

Міністерство освіти та науки України
Національний університет «Львівська політехніка»

На правах рукопису

Маркелов Олександр Едуардович

УДК 004.4'22:004.5:004.414.32

**Інструментальні засоби
автоматизації синтезу інтерфейсів
користувача на базі репозиторію
інтелектуальних патернів**

Спеціальність 05.13.12 – системи автоматизації проектувальних робіт

Дисертація на здобуття наукового ступеня
кандидата технічних наук

Науковий керівник:
Лобур Михайло Васильович,
д.т.н., професор кафедри САП
Національного університету
«Львівська політехніка»

Львів-2017

Зміст

Перелік умовних позначень.....	6
Список рисунків	7
Список таблиць	12
Вступ.....	13
Розділ 1. Сучасний стан побудови інтерфейсів користувачів ПЗ.....	25
1.1. Світові тенденції	25
1.2. Основні поняття інтерфейсів користувача	27
1.3. Діалогові процеси інтерфейсних взаємодій.....	29
1.4. Різновиди концепцій інтерфейсів користувачів	31
1.4.1. Інтерфейс командного рядка (CLI) і текстовий (TUI)	31
1.4.2. Графічний інтерфейсу користувача (GUI).....	33
1.4.3. Інтерфейс імітації кінетичних взаємодій цифрових об'єктів	34
1.4.4. Інтерфейс доповненої реальності (Augmented reality UI)	35
1.4.5. Матеріальний інтерфейс користувача (Tangible UI).....	36
1.4.6. Жестовий інтерфейс користувача (Gestures UI).....	37
1.4.7. Розчерковий інтерфейс (Calligraphic interface)	39
1.4.8. Мімічний користувацький інтерфейс (Mimic UI).....	40
1.4.9. Масштабований інтерфейс користувача (Zooming UI).....	40
1.4.10. Фізично трансформований інтерфейс користувача.....	41
1.4.11. Соціалізований інтерфейс користувача.....	42
1.5. Систематизація інтерфейсів користувача.....	43
1.5.1. Ієрархічна систематизація користувацьких взаємодій	43
1.5.2. Класифікація кінетично-орієнтованих інтерфейсів	43
1.6. Проектні процеси побудови інтерфейсів користувача	44
1.6.1. Проектування зручності та досвіду користувача.....	44
1.6.2. Складові процесу ергономічного проектування ПЗ	46
1.7. Огляд засобів проектування інтерфейсів користувача.....	47

1.7.1. Інтегровані середовища розробки ПЗ (IDE)	47
1.7.2. Середовища прототипування інтерфейсів ПЗ.....	50
1.8. Мови декларування користувацьких інтерфейсів.....	52
1.8.1. Основні характеристики мов декларування інтерфейсу.....	52
1.8.2. Порівняння характеристик мов розмітки UI.....	54
1.9. Висновки до розділу	56
Розділ 2. Метрики, конструктиви та методи побудов інтерфейсу користувача.....	57
2.1. Моделювання факторних параметри впливу користувача.....	57
2.2. Моделювання досвіду користувача накопиченням діалогової інформаційної діяльності	61
2.2.1. Формалізація функціонального наповнення (профілю) площинного компонування інтерфейсу.....	62
2.2.2. Формалізація взаємодій користувача інтерфейсу програмного забезпечення.....	64
2.2.3. Кінематичні індекси активності користувача.....	65
2.2.4. Візуалізація накопичених даних на зразках інтерфейсу.....	66
2.3. Систематизація візуальних конструктивів інтерфейсу.....	68
2.3.1. Типізація конструктивів графічного інтерфейсу.....	69
2.3.2. Спеціальні елементи GUI	76
2.4. Процес поділу поля макетів каркасів компонування ГІК.....	78
2.5. Метод оціночних характеристик засобів прототипування ГІК	83
2.6. Метод оціночних характеристик мов програмування ІК.....	89
2.7. Метод оцінювання інформаційних інструментів моніторингу функціонування ГІК ПЗ	96
2.8. Процес моніторингу інтерактивних взаємодій користувача із ГІК.....	98
2.9. Процес перетворення графічних прототипів віконних форм інтерфейсу користувача в програмний код.....	101

2.9.1. Основні кроки процесу ідентифікації конструктивних прототипів ГІК.....	102
2.9.2. Алгоритмізація процесу ідентифікації конструктивів КІ та синтезу	105
2.10. Процес перепроєктування інтерфейсу.....	106
2.11. Висновки до розділу	108
Розділ 3. Реалізація засобів автоматизації процесів інтерфейсних побудов	109
3.1. Засоби для накопичення діалогових взаємодій	109
3.1.1. Підсистема програмного АРІ доступу до інтерфейсних елементів та моніторингу подій.....	110
3.1.2. Підсистема локального накопичення даних.....	113
3.1.3. Підсистема візуалізації статистичної густини інтерактивної взаємодії.....	114
3.1.4. Серверна частина.....	114
3.1.5. Інформаційна модель збереження даних.....	117
3.1.6. Інформаційна інтеграція підсистем накопичення даних.....	118
3.2. Засоби формування репозиторію конструктивних елементів ІК.....	119
3.3. Засіб автоматизації візуальної ідентифікації конструктивів прототипів ГІК.....	121
3.3.1. Архітектурне рішення функціонування засобу	121
3.3.2. Ролі користувачів системи ідентифікації компонентів ГІК.....	123
3.4. Інтеграція засобів накопичення і метрик діалогових взаємодій з даними репозиторіїв патернів	125
3.5. Засоби реінженірінгу програмного коду інтерфейсів користувача.....	126
3.5.1. Концептуальна модель БД.....	129
3.5.2. Реляційна модель бази даних.....	131
3.6. Інформаційна система параметризованого порівняння засобів прототипування інтерфейсів користувача.....	135
3.7. Висновки до розділу	138

Розділ 4. Результати дослідження та прототипування інтерфейсів користувача.....	139
4.1. Статистичні оцінки аналізу використання конструктивів ГК.....	139
4.1.1. Оцінка реалізації інтерфейсних компонентів у САПР схемотехніки	139
4.1.2. Кількісна оцінка застосованих інтерфейсних компонентів у САПР машинобудування.....	140
4.2. Результати якісної оцінки інструментів моніторингу подій ГК ПЗ	149
4.3. Діалогове прототипування сценарії взаємодій користувача.....	152
4.4. Побудова сценарію діалогів для графічного редактора схем ІК.....	152
4.5. Перевірка автоматизації ідентифікації компонентів інтерфейсів користувача із графічних ескізів	157
4.6. Висновки до розділу	160
Висновки	161
Список використаних джерел	164
Додаток А. Акт впровадження в навчальний процес	183
Додаток Б. Акт впровадження (використання) в діяльності ТзОВ.....	184

Перелік умовних позначень

Скорочення, термін, позначення	Пояснення
англ.	Англійська мова.
2D	англ. – 2-Dimension. Два виміри. Площина. Двовимірна комп'ютерна графіка.
3D	англ. – 3-Dimension. Три виміри. Простір. Тривимірна комп'ютерна графіка.
API	англ. – Application programming interface. Інтерфейс програмування прикладних програм.
CAD	англ. – Computer-Aided Design. Автоматизоване проектування. Системи автоматизованого проектування. САПР
DB	англ. – Data base. База даних
GUI	англ. – Graphic User Interface. Графічний інтерфейс користувача.
HCI	англ. – Human-computer interaction. Людино-комп'ютерна взаємодія
HMI	англ. – Human-machine interface. Людино-машинний інтерфейс
ID	англ. – Identifier. Ідентифікатор
IDE	англ. – Integrated Development Environment. Інтегроване середовище розробки.
ISO	англ. – International Organization for Standardization. Міжнародна організація із стандартизації.
UI	англ. – User Interface. Інтерфейс користувача.
UX	англ. – User Experience. Досвід користувача.
UXD	англ. – User Experience Design. Проектування досвіду користувача
WYSIWYG	англ. – “What You See Is What You Get”. “Що бачиш, те й отримаєш”
WWW	англ. – World Wide Web. Система зв'язаних гіпертекстових документів у мережі Інтернет.
APM	Автоматизоване робоче місце.
ГІК	Графічний інтерфейс користувача. англ. – GUI
ДС	Діалогова система.
ІК	Інтерфейс користувача.
ІТ	Інформаційні технології.
ЕОМ	Електронно-обчислювальна машина.
ЛТС	Система «людина - техніка - середовище».
ПЗ	Програмне забезпечення.
САПР	Система автоматизованого проектування. англ. – CAD

Список рисунків

Рис. 1.1. Запити на дозвіл збирання даних про дії користувачів ПЗ ABBYY, Adobe, Google, Microsoft, Ahead	26
Рис. 1.2. Приклади текстового інтерфейсу користувача ПЗ.....	33
Рис. 1.3. Приклади графічного інтерфейсу користувача ПЗ.....	34
Рис. 1.4. Приклади інтерфейсу користувача з імітацією фізики.....	34
Рис. 1.5. Приклади інтерфейсу користувача доповненої проєкції.....	35
Рис. 1.6. Приклади інтерфейсу користувача з «матеріальними-бітами»	37
Рис. 1.7. Приклади жестового інтерфейсу користувача для «миші»	38
Рис. 1.8. Приклади жестового інтерфейсу користувача пальцями рук.....	38
Рис. 1.9. Ідея прототипу керування програмами жестами частин тіла	39
Рис. 1.10. Приклади пристроїв із розчерковим інтерфейсом.....	39
Рис. 1.11. Приклади ідеї прототипу керування мімікою	40
Рис. 1.12. Приклад масштабованого інтерфейсу	41
Рис. 1.13. Приклад концепції фізично трансформованого інтерфейсу.....	41
Рис. 1.14. Приклади соціалізованого інтерфейсу	42
Рис. 1.15. Схема систематизації діалогових користувацьких взаємодій ПЗ.....	43
Рис. 1.16. Схема кінематичних підходів проєктування інтерфейсів	44
Рис. 1.17. Схема етапів проєктування зручності	45
Рис. 1.18. Схема впливу галузей при проєктуванні досвіду користувача	45
Рис. 1.19. Схема видів послуг з проєктування досвіду користувача.....	46
Рис. 1.20. Приклади віконних інтерфейсів IDE з редакторами WYSIWYG і їхні набори елементів управління користувацьким інтерфейсом.....	49
Рис. 1.21. Приклади середовищ прототипування інтерфейсів користувача.....	50
Рис. 1.22. Приклади онлайн-сховища елементів інтерфейсу користувача UICloud	51

Рис. 2.1. Геометричне 3D представлення вагових ступенів функціонального призначення пікселя інформаційного поля інтерфейсу.....	63
Рис. 2.2. Геометрична 3D стовпцева декомпозиційна моделі інтерпретації областей екрану інтерфейсу користувача.....	64
Рис. 2.3. Різні точності реєстрації за рухом маніпулятора типу «миша»	66
Рис. 2.4. Віконні інтерфейси LayoutEdit of MEMS/IC CAD Tool.....	67
Рис. 2.5. Графічна візуалізація траєкторії «миші», часові записи користувацької активності дій з вікном LayoutEdit of MEMS/IC CAD Tool.	67
Рис. 2.6. Узагальнена «температурна карта» користувацької активності дій з інтерфейсами LayoutEdit of MEMS/IC CAD Tool.....	68
Рис. 2.7. Приклади графічних кнопок.....	77
Рис. 2.8. Приклади кільцевого меню (Pie menu).....	77
Рис. 2.9. Приклад циклічної кнопки (Cycle Button).....	78
Рис. 2.10. Приклади «тостових» виринаючих нагадувань (Toast).....	78
Рис. 2.11. Поділ розміщення компонент на графічному макеті програми	80
Рис. 2.12. Приклад поділу макету форми для програми AkeIPad.....	81
Рис. 2.13. Класифікаційний поділ макетів інтерфейсу користувача.....	82
Рис. 2.14. Класифікаційний поділ макетів програм з пропорціями максимальних розмірів	83
Рис. 2.15. Вагові бали загальної оцінки у часткових категоріях	92
Рис. 2.16. Діаграми співвідношення показників із врахуванням ваг категорій оцінки мов декларування користувацьких інтерфейсів.....	93
Рис. 2.17. Пелюсткова діаграма співвідношень показників у категоріях оцінки мов користувацьких інтерфейсів.....	93
Рис. 2.18. Стовпцева діаграма процентних співвідношень показників у категоріях оцінки мов користувацьких інтерфейсів.....	95
Рис. 2.19. Пелюсткова діаграма співвідношень показників у категоріях оцінювання інструментів для моніторингу	98
Рис. 2.20. Схема процесу ідентифікації та запису подій курсору «миші».....	99

Рис. 2.21. Схема процесу аналізування подій від «миші»	100
Рис. 2.22. Приклади ескізів графічних прототипів віконних форм для ідентифікації шаблонних конструктивних елементів ГІК.....	102
Рис. 2.23. Схема алгоритму процесу ідентифікації змістовних конструктивів ГІК та синтезу програмного коду.....	105
Рис. 2.24. Схема процесу автоматизації перепроєктування інтерфесу з врахуванням метрик досвіду користування і програмним патернів.....	107
Рис. 3.1. Схема зв'язків локальної бази даних інтерактивних дій користувача	110
Рис. 3.2. Прототип вікна вибору досліджуваної системи.....	111
Рис. 3.3. Вибір сценарію для відтворення	111
Рис. 3.4. Інтерфейс програми під час відтворення	111
Рис. 3.6. Налаштування колірної палітри візуалізації	112
Рис. 3.7. Вікно створення сценарію	112
Рис. 3.8. Вікно менеджера сценаріїв.....	112
Рис. 3.9. Функціональна схема підсистеми відслідковування та накопичення даних про інтерактивну взаємодію користувача з ПЗ.....	113
Рис. 3.10. Функціональна схема підсистеми візуалізації статистично оброблених даних інтерактивної взаємодії	114
Рис. 3.11. Схема інформаційних складових серверної частини.....	116
Рис. 3.12. Схема бази накопичення даних взаємодій користувача.....	117
Рис. 3.13. Загальна схема архітектуру інформаційного засобу накопичення даних взаємодій користувача	118
Рис. 3.14. Схема бази репозиторію конструктивних елементів ІК.....	119
Рис. 3.15. Схема технологічних зв'язків інформаційних технологій	122
Рис. 3.16. Схема алгоритму функціонування серверного модуля	122
Рис. 3.17. Схема класів ООП клієнтського модуля	123
Рис. 3.18. Use case діаграма ролей користувачів системи	124

Рис. 3.19. Реляційна модель даних інтеграції даних з репозиторію конструктивних елементів інтерфейсу та даних накопичення інтерактивного використання інтерфейсу користувача.....	125
Рис. 3.22. Концептуальна модель ООП структури класу	130
Рис. 3.23. Концептуальна модель семантичного аналізу.....	130
Рис. 3.24. Реляційна модель бази даних патернів перетворення програмних кодів інтерфейсів	133
Рис. 3.25. Головне вікно системи аналітичного пошуку засобу прототипування інтерфейсу	137
Рис. 4.1. Основні типові віконні форми ГІК FreeCAD	141
Рис. 4.2. Приклади спроектованих прототипів ГІК для на базі FreeCAD.....	142
Рис. 4.3. Основні типові віконні форми ГІК TurboCAD.....	142
Рис. 4.4. Приклади спроектованих прототипів ГІК для на базі TurboCAD.....	143
Рис. 4.5. Віконні форми ГІК «налаштування» проектів у CADwork 2D.....	143
Рис. 4.6. Приклади спроектованих прототипів ГІК для progeCAD	144
Рис. 4.7. Основні типові віконні форми ГІК SALOME.....	144
Рис. 4.8. Приклади синтезованого прототипу ГІК для SALOME.....	145
Рис. 4.9. Приклади фрагментів програмного коду мовами XUL та XAML.....	146
Рис. 4.10. Просторова гістограма спектру компонентів ГІК у САПР машинобудування.....	147
Рис. 4.11. Колова діаграма співвідношень розподілу компонентів ГІК у САПР машинобудування.....	148
Рис. 4.12. Загальна оцінка досліджених інструментів моніторингу ІК	151
Рис. 4.13. Діаграма переходів сценарію діалогу графічного редактора схем ІК153	
Рис. 4.14. Діаграма переходів сценарію діалогу для роботи з графічною схемою.....	154
Рис. 4.15. Діаграма переходів сценарію діалогу для редактора бібліотек	155
Рис. 4.16. Діаграма переходів сценарію діалогу для попереднього перегляду друку.....	155

