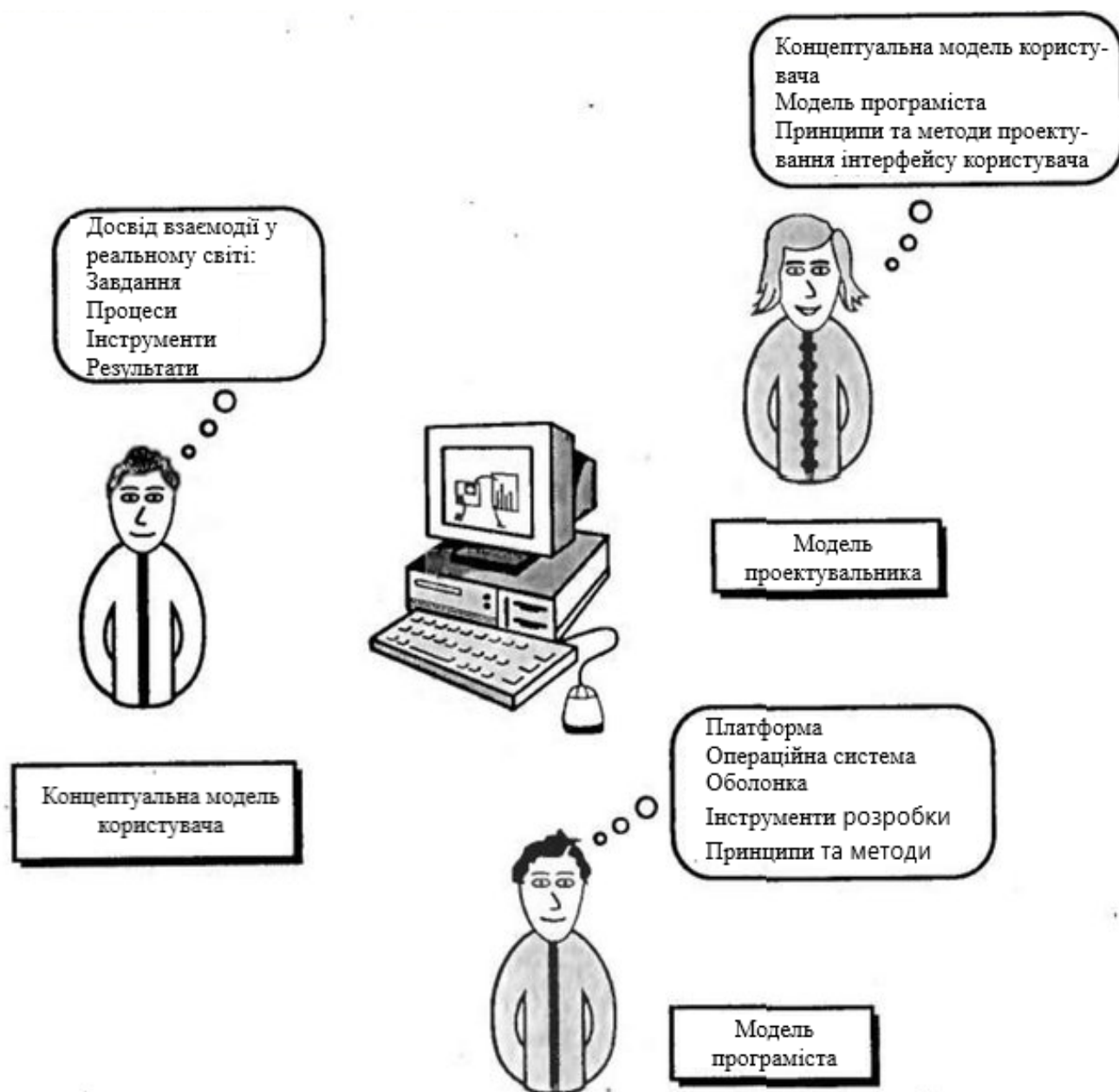


Рис. А.1. Класифікація моделей користувацького інтерфейсу (адаптовано з [14])