Рис. 4.17. Растровий приклад тестового графічного прототипу віконної форми інтерфейсу.....	158
Рис. 4.18. Виділення автоматично сегментованих позиціонування тексту та контурів ГІК	158
Рис. 4.19. Приклад автоматизовано згенерованого програмного коду прототипу інтерфейсної форми.....	159

Список таблиць

Таблиця 1.1.	IDE з конструкторами графічного інтерфейсу користувача.....	48
Таблиця 1.2.	Порівняльний аналіз засобів прототипування ІК.....	51
Таблиця 1.3.	Порівняльна таблиця мов розмітки UI.....	55
Таблиця 2.1.	Типи конструктивів графічного інтерфейсу користувача.....	69
Таблиця 2.2.	Співвідношення ваг критеріїв для оцінки прототипування.....	85
Таблиця 2.3.	Вагові розподіли у групових категоріях показників оцінки мов розмітки користувацьких інтерфейсів у застосуванні розробки ПЗ САПР	92
Таблиця 2.4.	Вагові розподіли у групових категоріях показників оцінки інструментів для моніторингу по функціональності	97
Таблиця 2.5.	Приклади графіків автокореляційної функції для елементів графічного інтерфейсу користувача	104
Таблиця 4.1.	Спектр застосування GUI компонентів в САП схемотехніки....	140
Таблиця 4.2.	Порівняльна таблиця оцінки інструментів для моніторингу за функціональністю	149
Таблиця 4.3.	Порівняльна таблиця оцінки інструментів для моніторингу за критеріями доступу до інтерфейсу користувача.....	150
Таблиця 4.4.	Матриця взаємодій об'єктів сценарію діалогів користувача.....	156

Вступ

Користувацький інтерфейс є одною з найбільш об'ємних і найбільш трудомістких частин розробки будь-якої системи автоматизованого проектування (САПР). Саме завдяки зручності й продуманості інтерфейсів користувача (ІК) значно підвищується ефективність та швидкість виконання проектувальником поставленої задачі. Практично усі сучасні САПР створені з використанням програмування користувацьких інтерфейсів (КІ) на базі стандарту інтерфейсних елементів операційної системи (ОС). Але ця ситуація є зручною з огляду на те, що якщо користувач навчився користуватися одною в програм, то він швидко засвоїть іншу. Водночас тривале використання програмного забезпечення (ПЗ) САПР спонукає досвідченого користувача-проектувальника з великим досвідом використання ПЗ САПР і досвідом у проектуванні адаптувати інтерфейси ПЗ до власних потреб з метою скорочення власного робочого часу доступу до типових інструментів – інтерфейсних об'єктів керування. У такому випадку забезпечення зручності кожному професіоналу-проектувальнику – це новий, адаптований інтерфейс, новий інформативний дизайн користувацької оболонки САПР. І тому, як наслідок, застосування програмних бібліотек для одного певного стандарту інтерфейсів є «свідомим» нав'язування політики обмеження свободи організації робочого місця проектувальника. Для розроблення інтерфейсів, використовується переважно «ручне» програмування, основою якого від початку є позасистемне задавання візуального оформлення інтерфейсів, програмних реакцій та станів для кожного з елементів користувацького інтерфейсу. При розробленні ПЗ великих САПР, цей «несистемний» підхід означає «початок кінця». Щоб зменшити негативні наслідки такого стану справ, при проектуванні ПЗ САПР варто звернутися до існуючих та перспективних стандартів у галузі опису

користувацьких інтерфейсів, які є об'єктами численних досліджень [1, 2, 3, 4, 5, 6, 7, 78].

Більшість сучасних популярних середовищ розробки програмного забезпечення [50, 51, 52, 53, 55] та прототипування ескізів інтерфейсів [56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72] не дозволяють здійснювати альтернативний вибір серед візуальних реалізацій елементів управління користувацького інтерфейсу чи то пошуку, підбору альтернативних варіантів програмних вихідних кодів реалізації цих елементів. Набір елементів інтерфейсу проектувальних ПЗ є як наслідок вибору самого середовища програмування чи прототипування інтерфейсів. Тому задача підбору власного набору елементів інтерфейсу для конкретного проекту не є незалежною задачею від вибору середовища програмування.

Внески у дослідження моделями поведінки й активності користувачів в системах “людина-техніка”, адаптивності інтерфейсів зробили: Смирнов Б.А., Гулий Ю.І., Погребняк Т.П., Орехова А.А., Зянчуріна І.М. (Національний аерокосмічний університет ім. М.Є. Жуковського “Харківський авіаційний інститут”); Ходаков Д.В., Радванська Л.М., Бень А.П. (Херсонський національний технічний університет); Січкаренко В.О., Пшонківська І.М., Зайчева С.В. (НАН України, Інститут кібернетики імені В.М.Глушкова); Пантелеймонов А.А. (Нац. техн. ун-т України «Київ. політехн. ін-т»); Скакун С.В. (НАН України. Нац. косм. агентство України. Ін-т косм. дослідж.); Ковальчук А.М. (НАН України, Інститут проблем моделювання в енергетиці ім. Г.Є. Глухова); Ночевнов Д.П. (Черкаський державний технологічний університет); реінженерія застосувань із графічним інтерфейсом: Пантелеймонов А.А. (Національний технічний університет України «Київський політехнічний інститут»). Серед іноземних дослідників слід виділити: Robert Torres, Teo Mandel, Jakob Nielsen, Tammy Sachs, Gary McClain та інші.

Огляд сучасного стану проектування програмного забезпечення, нових концепцій інтерактивних взаємодій користувачів з програмними продуктами, нових методик дослідження поведінки користувачів, набуття ними досвіду використання все більш складних автоматизованих систем, показав відокремленість засобів інтегрованих середовищ розробки програмних продуктів від розроблених при дослідженнях методик, моделей, принципів, методів, взірцевих схем інтелектуалізації синтезу екранних інтерфейсів користувачів програмного забезпечення. Переважна частина етапів розробки програмних інтерфейсів користувачів лишається слабо автоматизованою і покладена на досвід та інтуїцію проектувальника. У середовища програмування не включені численні моделі користувачів.

Актуальність теми. Актуальність роботи зумовлена необхідністю існування структурованої пошуково-інформаційного, програмно-методичного комплексу (репозиторію) інструментальних засобів для автоматизації процесу підбору (синтезу) відповідних компонентів інтерфейсу користувача (ІК) не лише за загальним графічним виглядом, а й програмним кодом, функціональними характеристиками, параметрами візуалізації, методами обробки дій користувача з підказками для вибору альтернатив з множин елементів інтерфейсу користувача, інтелектуалізацією рекомендацій для зваженого позиціонування компонентів інтерфейсу на полі макетування. Також актуальність роботи зумовлена загальними тенденціями розробників програмних продуктів до пошуку нових засобів організації централізованих інформаційних сховищ даних для патернів, метрик, досвіду проектування інтерфейсної взаємодії. Зростання кількості та варіантів елементів інтерфейсу користувача з багатоваріантною багатомовною програмною реалізацією потребує створення засобу для класифікації, систематизації та ефективного пошукового механізму. Також серед чинників, які спонукають до досліджень, є: експоненціальне зростання кількості програмного забезпечення (ПЗ) прототипування інтерфейсів із подібними функціями без

якісного зростання нових засобів оперування характеристиками компонентів ПК на ранніх етапах планування та розроблення; еволюційний, ітераційний розвиток способів та метафор інтерфейсу ПЗ і необхідності пошуку інструментів проектування інтерфейсів під ці нові підходи; необхідність збереження ефективності взаємодії в діалоговому режимі проектувальників та програмістів із інструментами прототипування; необхідність динамічної адаптації проектувальника під нові, ще більш інтенсивні, терміни виконання завдань; необхідність впровадження антропоцентрованості проектування інтерфейсних полів ПЗ. Отже тема досліджень є актуальною.

Зв'язок теми дисертації з державними програмами, науковими напрямами університету та кафедри. Тема дисертаційної роботи відповідає науковому напрямку кафедри «Системи автоматизованого проектування» Національного університету «Львівська політехніка»: «Розроблення та впровадження комп'ютерних систем автоматизованого проектування організаційно-технічних систем». Об'єкт, предмет та методи досліджень повністю відповідають напрямкам наукової роботи, що проводяться на кафедрі. Робота виконана в межах науково-дослідних робіт: «Розробка засобів моделювання критичних теплових режимів МЕР на основі прогресивних технологій проектування» (01.01.2003-31.12.2004 рр., № держ. реєстр. 0104U002326); «Розроблення складних інформаційних систем для колективного розподіленого проектування інженерно-технічних та еколого-економічних об'єктів» (01.06.2007-31.12.2009 рр., № держ. реєстр. 0107U006231);.

Мета і завдання дослідження. Метою роботи є розроблення на основі системного підходу програмно-методичного комплексу інструментальних засобів ефективної автоматизації синтезу екранних прототипів інтерфейсів користувачів програмних продуктів для забезпечення процесів діалогових взаємодій розробників шляхом пошуку, модифікації, інтеграції та створення гнучких варіантних діалогових послідовностей, кодів програмних реалізацій

інтерфейсів користувача на базі систематизованого репозиторія формалізованих принципів, методик, методів проектування інтерфейсів користувача, моделей поведінки користувача та патернів програмних реалізацій. Для досягнення поставленої мети необхідно було вирішити такі задачі:

- аналіз сучасного стану методів, концепцій, метафор, процесів, засобів побудови діалогових взаємодій користувачів програмного забезпечення;
- систематизація, класифікація типів інтерфейсів користувача; систематизація та класифікація метрик ПЗ прототипування користувацьких інтерфейсів; систематизація засобів оцінювання, опису поведінки користувачів; систематизація програмування інтерфейсів користувача, патернів розробки інтерфейсів користувача; систематизація та класифікація користувацьких компонентів інтерфейсу;
- розроблення та удосконалення методу оцінювання впливу користувачів на систему «людина-техніка», методу накопичення досвіду інтерактивних взаємодій користувача, методу оцінювання засобів моніторингу подій інтерфейсів користувача, методу оцінювання засобів прототипування та вибору мови декларування інтерфейсів користувача;
- розроблення інформаційної архітектури засобів репозиторію патернів користувацького інтерфейсу;
- розроблення інформаційного, лінгвістичного, програмного забезпечення репозиторія інтелектуальних патернів компонентів, конструктивів інтерфейсу користувача;
- розроблення програмно-методичного комплексу інструментальні засобів дослідження та накопичення даних про взаємодії з інтерфейсами, засобів розпізнавання шаблонних елементів, які би забезпечили автоматизацію програмування прототипів інтерфейсної складової, засобів синтезу програмних кодів інтерфейсів користувача;

- апробація розроблених методів, інформаційних архітектур, алгоритмів, засобів на тестових даних.

Об'єктом дослідження є процес автоматизованого проектування графічного інтерфейса користувача програмних систем.

Предметом дослідження є математичні моделі поведінки користувача ПЗ, технології проектування, макетування інтерфейсного поля користувача ПЗ на базі систематизованих патернів досвіду використання і програмування інтерфейсу користувача.

Методи дослідження. У процесі розв'язування поставлених задач використано методи: системного аналізу та проектування, прямого спостереження поведінки (input device tracking, mouse tracking, eye-tracking); статистичні методи; теорія ймовірності; теорія векторів; теорія графів (інформаційна пропускна здатність оперування, траєкторії діалогів); теорія масового обслуговування; множинний підхід опису; моделі діяльності; методи експертних оцінок; багатокритеріальні оцінки; теорія згорток; колірні моделі; координатні системи та координатні представлення; геометричні моделі; методи розпізнавання контурів; методи та патерни об'єктно-орієнтованого проектування та програмування.

Наукова новизна одержаних результатів.

Вперше:

- на основі використання методів критичного аналізу, методів систематизації й класифікації, здійснено комплексну систематизацію діалогових взаємодій користувача програмного забезпечення, в якій, на відміну від існуючих, здійснено ієрархічний поділ з врахуванням типів користувачів, концепцій організації діалогу, форми, способи інтелектуалізації, кінематичні способи керування інтерфейсом, що дає змогу єдиним чином відобразити предметну область та внести принципи групування й структурування даних в інформаційний репозиторій патернів синтезу інтерфейсів користувача;

- на основі методів системного аналізу, декомпозиції та системного проектування, запропоновано та обґрунтовано інформаційну інтегровану архітектуру засобів синтезу (побудови) програмних інтерфейсів користувача, з врахуванням даних накопичення досвіду використання, патернів проектування інтерфейсів варіантних типів, патернів візуальних, функціональних, програмних кодів конструктивних елементів інтерфейсу користувача, що дозволило за рахунок автоматизації процесів на 5 % пришвидшити адаптацію користувача до нового ПЗ, та пришвидшити роботу програмістів-розробників інтерфейсів ПЗ до 30%;

Вдосконалено:

- на основі використання методів статистичного аналізу та експертних оцінок, удосконалено методи оцінювання якісних характеристик існуючих програмних засобів прототипування та програмування інтерфейсів користувачів, що дало змогу оцінювати характеристики існуючих спеціалізованих мов програмування та декларування описів програмних кодів інтерфейсів користувачів, а отже й зменшувати час на визначення загального критерію використання певного засобу чи технології до застосування у проектах побудов інтерфейсів користувача;
- на основі використання методів геометричного моделювання удосконалено моделі ієрархічного поділу поля макетування компонентів графічного інтерфейсу користувача, що з врахуванням аналізу інтегрованих даних з репозиторію про використання, аналізу зв'язаних та взаємозамінних компонентів інтерфейсу, дає змогу трансформації (реінженірінгу) програмних лексем опису інтерфейсу під варіантні пропорційні розміри екранного відображення.

Практичне значення одержаних результатів.

- Одержані результати у вигляді ієрархічних класифікацій інтерфейсних взаємодій; якісні та кількісні статистичні дані застосованих

конструктивів інтерфейсу користувача в САПР, кроки та алгоритми методів оцінювання засобів прототипування інтерфейсів користувачів, якісні оцінки мов декларування ІК, інформаційні засоби та програмні засоби, які можуть використовуватися у подальшому розробниками для удосконалення існуючих засобів проектування чи прототипування інтерфейсів; спеціалістами з концептуального проектування інтерфейсних взаємодій користувача з ПЗ та у навчальному процесі.

- Одержані у дисертаційній роботі результати можуть бути застосовані, на прикладі ІТ індустрії України, для спеціалістів: дизайнер інтерфейсів; проектувальник взаємодій; спеціаліст, дослідник, експерт, тестувальник зручності використання; інформаційний архітектор; програміст користувацьких інтерфейсів. Натомість на прикладі вакансій ІТ індустрії світу: GUI Developer; Human (User) Experience Designer; Human Factors Engineer; Interface Designer; Usability Researcher Engineer; User Experience Strategist; Visual User Interaction Designer.
- Результати досліджень впроваджено у навчальний процес на кафедрі «Системи автоматизованого проектування», Національного університету «Львівська політехніка» у процесі викладання дисципліни «Методи побудови інтелектуального інтерфейсу користувача в автоматизованому проектуванні» для студентів спеціальності 7.05010103 / 8.05010103 «Системне проектування» (спеціалісти та магістри). Також результати досліджень впроваджені (використовуються) у виробничій діяльності програмістів і тестерів програмного забезпечення в ТзОВ «Едвантіс» (м. Львів, Україна), що підтверджено відповідними актами.

Особистий внесок здобувача. Усі наукові результати теоретичних та практичних досліджень, викладені в дисертаційній роботі, одержані автором особисто. У друкованих працях опублікованих у співавторстві, дисертанту належать:

у [10, 11] – критичний систематизований аналіз існуючих підходів розробки діалогових систем та оцінка придатності їхнього застосування в інтерфейсах користувача графічного редактора схем, реалізація редактора схем;

у [45] – визначення критеріїв до систематизації кінетичних типів інтерфейсів користувача, формування ієрархічної класифікації виділених об'єктів;

у [54] – ідея та спосіб специфічного застосування інтегрованого середовища програмування, зокрема, як редактора форм інтерфейсу користувача та, в якості препроцесорного засобу автоматизації проектування площинного компоновання корпусів мікросхем із подальшим скриптовим експортуванням у САПР SolidWorks для перевірки функціональних можливостей інтеграції API двох систем, опрацювання результатів тестових експериментів;

у [99] [100] – визначення критеріїв оцінювання, формалізація моделі порівняння мов декларування інтерфейсів користувачів, результати статистичного аналізу;

у [101] – визначення критеріїв впливу на людино-технічні системи, отримання інтегрованої оцінки тренерованості користувача на сценарій взаємодії;

у [107, 108] – формалізація моделювання досвіду користувач діалогових систем, узагальнення результатів «температурних карт» густини взаємодій»

у [115] – визначення критеріїв оцінювання, формалізація методу оцінювання, підбір досліджуваних засобів моніторингу подій ІК;

у [119] – ідея та алгоритмізація процесу автоматизації перетворення ІК з графічної форми в програмний код;

у [120] – опис етапів сегментації зображення на конструктивні елементи ГІК та проведення експериментальних випробовувань синтезу інтерфейсу;

у [124] – структура процесу автоматизованого перепрограмування інтерфейсу користувача з використанням репозиторію конструктивів ГІК та архітектури транслятора лексем кодів мов програмування;

у [125] – розробка та реалізація інформаційного забезпечення репозиторію лексем для реінженірінгу програмного коду рівня інтерфейсів користувача;

у [126] – статистичний аналіз конструктивів ГІК в середовищах розробки ПЗ, розробка та реалізація інформаційного забезпечення та програмного забезпечення репозиторію візуальних і програмних характеристик компонентів ГІК.

у [127] – проведення детального статистичного аналізу існуючих САПР машинобудування на реалізований в них спектр компонентів ГІК, систематизація та візуалізація результатів статистичних обчислень;

у [144] – розроблення та реалізація архітектури та інформаційного, лінгвістичного, програмного забезпечення засобу накопичення подій ІК;

Апробація результатів дисертації. Основні наукові, теоретичні положення та практичні результати дисертаційної роботи доповідалися та обговорювалися на міжнародних, всеукраїнських науково-технічних конференціях, науково-технічних семінарах, зокрема:

- науково-технічних семінарах кафедри «Системи автоматизованого проектування», Національного університету «Львівська політехніка» (2001-2002, 2011-2016 рр.);
- Міжнародній конференції «Сучасні проблеми радіоелектроніки, телекомунікацій, комп'ютерної інженерії», TCSET (м. Львів, с.м.т. Славсько, Україна, 2002, 2010 рр.);

- Міжнародній конф. «Computer Science and Engineering», CSE (м. Львів, Україна, 2010 р.);
- Міжнародній конференції «Computer Sciences and Information Technologies», CSIT (м. Львів, Україна, 2010, 2012-2014 рр.);
- Міжнародній конференції «The experience of designing and application of CAD systems in microelectronics», CADSM (с.м.т. Славсько, Україна, 2001, 2011, 2013 рр.);
- Міжнародні конференції «Perspective Technologies and Methods in MEMS Design», MEMSTECH (м. Львів-Поляна, Україна, 2010 р.);
- Україно-польській конференції «CAD in Machinery Design. Implementation and Education Problems», CADMD (м. Львів, Україна, 2010, 2012, 2014 рр.);
- Всеукр. школі-семінарі молодих вчен. і студ. «Сучасні комп'ютерні інформаційні технології», АСІТ (м. Тернопіль, Україна, 2011, 2014, 2016 рр.).

Публікації. За результатами досліджень, викладених у дисертації, опубліковано 28 наукових праць, з них:

- 7 статей опубліковано у фахових наукових виданнях України (з них: 4 одноосібно);
- 1 стаття [160] у науковому періодичному виданні іноземної держави;
- 20 публікацій в матеріалах та тезах науково-технічних конференцій, з яких: 15 міжнародні, 5 всеукраїнські (з них: 3 одноосібних публікації);
- 1 конспект лекцій.

Серед цих наукових праць 8 публікацій у виданнях, індексованих у міжнародних базах даних, а саме з них: 3 в Scopus [99, 107, 126] (індекс автора: 8393585000); 5 в IEEE [12, 13, 99, 107, 126] (індекси автора: 37284843500 та 38014474900); 8 в Google Scholar [12, 13, 99, 107, 108, 143, 145] (індекс автора: EWSnvSoAAAAJ); 5 в Research Gate [12, 13, 99, 107,

126]; 5 в Microsoft Academic [12, 13, 99, 107, 126] (індекси автора: 56148571; 50366576 та 49392523); 3 в ORCID [99, 107, 126] (індекс автора: 0000-0002-2432-0768)

Структура та обсяг дисертації. Дисертаційна робота складається із переліку умовних скорочень, списку рисунків, списку таблиць, вступу, чотирьох розділів, висновків, списку використаних джерел із 160 найменувань на 18 сторінках та 2-х додатків. Загальний обсяг дисертаційної роботи складає: 186 сторінок, з них 154 сторінки основного тексту, 90 рисунків, 12 таблиць.

Розділ 1. Сучасний стан побудови інтерфейсів користувачів ПЗ

У першому підрозділі розглянуті основні світові тенденції провідних компаній виробників ПЗ в напрямку дослідження взаємодій користувачі з інтерфейсами ПЗ, розглянуті визначення у межах проектування інтерфейсів та патернів проектування, проаналізовані діалогові процеси, представлений реферативний огляд існуючих концепцій, метафор побудови ІК, запропонована ієрархічна систематизація користувацьких взаємодій та кінетично-орієнтованих інтерфейсів, розглянуті проектні процеси на етапах планування, прототипування, макетування і побудови користувацьких інтерфейсів, здійснений порівняльний аналіз інформаційних засобів прототипування, розробки ІК та порівняльний аналіз мов декларування інтерфейсу користувача в ПЗ.

Основні результати розділу опубліковані у працях [10, 11, 19, 20, 21, 45, 99, 100]

1.1. Світові тенденції

За останні роки спостерігається загальна тенденція великих корпорацій з розробки програмного забезпечення, таких як Google, Microsoft, Adobe, Apple, АВВУУ, Ahead, Eset та інших, до включення до складу своїх програмних продуктів засобів інтегрованої методики дослідження кінцевих користувачів, їхньої поведінки при взаємодії з ПЗ та уподобаннями у користуванні. Як правило, під час інсталювання програмного продукту користувачу пропонується добровільно погодитися з тим, що програма буде надсилати розробникам (мережею Інтернет) анонімну інформацію про власні налаштування, тенденції використання та іншу статистичну інформацію для удосконалення програмного забезпечення. На рис. 1.1 представлені деякі з

таких запитів на анонімну співпрацю. З огляду програмних продуктів у галузі САПР, а також дотичних програм наукового спрямування, не дав результатів про те, чи розробники ПЗ впроваджують автоматизовані засоби дослідження досвіду користувачів власних програмних продуктів з врахуванням специфіки галузей САПР.

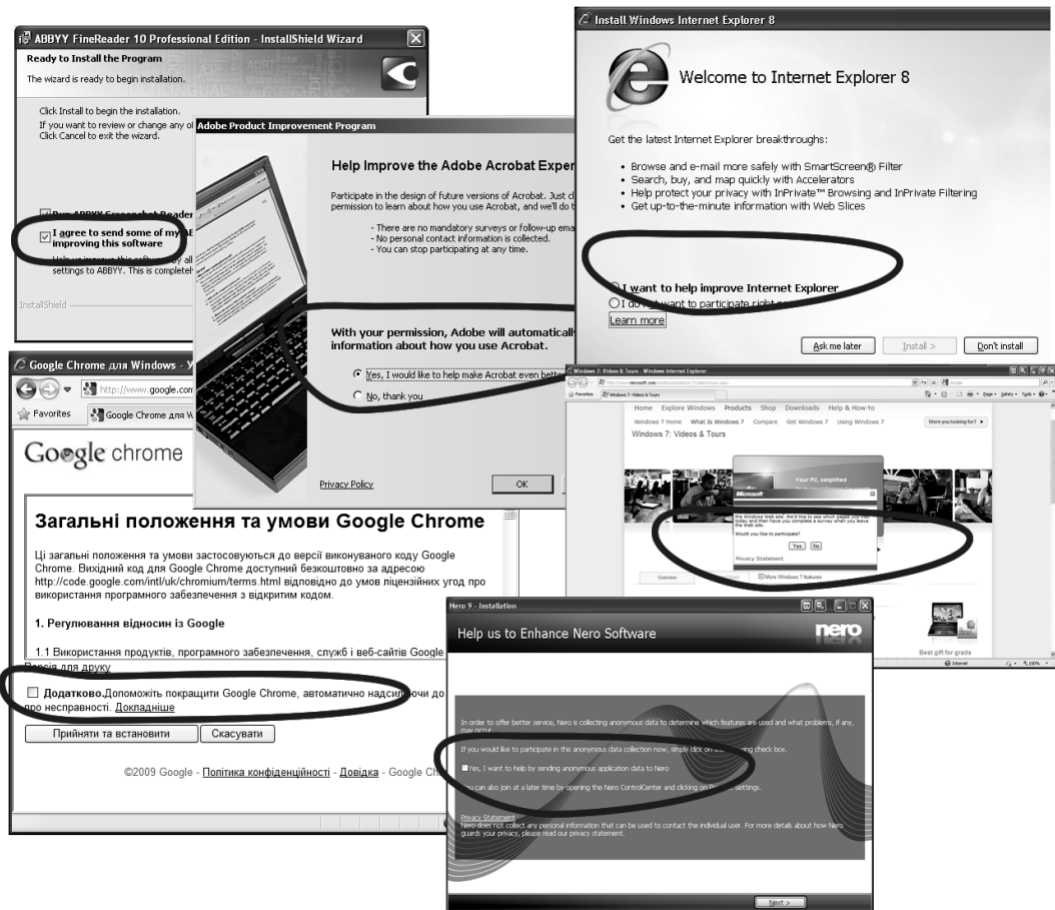


Рис. 1.1. Запити на дозвіл збирання даних про дії користувачів ПЗ АБВУУ, Adobe, Google, Microsoft, Ahead

Отже для проведення дослідження «взаємодії комп'ютер-користувач» та «взаємодії користувач-комп'ютер» необхідно:

- оцінити досвід використання користувачем інтерфейсів програмного середовища САПР;

- оцінити психологічну, ергономічну, соціальну, організаційну поведінку проектувальника-користувача з ПЗ САПР;
- оцінити факт ознайомчої, навчальної та впевненої трудової поведінки користувача-проектувальника з інтерфейсами ПЗ САПР.

Як правило, тестування зручності користування (англ. термін usability, юзабіліті) є трудомісткий [102].

1.2. Основні поняття інтерфейсів користувача

Інтерфейс користувача, деколи ще називають користувацький інтерфейс (англ. User interface, UI) – є різновидом інтерфейсів, коли з одного боку є людина-користувач, іншого боку є технічний пристрій чи програмний продукт. При цьому в сукупності розглядаються засоби й методи для організації взаємодії й обміном інформацією під керуванням людиною.

Людино-машинна (комп'ютерна) взаємодія – термін, що поєднує знання із таких галузей, як психологія пізнання, соціологія, ергономіка, проектування ПЗ й комп'ютерних систем, системний аналіз, засоби управління і промисловий дизайн. Проектант-програміст у галузі комп'ютерних технологій повинен знати й вміти розробляти чи адаптувати користувацький інтерфейс під широкий клас користувачів; досягти ефективного використання програмних додатків.

Взаємодія користувача, досвід користувача (англ. User experience, UX) – це те, як людина відчувається при використанні продукту, системи або послуги. Взаємодія користувача свідчить про досвід, ефективні, значимі й цінні аспекти взаємодії людини з комп'ютером та продуктів, але також містить уявлення людини про практичні аспекти, такі як функціональність, простота використання та ефективність системи. Взаємодія користувача має суб'єктивний характер, оскільки мова йде про почуття людини і думки про систему. Досвід користувача є динамічним, адже він змінюється з часом у зв'язку з обставинами, що змінилися.

Проектування досвіду (взаємодії) користувача (англ. User experience design, UXD) є підмножиною області досвіду проектування, що стосується створення архітектури та моделей взаємодії, які впливають на зручність роботи з комп'ютером. Сферу досліджень скеровано на зміну всіх аспектів взаємодії користувача з продуктом: як він сприймає, навчається і використовує.

Людино-машинний інтерфейс (англ. Human machine interface, HMI) – це розширений термін, який відображає інженерні рішення, щоб забезпечити керування пристроями діями оператора. Проектування таких систем інтерфейсу можна асоціювати із ергономікою, в водночас ці поняття не є тотожні. Такий інтерфейс акцентує увагу на організації місця оператора: робочого столу, пульта управління, розміщення та склад елементів керування тощо. Особливу увагу приділяють діям оператора з функціональними пристроями керування, їхнім рівнем доступності й прикладанням зусилля, узгоджування чи несуперечення дій керування, «захист від дурня», тип і розміри написів, розташування вікон відображення та інше. Для реалізації інтерфейсу застосовують панелі операторів, панелі комп'ютеризованих пристроїв, типових бібліотек конструктивів ГІК програмного забезпечення. У межах цього головним завданням проектування інтерфейсу систем стає полегшення трудомісткістю користувача у спосіб відображення на екранах, вікнах комп'ютерів інтуїтивно зрозумілої інформації.

Автоматизоване робоче місце (АРМ) – це комплекс технічних і програмних засобів для індивідуальної автоматизації роботи професіонала-спеціаліста, у якому можна забезпечити початкову підготовку, подальше редагування, розширений пошук і представлення на екран чи видрукування проектних, дослідницьких документів чи даних.

Шаблон проектування (англ. Design pattern) ще скорочено називають **патерном** підчас розроблення програмного забезпечення – це повторювання типової конструкції, конфігурації інформаційної архітектури, при цьому це

стає рішенням завдання проектування для певного часто повторюваного контексту, для певних меж даних. Такий шаблон (патерн) ще не є фінальним зразком реалізації, і його ще неможливо напряму перетворений у програмний код; патерн дає лише типовий приклад розв'язання завдання, і який використовують у різноманітних ситуаціях. Найпоширенішими є об'єктно-орієнтовані патерни, які відображають відношення і взаємодії між класами чи об'єктами, без визначення кінцевих класів або об'єктів. Для врахування специфічних особливостей конкретних мов програмування чи декларування застосовують «низькорівневі» патерни.

1.3. Діалогові процеси інтерфейсних взаємодій

Об'єктом системного аналізу є діалогова система (ДС) [10, 16, 17, 18] як сукупність технічних, програмних та інформаційних засобів, що надає певному колу користувачів можливістю розв'язувати деяку множину задач в інтерактивному режимі взаємодії з комп'ютером відповідно до сценарію діалогу.