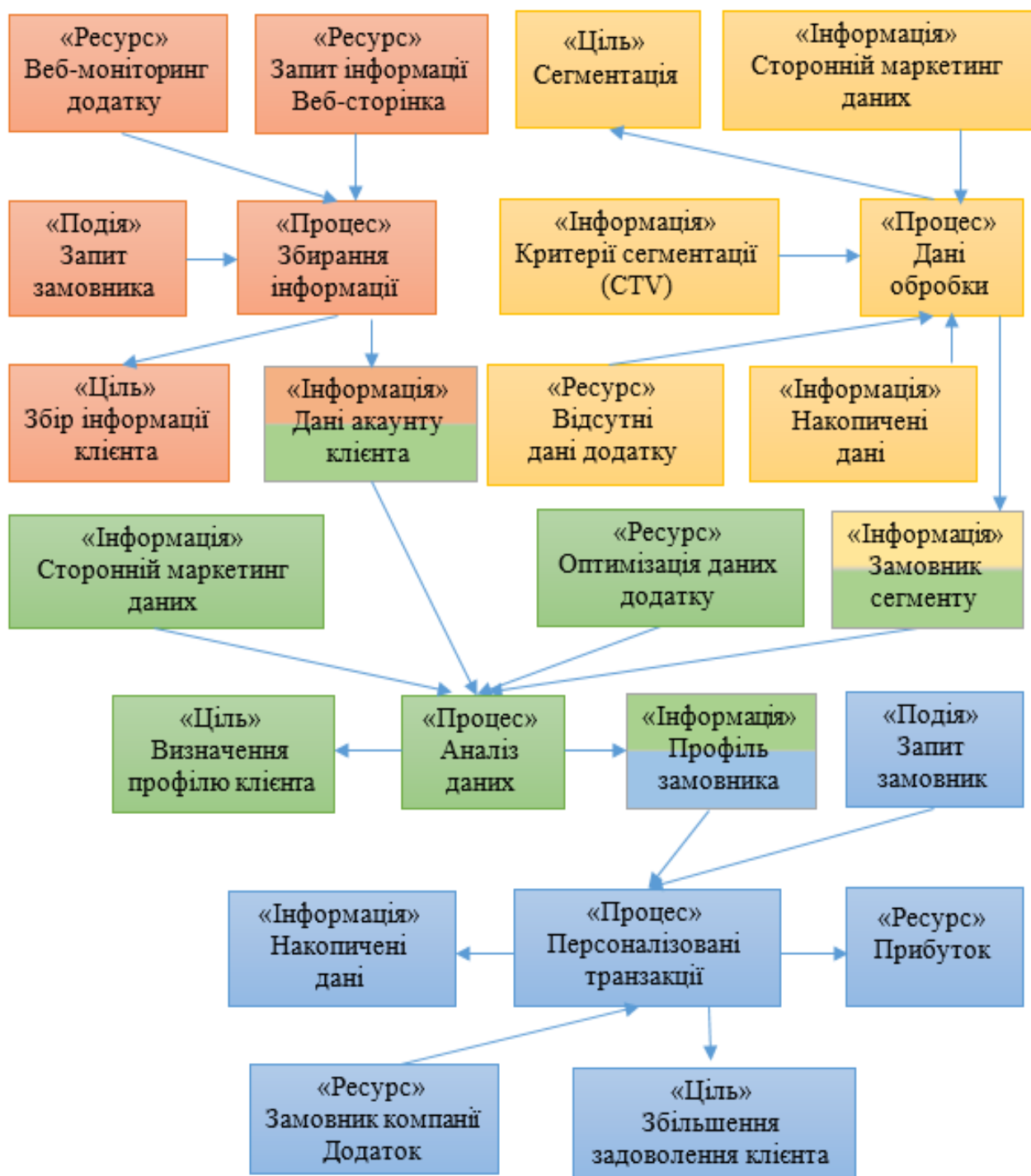


Рис. Б.1. Комплексна модель електронної системи управління відносинами з клієнтами (адаптовано з [3], ресурс здобувача)