Системному аналізу підлягають такі сторони об'єкту ДС, як:

- класифікація за різними ознаками (типом користувачів, проблемною орієнтацією, методами організації діалогу та програмного забезпечення тощо);
- систематизація та класифікація типів, форм та режимів діалогу;
- аналіз метафор, які покладено в основу діалогу, аналіз позитивних якостей інтерактивної графіки і їх вплив на ефективність роботи користувача. [17]

При організації діалогу розвивається тенденція інтелектуалізації інтерфейсу [10, 21] користувач-комп'ютер. «Дружність» інтерфейсу забезпечується імітацією звичної для користувача роботи з об'єктом, заснованій на максимальному використанні технічних можливостей

персональних комп'ютерів. При цьому старі методи роботи, що реалізуються новими засобами, набувають метафоричного характеру [11].

При інтелектуалізації спілкування проектувальників САПР можуть використовуватися наступні прийоми та принципи:

а) метафора письмового столу – імітація на екрані поверхні письмового столу з розміщеними на ньому документами (при цьому застосовується так зване багатовіконне відображення інформації: екран розбивається на прямокутні чи іншої геометричної форми вікна, які можуть перекриватися, у кожному вікні відтворюється фізичний образ документа чи його частини і засоби керування курсором, миша та клавіатура, дозволяють знаходити, переглядати, редагувати, переміщувати чи знищувати документи на екрані) [14] ;

б) метафора великоформатного бланка – розрахунок і представлення результатів у вигляді двомірної електронної таблиці (кожна комірка таблиці має певне значення і може бути зв'язана з розрахунковою формулою, і при внесенні змін у будь-якій комірці таблиці проводиться автоматичний перерахунок усіх залежних від неї величин в інших комірках) [14] ;

в) принцип безпосереднього спілкування – заснований на максимальному відштовхуванні від мовного спілкування (директивного й навіть природно-мовного) на користь безпосередньої безмовної взаємодії користувачів із системою (робота користувача за дисплеєм носить маніпулятивний характер, дивитися та діяти, а не запам'ятовувати та набирати текст; для виконання маніпуляцій використовуються як традиційні, так і нові засоби керування рухом курсору; перевагами такої організації взаємодії є використання звичної моторики роботи за столом і залучення образно-геометричного сприйняття людини натомість абстрактно-логічному [16];

г) метафора редагування – єдиний стиль спілкування користувача при роботі з представленням об'єктів (для підтримки відповідності між об'єктами

та їхніми зовнішніми представленням пропонується стандартний набір операцій редагування: переміщення об'єкта вздовж вікна; редагування атрибутів об'єкта; видалення об'єкта; рух у середину – розкриття вмісту об'єкта; рух назовні – визначення місця об'єкта у світі об'єктів; виконання дій; вибір активного вікна; стандартні операції редагування викликаються при натисненні певних функціональних клавіш а крім стандартних операцій, можуть виконуватися нестандартні дії, що залежать від конкретної програми) [16];

д) інтеграція програмних засобів – забезпечує комплексну інтелектуалізацію діяльності проектувальників (необхідні засоби об'єднуються в так звані системи керування проектом чи інтегровані програмні системи й дозволяють одночасно підтримувати керування базами даних, комунікаційні можливості, засоби ділової графіки та інтелектуальний діалоговий інтерфейс; основні задачі при створенні інтегрованих систем – забезпечення інформаційного зв'язку між програмами, базами даних та екраном; перемикання активності прикладних програм; підтримка мінімального набору операцій, що надані користувачу для роботи з різними програмами; процес інтеграції функцій впливає на характер зв'язку прикладних програм та операційної системи; багато функцій операційної системи переносяться на прикладний рівень, збагачуючий і розвиваючи можливості прикладних програм [16]).

1.4. Різновиди концепцій інтерфейсів користувачів

1.4.1. Інтерфейс командного рядка (CLI) і текстовий (TUI)

Інтерфейс командного рядка (англ. Command-line interface, CLI) – це різновид інтерфейсу користувача символічно-текстового типу, в якому інструкції дій для ПЗ комп'ютера здійснюються лише введенням з клавіатури текстових рядків, що містять наперед визначені команди. Друга назва цього

типу інтерфейсу – «консоль». У інтерфейсі командного рядочка формат інформації зазвичай регламентований. У різній літературі ще застосовують термін текстовий інтерфейс користувача (англ. Text user interface, TUI).

Короткі характеристики:

1. У пам'яті програмного забезпечення займає мало пам'яті, до порівняння із системою графічних меню.

2. У програмних продуктах із великою функціональністю, до прикладу САПР, значна частина команд використовується нечасто але внесена розробниками. У такому випадку доступ до команд здійснюється через командні стрічки. У деяких випадках набирати команди чи серію команд є швидше за навігацію у панелях меню графічного інтерфейсу.

3. Ще одне застосування – створення пакетного текстового інтерфейсу. У цьому випадку у файл форматовано записуються послідовно командні дії з параметрами, потім файл відкривають у програмі, послідовне автоматичне виконання дій з файлу еквівалентні якби ці команди по чергово вводили окремо в командний рядок (приклади: bat-файли в Windows, shell-скрипти в Unix-системах).

Переваги:

1. Для активації команди достатньо невеликої кількості натискань клавіш.

2. Пакетні файли – нагадує програмуваність кроків.

3. Необхідно менше ресурсів для віддаленого керування, без потреби графічного інтерфейсу.

4. Послідовне введення команд у консолі відображає хронологію дій користувача та відповідей програмного засобу.

Недоліки:

1. Для непідготовленого користувача інтерфейс командного рядка потребує більше часу на вивчення.

2. Виникає необхідність у створенні довідників з описами команд.

3. Є складно вводити форматовані параметри з клавіатури.

На рис. 1.2 показано приклади інтерфейсів. Перший демонструє можливості програмної бібліотеки Turbo Vision UI Framework (1992 рік), коли елементи користувацького інтерфейсу організуються псевдографікою із текстових символів.

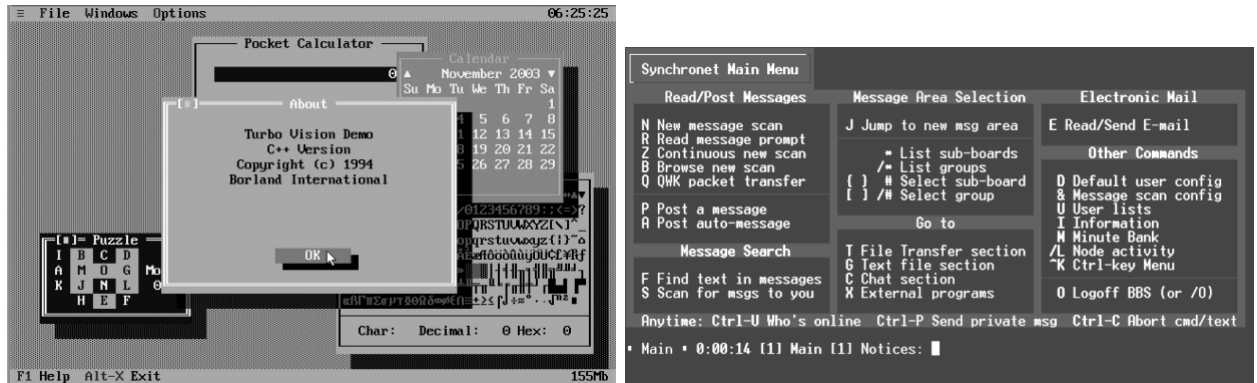


Рис. 1.2. Приклади текстового інтерфейсу користувача ПЗ

1.4.2. Графічний інтерфейсу користувача (GUI)

Графічний інтерфейс користувача (ГІК, англ. GUI, Graphical user interface) – це такий інтерфейс між користувачем та технічним пристроєм, здебільшого програмним продуктом, у якому за допомогою піктограм, ієрархічних меню, візуальних кнопок і перемикачів керування, курсорами задається функціональність для виконання команд. Переважна більшість програмного забезпечення застосовує багатовіконні режими для розміщення необхідних конструктивів керування функціями ПЗ. При цьому, до порівняння із командним рядком, користувач у власний довільний спосіб чи клавіатурою, чи пристроєм екранного позиціонування («миша», сенсорний дисплей) керує всіма видимими екранними графічними конструктивами ГІК (приклади на рис. 1.3). Уперше концепція ГІК була представлена дослідницькою лабораторією компанії Херох в 1970-х, у подальшому комерційно реалізовано у програмних продуктах корпорації Apple.

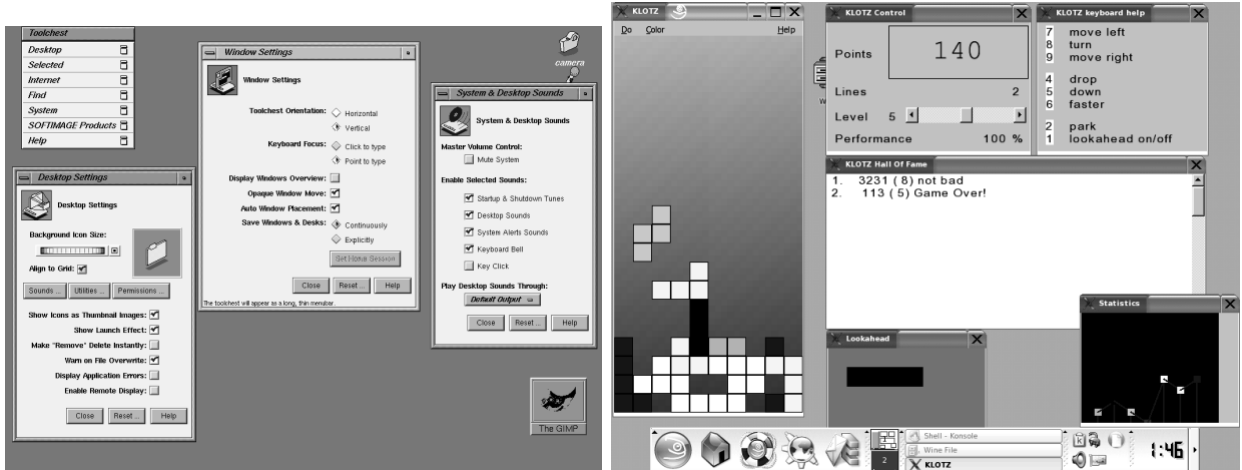


Рис. 1.3. Приклади графічного інтерфейсу користувача ПЗ

1.4.3. Інтерфейс імітації кінетичних взаємодій цифрових об'єктів

Користувацький інтерфейс з імітацією фізики кінетичних взаємодій (англ. Physics imitation user interface, PIUI) – спосіб організації на екрані візуальних об'єктів та елементів користувацького інтерфейсу, які мають візуальну графічну поведінку як фізичні об'ємні фігури реального світу, і при якому діють закони фізики, як: гравітація, геометричні розміри, глибина, пружне зіткнення один з другим, сила тяжіння тощо. Прикладом реалізації є VumpTop 3D Desktop [22], віконні інтерфейси якого наведені на рис. 1.4.

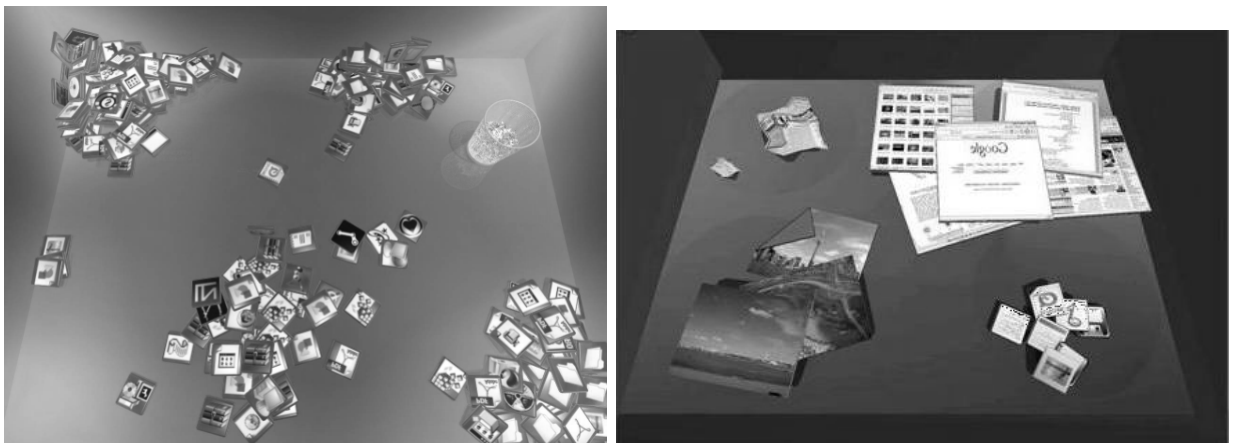


Рис. 1.4. Приклади інтерфейсу користувача з імітацією фізики

1.4.4. Інтерфейс доповненої реальності (Augmented reality UI)

Яскравими прикладами [23, 24] інтерфейсу за типом доповненої реальності (англ. Augmented reality interface) є графічне проєціювання на фізичні поверхні інтер'єру елементів програмного користувацького інтерфейсу з візуальним чи сенсорним розпізнаванням взаємодії. На рис. 1.5 показано приклади прототипів проєкцій доповненої реальності для light-touch [23] та SixthSense [24].

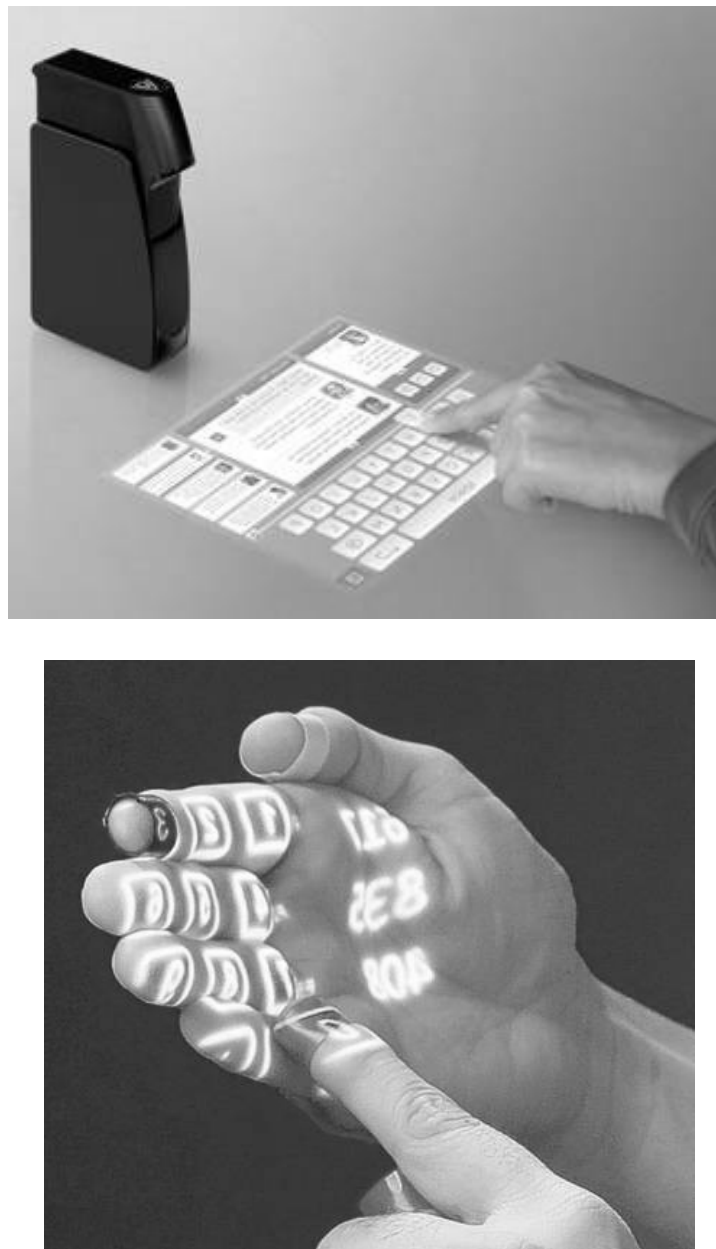


Рис. 1.5. Приклади інтерфейсу користувача доповненої проєкції

1.4.5. Матеріальний інтерфейс користувача (Tangible UI)

Матеріальний інтерфейс користувача (англ. Tangible user interface) – це різновид інтерфейсу, у якому взаємодія з цифровою інформацією на екрані здійснюється за допомогою матеріальних конструкцій. «Матеріальні біти» – спроба надати цифровій інформації фізичне обличчя. Переміщення частини фізичного об'єкта призводить до дії над еквівалентною частиною цифрової інформації. Маніпулювання з фізичними формами спрощує доступ до цифрової інформації. Ґрунтовні дослідження ведуться [25, 26, 27]. Елементами користувацького інтерфейсу є різноманітні фігури з графічними мнемоніками, які під час автоматичного оцифрування перетворюють свої закодовані дані на цифрову форму.

Особливості матеріального користувацького інтерфейсу (англ. Material / Tangible User Interface) [26]:

- фізичні блоки-фігурки з маркерами та Multi-Touch дисплей роблять його прямим і природньо контрольованим;
- усі інтерактивні процеси візуалізуються в реальному часі;
- спільність – кілька людей можуть маніпулювати даними одночасно;
- модульність – різні фігурки-модулі для визначених функцій користувацького інтерфейсу;
- гнучкість – високий рівень налаштовуваності. Розширювані візуальні ефекти і бібліотеки користувацького інтерфейсу;
- розпізнавання об'єктів [26, 27] – користувачі можуть розміщувати фізичні об'єкти на екрані для запуску різних типів цифрових відповіді, забезпечуючи безліч додатків і передавання цифрового контенту на мобільні пристрої.

Приклади керування цифровими даними музичного інструменту за допомогою фізичних предметів показано на рис. 1.6, а [26, 27]; розпізнавання предметів (фотокамер, телефонів) на Microsoft Surface (рис. 1.6, б) [27].



Рис. 1.6. Приклади інтерфейсу користувача з «матеріальними-бітами»

1.4.6. Жестовий інтерфейс користувача (Gestures UI)

Жестикулярний (жестовий, англ. Gestures) інтерфейс користувача – це спосіб керування програмою за допомогою: набору рухів (англ. Motion) мишею (рис. 1.7) [28]; набору дотиків пальців до поверхні [29] за певною траєкторією (рис. 1.8); візуального розпізнавання набору рухів частин тіла користувача (рук, ніг, тулуба); розпізнавання положення спеціалізованими маніпуляторами, приставок кінематики (як, наприклад, Nintendo Wii [30]) на поверхні або у просторі. Ідея полягає у заміні команд меню на ввід команд за допомогою умовних знаків, які жестикулярно малюються рухами маніпулятора. Одна з ідей концепції керування рухами тіла представлена як Google Gmail Motion [31] – керування поштовою системою Gmail (рис. 1.9). Наразі цей засіб не реалізований Google і описаний як жарт (01.04.2011 р.). Хоча інші розробники підхопили цю ідею і використовуючи Microsoft Xbox 360 Kinect [32] – апаратний датчик для роботи з додатками для Windows, – перетворили концептуальну ідею на реальну робочу версію продукту [33].

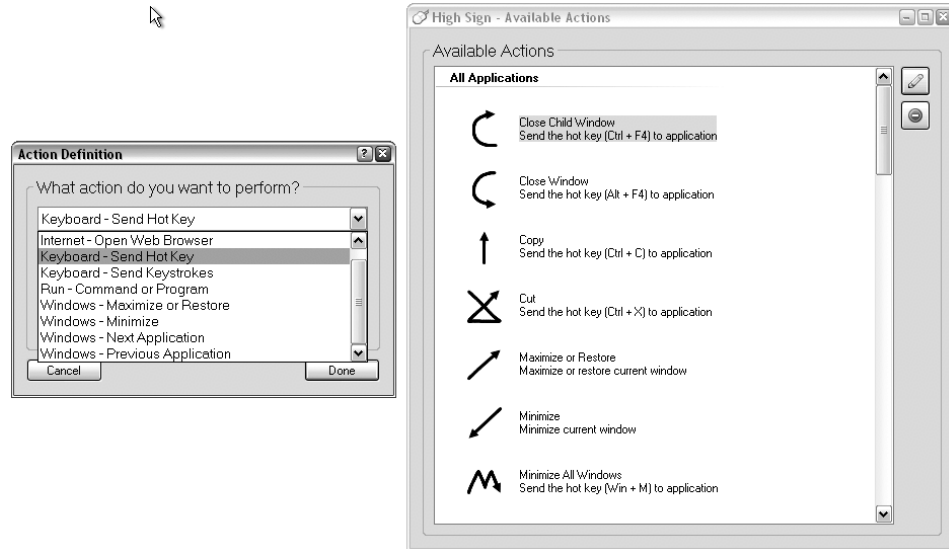


Рис. 1.7. Приклади жестового інтерфейсу користувача для «миші»

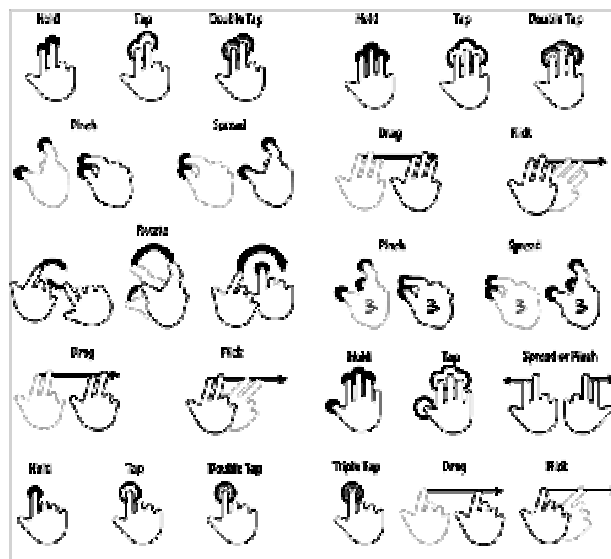


Рис. 1.8. Приклади жестового інтерфейсу користувача пальцями рук

Жест пристрою-вказівника або жест «миші» є способом об'єднання руху маніпулятора і клацання, які програмне забезпечення визнає як конкретні команди. Жести пристроями вводу можуть забезпечити швидкий доступ до часто використовуваних функцій програми.

Рухові жести – розпізнавання жестів рук, тулуба людини і трансформація у команди інтерфейсу ПЗ. Як ідея для прототипу керування частинами тулуба тіла описано в [31].

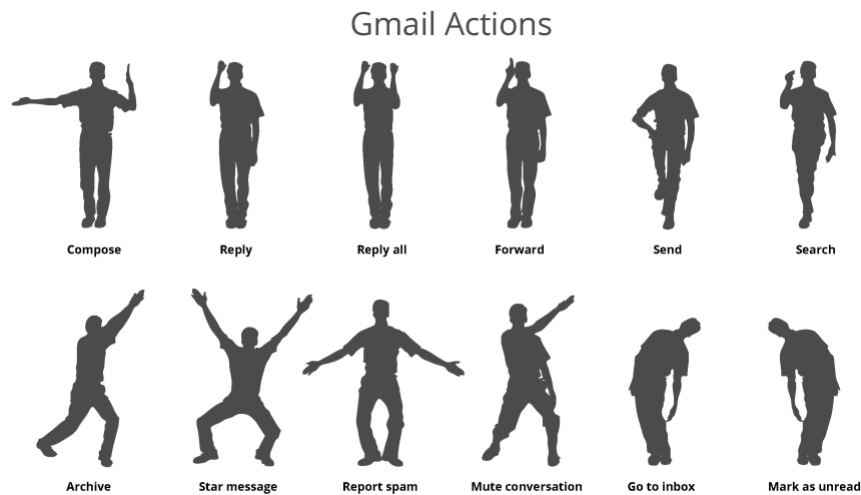


Рис. 1.9. Ідея прототипу керування програмами жестами частин тіла

1.4.7. Розчерковий інтерфейс (Calligraphic interface)

Розчерковий (рукописний) інтерфейс (англ. Calligraphic interface, Handwrite interface) аналогічний до жестикулярного, але зображення промальовується на сенсорних графічних панелях чи графічних планшетах з подальшим розпізнаванням написаного чи намальованих геометричних фігур. На рис. 1.10 показані приклади Samsung Galaxy Note 3 [34], Microsoft Surface Pro 3 [35].



Рис. 1.10. Приклади пристроїв із розчерковим інтерфейсом

1.4.8. Мімічний користувацький інтерфейс (Mimic UI)

Мімічний користувацький інтерфейс (англ. Mimic user interface) – подібний до жестикулярного інтерфейсу, але керування функціями програм здійснюється після обробки зображень станів мімічних гримас на обличчі. Одна з ідей для прототипу керування Інтернет браузером належить Opera Labs [36] наведена на рис. 1.11. Ця концепція сьогодні не є реалізованою .



Рис. 1.11. Приклади ідеї прототипу керування мімікою

1.4.9. Масштабований інтерфейс користувача (Zooming UI)

Масштабований інтерфейс користувача (англ. Zooming user interface, ZUI [37, 38]) – графічний інтерфейс користувача, де робочий простір є великою необмеженою площиною, на якій розміщено елементи. Їхні властивості та вміст стають доступними при наближенні шляхом збільшення (як мікроскоп). Наближення робить вміст доступним на глибших рівнях. Яскравим прикладом (показано на рис. 1.12) реалізації є картографічна онлайн система Google Maps [39].

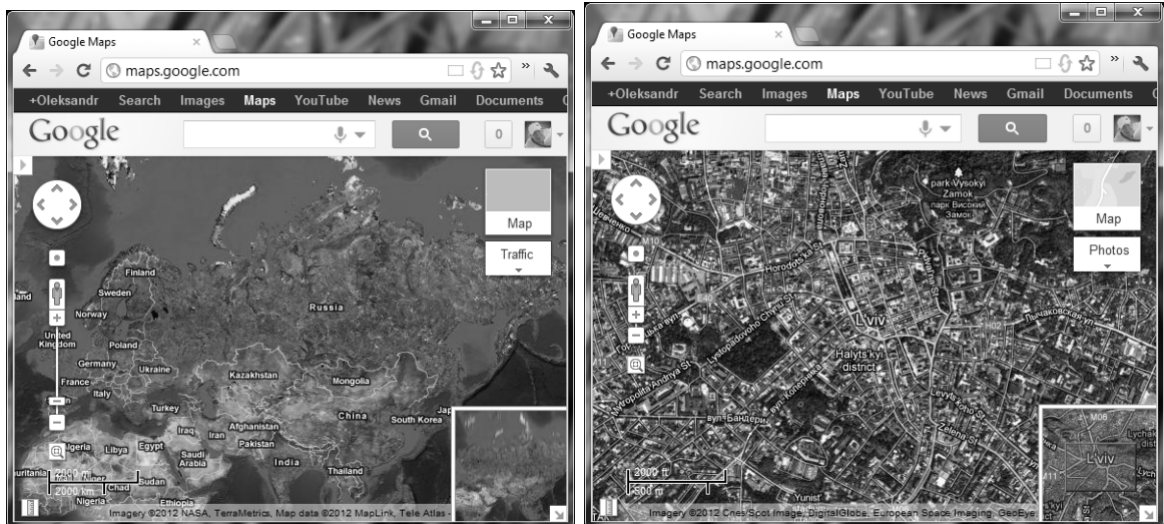


Рис. 1.12. Приклад масштабованого інтерфейсу

1.4.10. Фізично трансформований інтерфейс користувача

При фізично трансформованому інтерфейсі інформаційні дані та елементи керування інтерфейсами ПЗ автоматично з'являються і змінюють функції при зміні зовнішньої форми конструкції пристрою. Прикладом [40] прототипу такого інтерфейсу може бути концепт, який показано серією послідовних перетворень (рис. 1.13), коли форма корпусу мобільного пристрою змінюється від «планшету» до «телефону», як «розкладачка».

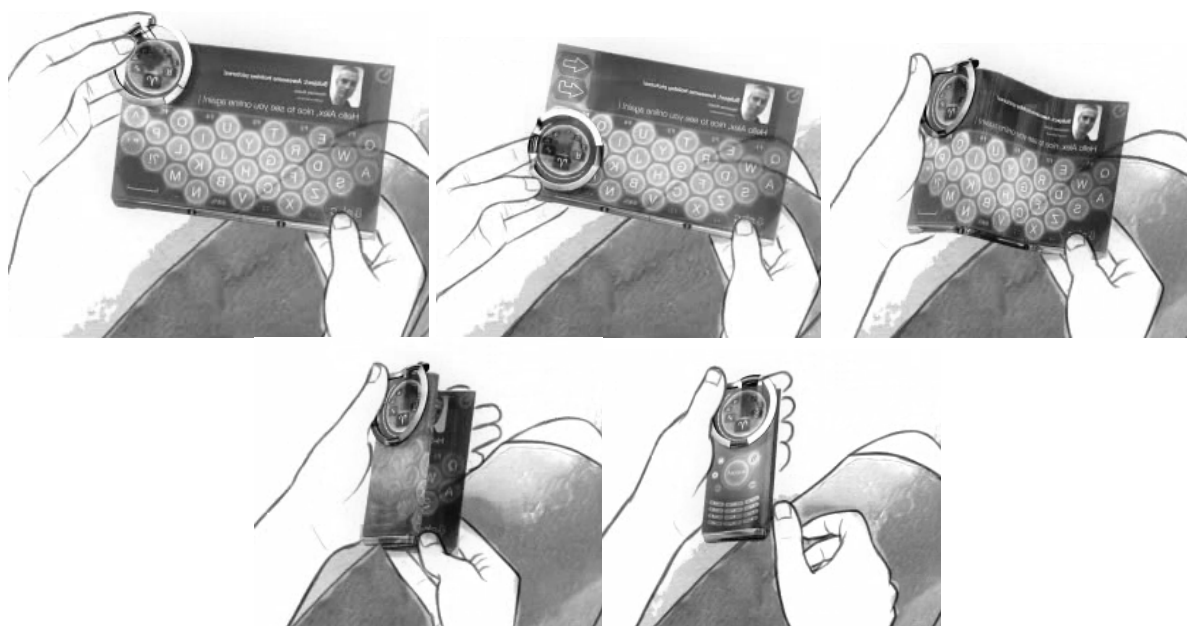


Рис. 1.13. Приклад концепції фізично трансформованого інтерфейсу

1.4.11. Соціалізований інтерфейс користувача

Соціалізований користувацький інтерфейс [41] – на екрані наявні знаки, які відповідають правилам інтерактивності та ввічливості, поширені серед людей. Соціалізовані користувацькі інтерфейси містять програмні інтелектуалізовані агенти (помічники). Задача соціалізованих програмних користувацьких інтерфейсів – приховати внутрішню складність структури інформаційної системи від кінцевого користувача. Сутність проектування такого інтерфейсу – створення у користувача враження спілкування з комп’ютером не як з бездушної технікою, а як з живою істотою. Прикладом може бути асистент Speaktoit [42] – це віртуальний помічник для iOS чи Android-пристроїв, що використовують природну мову для відповіді на питання, пошуку інформації, запуску додатків і роботи з різними Інтернет-сервісами, такими як Yandex, Google, Facebook, Twitter та багатьма іншими. Його показано на рис. 1.14, а. Також прикладом [43] може бути віртуальний помічник (показано на рис. 1.14, б) довідкової підсистеми із пакету офісного ПЗ Microsoft Office 97.