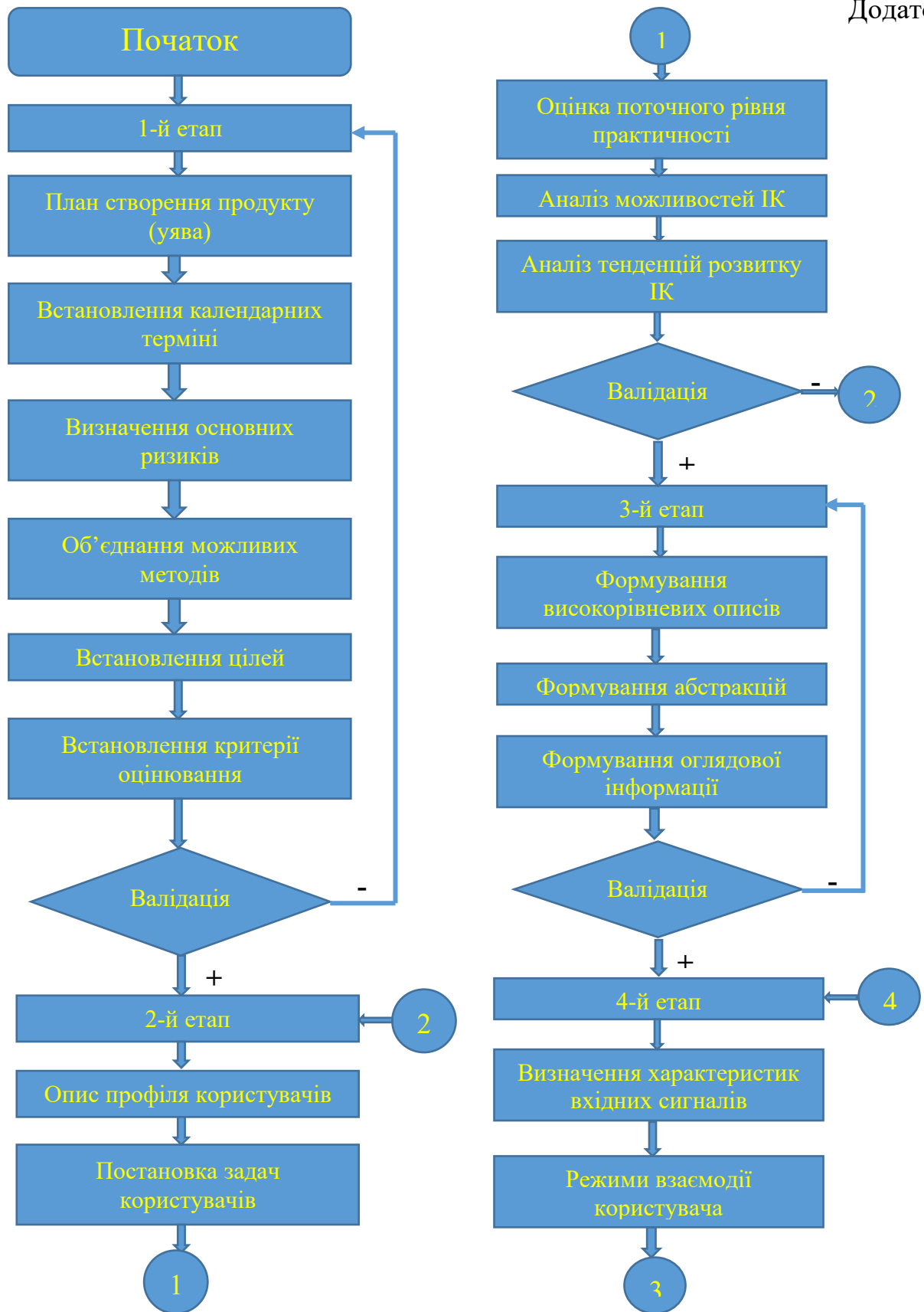


Рис. В.1. Структурна схема алгоритму проектування інформаційної моделі інтерфейсу користувача з застосуванням валідації на кожному етапі