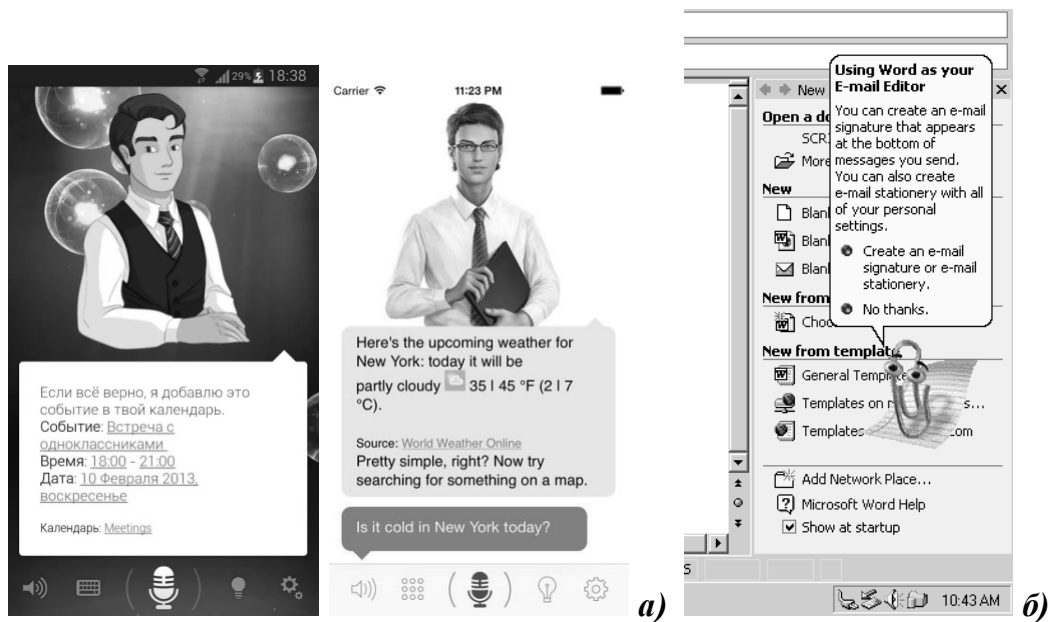


Рис. 1.14. Приклади соціалізованого інтерфейсу

1.5. Систематизація інтерфейсів користувача

1.5.1. Ієрархічна систематизація користувацьких взаємодій

На рис. 1.15 показана схема систематизації [19, 20] основних діалогових [10, 11] користувацьких взаємодій програмного забезпечення.



Рис. 1.15. Схема систематизації діалогових користувацьких взаємодій ПЗ

1.5.2. Класифікація кінетично-орієнтованих інтерфейсів

На основі результатів огляду [10, 19, 20, 45] новітніх концепцій розвитку інтерфейсів користувача, що представлені у попередніх підрозділах було розроблено ієрархічну класифікаційну схему кінетичних користувацьких інтерфейсів (англ. Kinetic User Interface). Вона представлена на рис. 1.16. Цей рисунок можна розглядати як доповнення до рис. 1.15, щоб акцентувати увагу на формах, видах і концепціях дизайну кінетичної користувацького інтерфейсу для взаємодії з програмним забезпеченням.



Рис. 1.16. Схема кінематичних підходів проектування інтерфейсів

1.6. Проектні процеси побудови інтерфейсів користувача

1.6.1. Проектування зручності та досвіду користувача

Проектування зручності (англ. Usability Engineering) – науково-прикладна дисципліна [46], метою якої є підвищенню ефективності, продуктивності та зручності використання інструментів діяльності.

Ця галузь пов'язана із взаємодією людини з комп'ютером і зокрема зі створенням людини-комп'ютерних інтерфейсів, які мають високу практичність і зручність у використанні. Оцінює зручність інтерфейсу і рекомендує шляхи підвищення його компетенції юзабіліті-інженер для поліпшення юзабіліті програмного забезпечення графічного інтерфейсу користувача (англ. GUI), веб-інтерфейса користувача і голосових користувацьких інтерфейсів (англ. Voice UI). Поєднує дисципліни: психологія, людський фактор і когнітивні науки. Основні області: людське сприйняття і дії, людське пізнання, поведінкові методи дослідження, кількісні та статистичні методи аналізу.

Термін інженерного використання (на відміну від дизайну і проектування взаємодії користувачів) застосовується для оцінювання та вироблення рекомендацій для покращення юзабіліті, ніж для дизайну. На рис. 1.17 наведено послідовність етапів процесу проектування зручності використання [46].

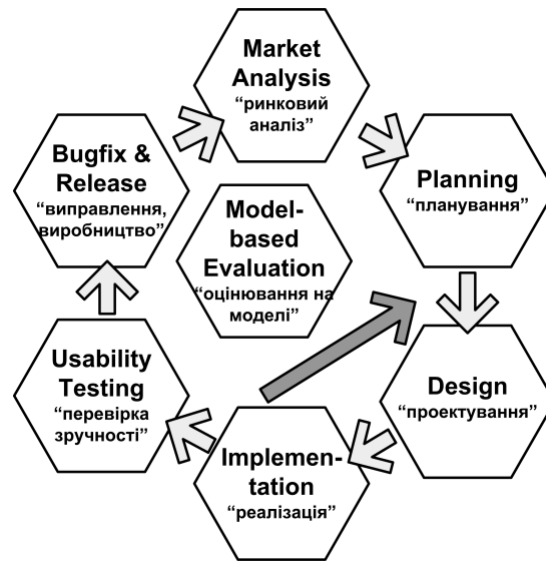


Рис. 1.17. Схема етапів проектування зручності

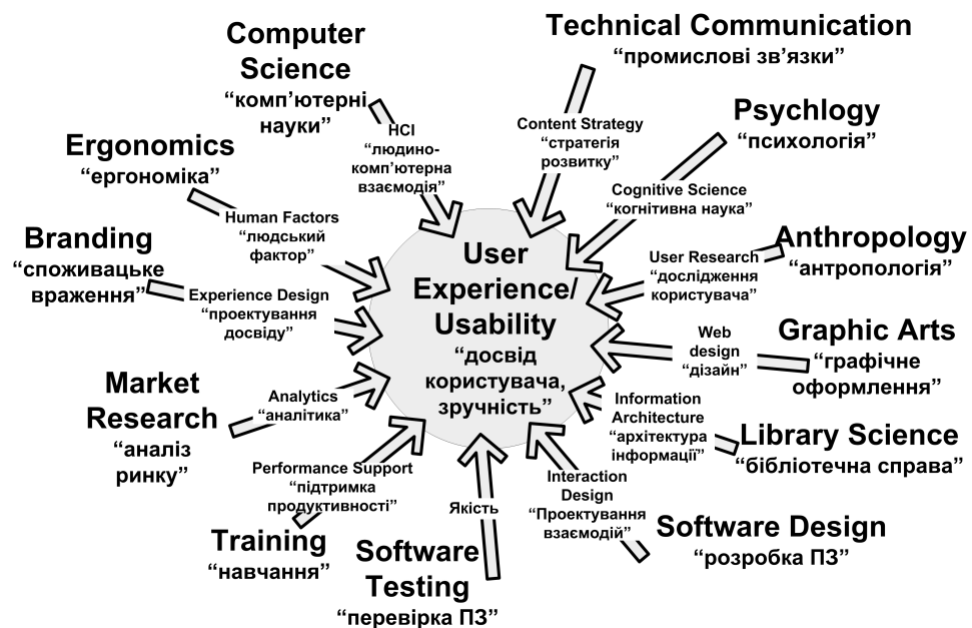


Рис. 1.18. Схема впливу галузей при проектуванні досвіду користувача

Проектування досвіду користувача (англ. User Experience Design, UXD) (схема на рис. 1.18) є підмножиною області досвіду проектування, яка належить до створення архітектури та моделей взаємодії, які впливають на зручність роботи з пристроєм або системою. Є спрямованість на всі аспекти взаємодії користувача з продуктом: як сприймається, пізнається і використовується. [47]

Досвід користувача (англ. User Experience, UX) – те, як людина відчувається при використанні продукту, системи або послуги. Досвід користувача має суб’єктивний характер, оскільки йдеться про почуття людини і думки про систему. Досвід користувача є динамічним, оскільки змінюється з часом, як і зміни обставин використання.

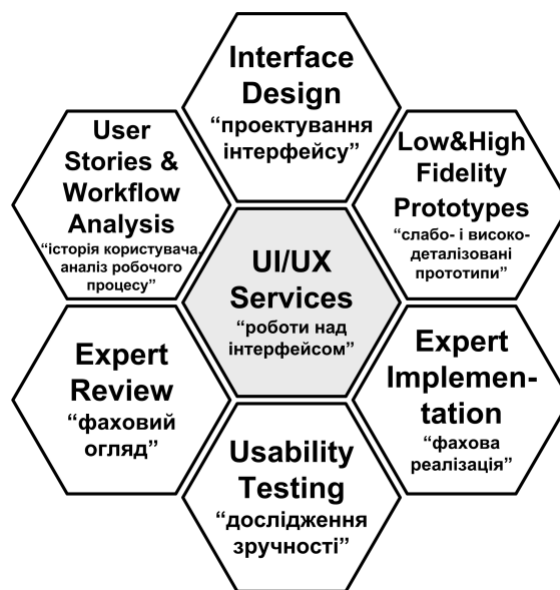


Рис. 1.19. Схема видів послуг з проектування досвіду користувача

1.6.2. Складові процесу ергономічного проектування ПЗ

В україномовній літературі часто термін «зручність використання» заміняють фонетично коротшим терміном «юзабіліті» від англ. Usability.

1. Команда ергономічних проектувальників [49]. Взаємодія з іншими розробниками системи:

- Юзабіліті-менеджер; юзабіліті-аналітик; юзабіліті-експерт; юзабіліті-інженер; юзабіліті-тестер
 - Ергономіст-проектувальник.
 - Спеціаліст із методів антропологічного спостереження (інженерний психолог, етнограф, соціолог).
 - Технічний письменник / письменник користувацької документації.
 - Дизайнер графічного інтерфейсу користувача (англ. GUI designer).
 - Спеціаліст із навчання користувачів.
2. Попереднє проектування (передпроектний аналіз, аванпроект, постановка завдання):
- Експертиза аналогів і прототипів.
 - Розробка ергономічних вимог.
 - Юзабіліті-дослідження.
 - Постановка завдань.
 - Планування робіт.
3. Розробка (проекування, конструювання):
- Проектна документація.
 - Проектні вимоги.
 - Концептуальне проектування.
 - Детальні прототипи.
 - Користувацька документація.
4. Тестування (внутрішнє) та випробування (b-тестування):
- Впровадження.
 - Юзабіліті-дослідження.
 - Навчання.
 - Супровід, збір та аналіз даних.
 - Фіксування досвіду проектування.

1.7. Огляд засобів проектування інтерфейсів користувача

1.7.1. Інтегровані середовища розробки ПЗ (IDE)

Більшість сучасних популярних середовищ розробки програмного забезпечення [50, 51, 52, 53, 55] не містять засобів для врахування аспектів історичної реалізації, еволюції компонентів графічного інтерфейсу

користувача (англ. – GUI), щоб забезпечити можливість вибору альтернатив для інтерфейсного управління. При створенні нових версій програмного забезпечення програмісти та менеджери повинні мати змогу порівняти компоненти екранного інтерфейсу користувача та інформаційні технології [99, 100] для обрання їх для розроблення САПР. У інтегрованих середовищах [50, 51, 52, 53, 55] розробки програмного забезпечення (англ. IDE) використовуються набори компонентів інтерфейсу користувача. Переважна більшість цих компонентів є невід’ємною складовою середовищ. Набори встановлених заздалегідь компонентів інтерфейсу з прив’язкою вихідних кодів до мові програмування, для якої призначена IDE, є теж недоліком такої ситуації. У табл. 1.1 показані кількісні характеристики наявних компонентів інтерфейсу користувача для різних середовищ розробки ПЗ.

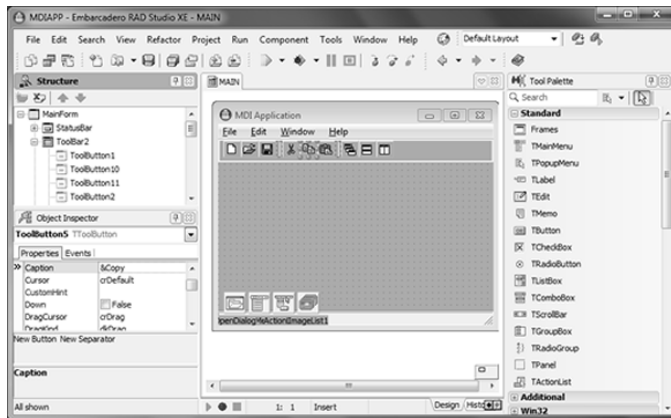
Таблиця 1.1.

IDE з конструкторами графічного інтерфейсу користувача

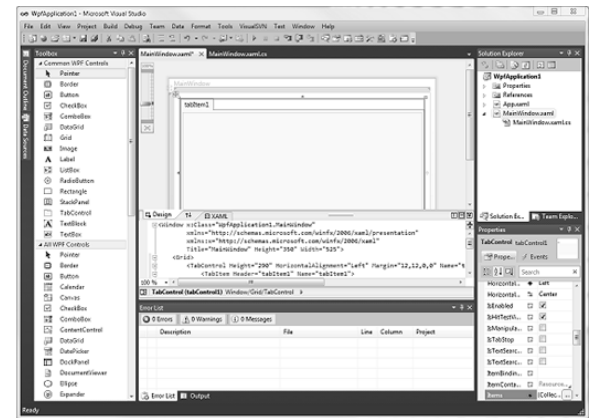
Назва інтегрованого середовища розробки ПЗ (IDE)	Мова програмування	Кількість типових компонентів інтерфейсу користувача
RAD Studio XE (Delphi)	ObjectPascal, C++, HTML	80+
Lazarus	ObjectPascal	50+
Microsoft VisualStudio	VB, C/C++, C#, HTML	80+
MonoDevelop	C#	45+
NetBeans/Eclipse	Java, JS	40+
QT (Nokia)	C++, JS	40+

Розробники не мають змоги перемикатися або експортувати / імпортувати готові форми GUI з однієї мови програмування в іншу.

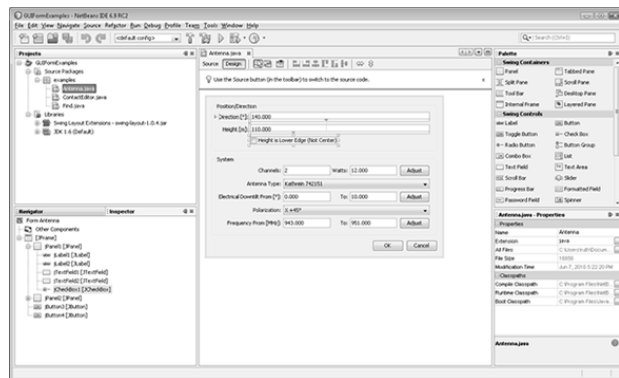
На рис. 1.20 показані центральні віконні форми середовищ розробки ПЗ із візуальними редакторами [54] інтерфейсних форм з наборами стандартних елементів інтерфейсу.



а) Embarcadero RAD Studio [50]



б) Microsoft Visual Studio [53]



в) NetBeans IDE GUI Builder [52]



г) Qt Creator IDE [55]

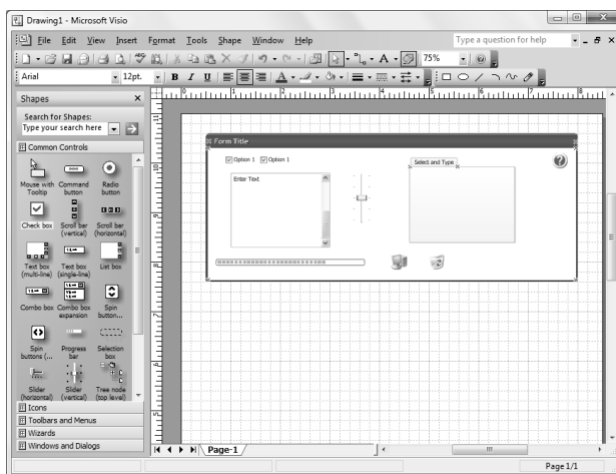
Рис. 1.20. Приклади віконних інтерфейсів IDE з редакторами WYSIWYG і їхні набори елементів управління користувацьким інтерфейсом

Редактори віконного інтерфейсу користувача містять лише основні елементи інтерфейсу. Також допускається збільшення кількості компонентів інтерфейсу, за рахунок підключення додаткових бібліотек сторонніх виробників. Програміст вручну вибирає тільки ті компоненти, які доступні зі списку IDE. Для розробників нових програмних продуктів (або для оновлення вже існуючих засобів ПЗ) буде зручно мати централізовану специфікацію та інструмент пошуку, який включає в себе реалізацію варіантів користувацького інтерфейсу.

1.7.2. Середовища прототипування інтерфейсів ПЗ

Крім середовищ проектування інтерфейсу є середовища прототипування інтерфейсів [56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70]. Прототип проекту інтерфейсу – це версія програми або веб-сайту у спрощеному вигляді. Прототипи створюються у спеціалізованих редакторах. Прототипи називаються каркасними, коли вони абстрагуються від деталей графічного дизайну і зосереджені на плануванні й структуруванні графічного інтерфейсу користувача, з використанням блоків-наповнення. Прототип дозволяє показати розміщення елементів інтерфейсу на екрані пристрою.

На рис. 1.21 представлені приклади віконних форм двох середовищ прототипування інтерфейсів користувача. У всіх розглянутих середовищах нема функціоналу для генерування програмного коду реалізації розроблених прототипів інтерфейсу.



а) Microsoft Visio [56]

б) iPhone MockUp Online tool [69]

Рис. 1.21. Приклади середовищ прототипування інтерфейсів користувача

Аналогом онлайн-сховища прикладів елементів інтерфейсу користувача можна виділити UICloud [70] (віконні форми представлені на рис. 1.22). Недоліки: не представлені всі типи елементів, нема різноманіття реалізації програмними кодами.

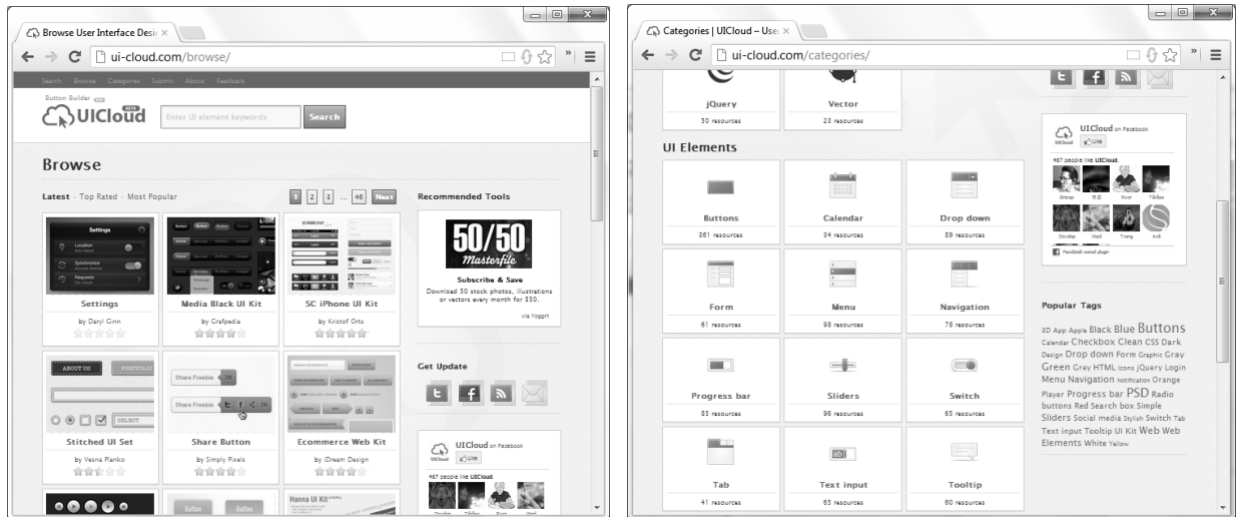


Рис. 1.22. Приклади онлайн-сховища елементів інтерфейсу користувача
UICloud

Таблиця 1.2.

Порівняльний аналіз засобів прототипування ІК

Критерій порівняння	Програмний засіб прототипування				
	PopApp	Protosketch	Napkee	Wire2app	Необхідно
Прототипування мобільних інтерфейсів	+	+	-	-	+
Прототипування десктопних інтерфейсів	-	-	+	+	+
Робота на мобільних пристроях	+	+	+	+	+
Робота на десктопних комп'ютерах	-	-	+	+	+
Робота зі сфотографованими ескізами	+	+	-	-	+
Робота з файлами розробленими у граф. редакторах	-	-	+	+	+
Генерування програмного коду прототипу	-	-	+	+	+

У табл. 1.2 відображені результати порівняльного аналізу функціональних можливостей існуючих програмних засобів [71, 72, 73] прототипування графічних ескізів інтерфейсних форм ГПК.

1.8. Мови декларування користувацьких інтерфейсів

1.8.1. Основні характеристики мов декларування інтерфейсу

Серед мов [21] декларування інтерфейсу користувача наявні такі:

UIML (англ. *User Interface Markup Language*) – у цю мову закладено концепцію, передавання даних від ядра програми до фізичного відображення через рівень логіки інтерфейсу і представлення, що відповідає сучасному MVC (англ. *Model View Controller*) підходу багаторівневої розробки ПЗ. Визначає: складені елементи UI, способів відображення, тип вмісту, реакції на дії користувача, тип контролю за діями, зв'язок із зовнішніми API [78, 79, 80, 81, 82].

XUL (англ. *XML User Interface Language*) – мова опису розмітки для динамічного створення користувацьких інтерфейсів на базі описів мовою XML (англ. *eXtensible Markup Language*). Дає можливість розділити представлення даних і логіку програмного ядра. Діє за принципом абстрактних структурних шарів: вміст, оформлення, локалізація. Ефективна мова для розроблення будь-яких програм та розширень, пов'язаних з веб-ресурсами. Розроблена Mozilla Foundation [78, 83, 84, 85, 86].

XAML (англ. *eXtensible Application Markup Language*) – розширювана мова опису розмітки декларативного програмування, розроблена Microsoft. Визначає набір властивостей, методів та подій, які дозволяють об'єднати веб-документи в об'єднану прикладну програму. Програмні об'єкти контролюють виконання програми і генерують події для коду UI. У XAML-документ також включено програмну логіку та стилі. Функціонально її можна розглядати як комбінацію XUL, SVG (англ. *Scalable Vector Graphics*),

CSS (англ. Cascading Style Sheets) та JavaScript (JS) в одній схемі XML [78, 87, 88, 89].

MXML (англ. *Macromedia eXtensible Markup Language*) – мова опису розмітки користувацького інтерфейсу, заснований на XML. Використовується переважно для декларативного опису інтерфейсу програми, а також для реалізації бізнес-логіки поведінки й інтернет-додатків. Мова дозволяє наочно описати структуру користувацького інтерфейсу. Розробляється специфікація фірмою Adobe в рамках технологій Flex та AIR [78, 90, 91, 92, 93].

Слід зазначити наявність інших мов опису користувацьких інтерфейсів, які через малу популярність серед програмістів, або з причин відмови від підтримки розвитку розробниками, або через вузькоспеціалізованість застосування втратили свою теперішню актуальність для розвитку в САПР. Серед них:

AUIML (англ. *Abstract User Interface Markup Language*) – це діалект XML, який є платформою і технологічно нейтрального подання компонентів інтерфейсу користувача. AUIML відображає інформацію відносного розташування компонентів графічного інтерфейсу і делегатів, показує їх у конкретній платформі рендерингу зображення. Залежно від платформи або пристрою візуалізації AUIML вирішує, як найкраще представити графічний інтерфейс користувача і отримувати користувацьке керування. AUIML дозволяє розробникам створювати додатки і запускати їх в Swing або на WWW без будь-яких змін. Проект ініційований компанією IBM, але зараз розвиток мови припинено [94].

XAL (англ. *eXtensible Application Language*) – технологія створена Nexaweb. Для розробки додатків використовує розмітку XML, яка задає інтерфейс користувача. XAL є відкритою декларативною мовою для створення корпоративних додатків Web 2.0. Вона була розроблена для роботи з іншими провідними специфікаціями для декларативного

розроблення додатків і масштабування під час виконання операції в межах програмного середовища Universal Client Framework (UCF) [95].

LZX (англ. *OpenLaszlo XML*) – декларативна мова опису користувацького інтерфейсу, яка визначає елементи інтерфейсу, компонування програмного додатку та скриптові виклики подій. Є вільною відкритою технологією. OpenLaszlo підтримує об'єктно-орієнтований підхід та прив'язку даних до елементів інтерфейсу відображення дозволяє створювати програмні додатки в одному файлі зі всіма вкладеними даними [96].

HMVCUL (англ. *Hierarchical Model View Controller User Interface Language*) – мова користувацького інтерфейсу, заснована на XML, яка підтримує створення та зв'язування елементарних компонент у MVC триади. Зв'язування досягається відслідковуванням деревовидної структури, яку описано в файлі. Крім того, оскільки користувацький інтерфейс побудований з функціональних атомарних блоків, то блоки можуть бути повністю багаторазові й взаємозамінні між додатками [97].

UIDL (англ. *User Interface Description Language*) – об'єктно-орієнтована мова заснована на мові JavaScript. Коди виконуються у спеціалізованому клієнтському середовищі з асинхронним доступом до серверних даних. Користувацький інтерфейс може бути одночасно описаний вставками на мові текстової розмітки HTML [98].

1.8.2. Порівняння характеристик мов розмітки UI

У порівняльній табл. 1.3 показані критерії оцінки мов розмітки інтерфейсів користувача ПЗ для застосування у розробці САПР.

Таблиця 1.3.

Порівняльна таблиця мов розмітки UI

Мова Критерій	UIML	XUL	XAML	MXML (Flex)
1	2	3	4	5
Якість користувацьких інтерфейсів проекту САПР (Quality) → max				
Розширювані елементи користувацького інтерфейсу, навігації та контролю даних	Нема безпосередньої підтримки	Достатні можливості	Добре	Відмінно
Спецефекти елементів користувацького інтерфейсу	Java AWT.	Доступно через CSS, розширення	Відмінно	Відмінно
Візуальна деталізація	Є підтримка	Є підтримка	Відмінно	Відмінно
Векторна графіка	ні	SVG	Відмінно, SVG	Відмінно
Самотестованість	XML, DTD..	XML, DTD	XML, DTD	XML, DTD
Автоналаштуваність	Є підтримка	Є підтримка	Є підтримка	Є підтримка
Витрати на реалізацію користувацьких інтерфейсів САПР (exPenses) → min				
Ліцензія	Нема, вільне використання	Вільне використання, MPL/GPL/LGPL	Microsoft Public License	Mozilla Public License
Доступність платформи	v.3.1 (березень 2004)	Постійне розширення Mozilla Foundation.	Microsoft Windows Presentation Foundation	Постійне розширення Adobe
Серверна частина	Довільний процесор	Нема вимог	Windows Server	Adobe Flex Server, PHP .
Об'єктно-орієнтована мова програмування	JS, Java	JS, Java, C++,., Python.	C#, VB.NET..	ActionScript.
Середовище розробки	Довільний редактор	Довільний редактор. XULExplorer	MS Expression, MS VisualStudio, XAMLPad	Adobe Flex Builder

Таблиця 1.3 (продовження)

1	2	3	4	5
Час на реалізацію/модифікацію користувацьких інтерфейсів САПР (pRoductive time) → min				
Готовність впровадження технології	Висока. Бібліотеки під .NET, Java, J2EE, HTML, WML., VB., C++, Symbian, CORBA	JS, DOM, CSS.	Висока для платформи Microsoft.NET Framework 3.0	Adobe Flex, ActionScript, .NET, Java
Технічна сумісність, відповідність стандарту	OASIS, W3C XML	W3C XML	W3C XML	XML, ECMAScript, CSS
Макровизначення	Є підтримка	Є підтримка	Є підтримка	Є підтримка
Складність реалізації /модифікації користувацьких інтерфейсів САПР (complexity) → min				
Асоціативний рівень сприйняття програмної моделі (явна семантика)	Добре	Відмінно	Добре	Добре
Модель подій інтерфейсу (діапазон контрольованих подій, ефективність моделі)	Відмінно	Добре	Добре	Добре
Документація	Добре	Відмінно	Середньо	Відмінно

1.9. Висновки до розділу

Отримані результати цього розділу дозволили узагальнити концепції, підходи, стилі, форми, ролі, режими, патерни, метафори побудови інтерфейсів користувачів, що дало змогу здійснити систематизацію понять. Оцінені недоліки та переваги інформаційних засобів програмування, прототипування та описів кодів інтерфейсів користувача. Розгляд процесів проектування інтерфейсів користувача показав відсутність прямого зв'язку та засобу між даними використання готових інтерфейсів і можливості їх використання безпосередньо на стадіях планування, прототипування та реалізації.

Розділ 2. Метрики, конструктиви та методи побудов інтерфейсу користувача

У другому розділі виявлені та описані факторні параметри та метрики, які мають вплив на якість створюваних інтерфейсів користувачів, запропонований спосіб формалізації досвіду користування накопичення діалогової інформації на площинного компонуванні інтерфейсу, здійснена та представлена систематизація типових конструктивів графічного інтерфейсу користувача ПЗ, запропонована схема процесу поділу макетів каркасів компонування ГІК, запропоновано критерії для методу згорток для оцінювання характеристик засобів прототипування ІК та мов програмування ГІК. Описані процеси моніторингу інтерактивних взаємодій користувачів із ГІК та процес автоматизації перетворення візуальних графічних прототипів віконних форм інтерфейсу у програмний код. Показаний алгоритм процесу візуальної ідентифікації конструктивів ІК та синтезу програмного коду. Модифіковано процес перепроєктування інтерфейсу користувача з одних програмних кодів в інші.

Основні результати розділу опубліковані автором у працях [21, 54, 76, 99, 100, 101, 107, 108, 115, 119, 120, 124]

2.1. Моделювання факторних параметри впливу користувача

До параметрів людино-технічних, людино-інформаційних комплексів САПР належать як параметри апаратного і програмного забезпечення інформаційно-обчислювальних комплексів, так і параметри користувача (оператора), людини.

Виокремлення людських факторів [101] у інформатизованих комплексах та їхній вплив на сумарну продуктивність та стійкість інформаційно-обчислювальних комплексів є факторами загальної

інтелектуалізації шаблонів (патернів) проектування інтерфейсів користувачів.

У багатокористувацькому комплексі САПР ефективність діяльності групи користувачів визначається: злагодженістю колективу, умінням операторів взаємодіяти між собою, психологічною сумісністю групи. Для забезпечення якісної спеціальної роботи користувачів необхідно підбирати їх за певними антропогенними параметрами; необхідно попередньо проводити навчально-тренувальні процедури тривалий час; забезпечувати тестування і адаптацію на початку роботи. При проектуванні інформаційного комплексу здійснюється відбір і тестування «ідеального» оператора-користувача. Формальна модель користувача відображає середньо-статистичні параметри.