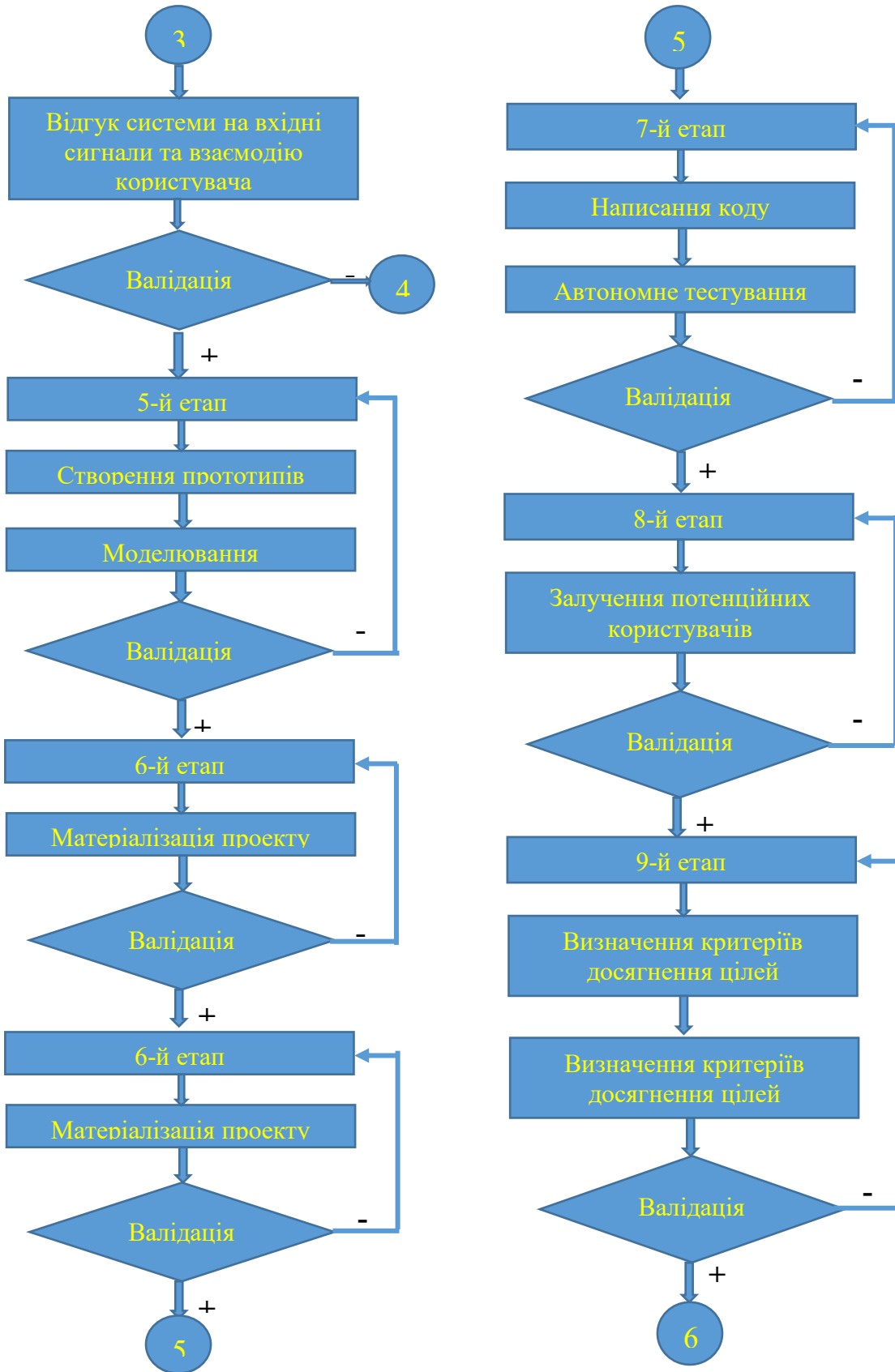


Рис. В.2. Структурна схема алгоритму проектування інформаційної моделі інтерфейсу користувача з застосуванням валідації на кожному етапі

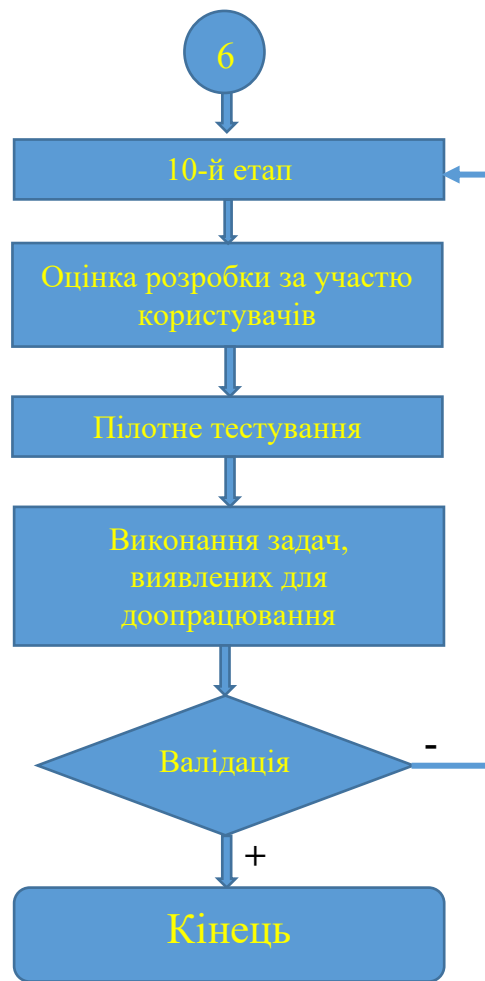


Рис. В.3. Структурна схема алгоритму проектування інформаційної моделі інтерфейсу користувача з застосуванням валідації на кожному етапі