Впливи на роботу користувача поділяються на дві групи: суб'єктивні і об'єктивні фактори. Суб'єктивні: фізіологічний стан оператора (тиск, температура, гормональний стан, ситість, сонливість ...), психологічний стан (врівноваженість, дисциплінованість, бажання добре виконувати задану роботу), рівень підготовки тощо. Ці фактори займають дуже великий сектор імовірності успішного виконання завдання. Об'єктивні фактори поділяються на дві групи: 1) фактори зовнішнього середовища, в якому користувач людино-технічного (інформаційного) комплексу буде виконувати свої функції; 2) параметри та характеристики апаратного забезпечення (апаратні фактори). Фактори зовнішнього середовища: тривалість робочих змін, апаратні фактори організації робочого місця користувача, кількість потоків інформації (сприйняття й аналізу), відповідність кількості інформації за одиницю часу (яка відповідає можливостям її сприймати і переробляння), підвищений ступінь відповідальності та ефективність роботи оператора в аварійних ситуації (при збоях функціонування).

Системою тестування визначається працездатності користувача як у критичних ситуаціях, так в складних інформаційних ситуаціях (наприклад стислі часові терміни проектування). Користувач-оператор під час виконання

робіт повинен відчувати відповідальність, а з другого боку його емоційні фактори не повинні спонукати його до паніки чи розгубленості. Користувач у кризах має стати емоційно готовий приймати правильні рішення за короткі терміни. Користувачам яким важко приймати рішення при обранні правильного способу оперування з інтерфейсами ПЗ, слід застосовувати попередні тренування за сценаріями інтерактивних взаємодій. Набори тестів, методів перевірки і діагностування параметрів користувачів, як правило, містять такі параметри: швидкість сприйняття тестової інформації з екрану (наприклад: кількість символів за хвилину), швидкість сприйняття інформації, швидкість прийняття рішення, формування команд на робочому місці тощо. Такі дані можуть мати певну вірогідність для конкретного робочого місця і конкретних задач. Така оцінка може бути сформована як достовірність отриманого позитивного результату, або інтегральна оцінка, яка враховує часові параметри і якість виконання конкретного тесту.

Інтегральний [101] показник «якості» оператора-користувача можна представити залежністю:

$$Q = \sum_{i=1}^n \frac{P_1 + P_2 + P_3 + \dots + P_t}{P_{1\max} + P_{2\max} + P_{3\max} + \dots + P_{t\max}} \quad (2.1)$$

де: $P_1 \dots P_t$ – параметри оператора на i -му тесті із n ; $P_{1\max} \dots P_{t\max}$ – максимальне значення параметрів оператора на i -му тесті.

Звісно, що доцільність організації користувацько-інтерфейсної взаємодії:

$$S = I_{hc} / I_h \quad (2.2)$$

де: I_{hc} – ймовірність виконання завдання із заданими показниками якості при користувацько-інтерфейсній взаємодії, I_h – ймовірність виконання

завдання з тими ж показниками якості без застосування інтерфейсних засобів.

Та коефіцієнт витривалості інформаційної системи з інтерфейсом користувача при вирішенні великої кількості складних завдань у процесі функціонування без погіршення якості виконання буде в першу чергу залежати від людського фактору:

$$H = t_h / t_{h_best} \quad (2.3)$$

де: t_{h_best} – найкращий час виконання завдання користувачем, t_h – загальний час рішення серії однорідних задач користувачем.

Варіативність взаємодії користувача із інтерфейсом є кількістю способів використання конструктивних елементів інтерфейсу для вирішення того самого завдання. Для різних користувачів в системі ефективність різних варіантів є неоднакова в різних робочих умовах. Чим більше варіантів роботи, тим вища ймовірність як виконання дії, так і поступового обрання користувачем зручного варіанта управління, близького до його індивідуальних особливостей.

$$V_i = F_i \cdot U_i \quad (2.4)$$

де: V_i – ймовірність виконання дії, F_i – ймовірність що користувач знає 1-й спосіб виконання дії, наданих автоматизованою системою, U_i – ймовірність що не буде відмови системи при цих діях.

Ймовірність успішного виконання дії користувач-інтерфейс системи яких-небудь способом:

$$E = 1 - \prod_{i=1}^M (1 - V_i) = 1 - \prod_{i=1}^M (1 - F_i \cdot U_i) \quad (2.5)$$

де: M – кількість варіантів шляхів взаємодії.

Час досягнення області екрану (цілі) для взаємодії з інтерфейсним елементом прямо пропорційно дистанції до цілі й обернено пропорційно розміру цілі (зони активності елементів керування інформаційною системою):

$$t = a + b \cdot \log_2((d/z) + 1) \quad (2.6)$$

де: a, b – встановлюються дослідним шляхом за параметрами продуктивності користувача, d – відстань від поточного місця активності до точки нової активності, z – розмір зони активності у напрямку руху активації взаємодії користувача із елементом керування системи.

2.2. Моделювання досвіду користувача накопиченням діалогової інформаційної діяльності

Ефективність та зручність використання користувачем-проектувальником віконних інтерфейсів програмного забезпечення САПР для розв'язку своїх проектних задач залежить від площинного компоновання інтерфейсних візуальних об'єктів функціонального керування (меню, кнопок керування, панелей інструментів, діалогових вікон, полів введення даних тощо). У такому випадку оцінкою забезпечення зручності кожному користувачу оболонки САПР може виступати збір (накопичення) статистичної діалогової інформаційної діяльності в межах інтерфейсного віконного обмеження свободи організації робочого місця проектувальника.

Користувачі головним чином оперують інтерфейсом ПЗ за допомогою маніпулятора типу «миша», і в меншій мірі клавіатурою [103, 104, 105]. Більшість користувачів переміщують маніпулятор типу «миша» у відповідності до тих функціональних інтерфейсних елементів ПЗ, на яких зосереджують свою увагу.

Метою дослідження [108] є: розгляд концепцій і здатностей автоматизованого програмного відстежування, реєстрації та візуалізації

діалогових взаємодій користувача інтерфейсів ПЗ САПР для подальшого їхнього аналізу зі створенням моделей інтерактивної густини інтерфейсу та трудової діяльності користувача в межах інтерфейсу.

Шляхи досягнення оцінки:

- автоматичне спостереження користувацьких взаємодій з інтерфейсами САПР (здійснено на прикладі експериментальних зразків екранних інтерфейсів САПР мікро електромеханічних систем (МЕМС));
- фіксування координат та вектору переміщень миші, пера планшету;
- фіксування натискання, кліків, виділення, наведень активації тощо;
- фіксування часу на читання тексту, вивчення варіантів елементів керування інтерфейсу.

Шляхи застосування отриманих даних:

- підвищення ефективності взаємодії з користувацьким рівнем системи САПР за рахунок допомоги проектній групі програмістів САПР прийняти рішення корегування сценаріїв діалогів [12, 13] ;
- тестування зручності користування на існуючому інтерфейсі;
- оцінки продуктивності переміщення курсору, рухові здібності користувача-проектувальника в САПР.
- формування оцінки навігаційної кінематичної поведінки користувача інтерфейсу.

2.2.1. Формалізація функціонального наповнення (профілю) площинного компоновання інтерфейсу

Матриця заповнення віконного інтерфейсу програми ($n \times m$ пікселів) елементами керування (здійснили програмісти):

$$MD = \left\| f_{xy} \right\|_{\substack{x=1..n \\ y=1..m}}, \quad (2.7)$$

де $f_{xy} \geq 0$ – ступінь функціонального призначення пікселя (належить піксель кнопці, тексту меню, прокрутці, групі елементів, панелі тощо).

Допустимі рівності близьких по сусідству значень $f_{xi} = f_{xj}, f_{iy} = f_{jy}, i \neq j$ – що визначають групу пікселів певного елемента керування інтерфейса вікна.

Для зручності візуального сприйняття все поле площини ІК поділяємо на сектора (доприкладу прямокутні зони) і за значенням ступеня функціонального призначення групи пікселів «піднімаємо» над поверхнею інтерфейса, з чого отримуємо геометричну інтерпретацію, яка представлена на рис. 2.1.

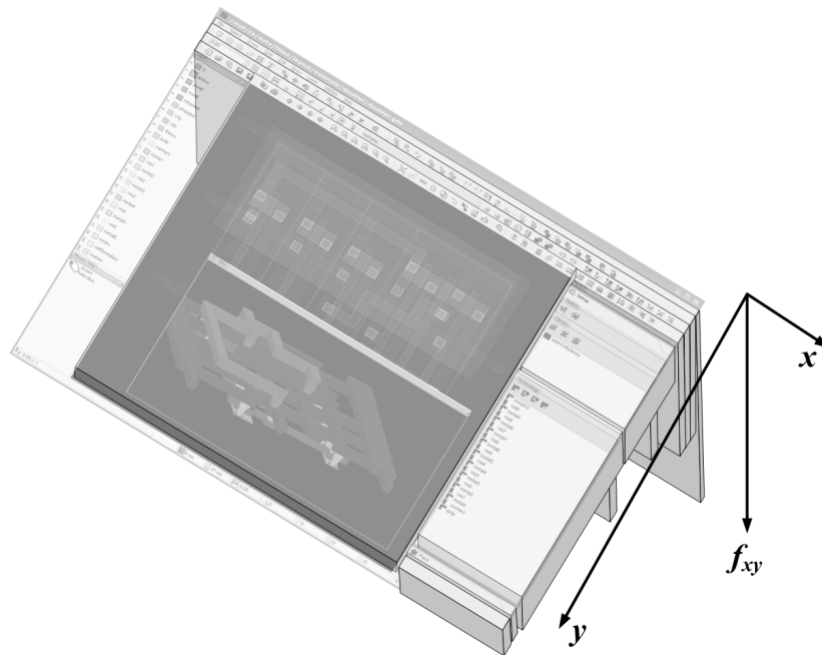


Рис. 2.1. Геометричне 3D представлення вагових ступенів функціонального призначення пікселя інформаційного поля інтерфейсу

Насиченість інтерактивними функціональними взаємодіями на полі екрану можна представити декомпозиційною моделлю у вигляді сукупності паралелепіпедів, що зображено на рис. 2.2.

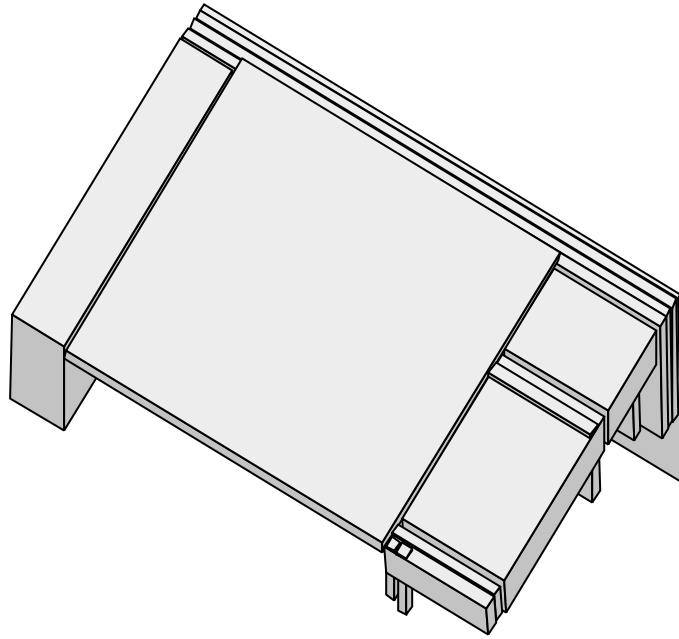


Рис. 2.2. Геометрична 3D стовпцева декомпозиційна моделі інтерпретації областей екрану інтерфейсу користувача

Ширина та довжина одної грані паралелепіпеда відповідає розмірам інтерфейсного елемента чи групованій області, а висота паралелепіпеда задається пропорційно до визначеного ступеня функціонального призначення чи інтенсивності густини взаємодії з цією областю екрану. Здійснюючи горизонтальні повздовжні зрізи на різних висотах, можна отримати кількісні дані площ з найбільшою густиною взаємодій користувача з зонами інтерфейсу.

2.2.2. Формалізація взаємодій користувача інтерфейсу програмного забезпечення

Матриця фіксації активних дій користувача віконного інтерфейсу програми:

$$MU = \left\| t_{xy} \right\|_{\substack{x=1..n \\ y=1..m}} \quad (2.8)$$

де $t_{xy} \geq 0$ – консолідований час активності над пікселем (переміщення, утримання, натискання, виділення, наведення тощо).

Оскільки у різних програмістів-проектувальників програмного забезпечення САПР є свої уподобання щодо розміщення елементів керування та рівня функціонального призначення, а у користувача інтерфейсами САПР свої, то варто звести шкали до нормалізованих:

$$fn_{xy} = \frac{f_{xy}}{(f_{x1} + \dots + f_{xm})}, \quad tn_{xy} = \frac{t_{xy}}{(t_{1y} + \dots + t_{ny})} \quad (2.9)$$

2.2.3. Кінематичні індекси активності користувача

Слід відзначити ще дві характеристики, що необхідно ввести в модель:

1. Статична активність користувача (A_s) – характеристика уваги:

$$A_s = \frac{n_s}{T \cdot f_{ps}} \quad (2.10)$$

де n_s – кількість зареєстрованих точок спостереження за положенням «миші» без переміщення; T – загальний час сесії активності користувача; f_{ps} – частота спостереження відліків за секунду.

2. Рухома активність користувача (A_m) – характеристика дієвості:

$$A_m = \frac{n_m}{T \cdot f_{ps}} \quad (2.11)$$

де n_m – кількість зареєстрованих точок спостереження за положенням «миші» з переміщення; T – загальний час сесії активності користувача; f_{ps} – частота спостереження відліків за секунду.

Загальна кількість зареєстрованих точок спостереження:

$$n_{s,m} = n_s + n_m \quad (2.12)$$

Доприкладу на рис. 2.3 показані три різних рівні точності реєстрації точок спостереження і вимірюваний у кількості відліків (кадрів) за секунду.

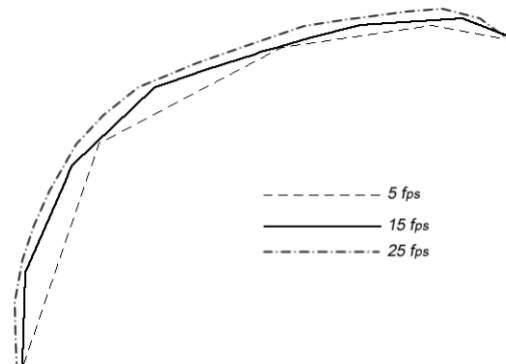


Рис. 2.3. Різні точності реєстрації за рухом маніпулятора типу «миша»

2.2.4. Візуалізація накопичених даних на зразках інтерфейсу

Як експериментальні зразки екрану користувача були використані віконні інтерфейси (рис. 2.4) "LayoutEditor" CAD Tool (<http://www.layouteditor.net>). "LayoutEditor" – це програма для розробки та редагувати макетів для виготовлення МЕМС/ІС. Вона також використовується для мультикристальних модулів (Multi-Chip-Modules), інтегральних схем (Chip-on-Board), низькотемпературного сплавлення кераміки (англ. LTCC), монолітних мікрохвильових інтегральних схем (англ. MIC), друкованих плат (англ. PCB), товстоплівкових технологій, тонко плівкових технологій. Тут задіяні складні функції: кути, генератор шрифту, C++ макроси, python сценарії, логічні операції, перевірка проектних норм, Netlist планування і LVS, 3D види. Файл формату Calma GDSII використовується як основний формат файлів. Крім того, формати файлів: OASIS (Open Artwork System Interchange Standard), OpenAccess, DXF, CIF (Caltech Intermediate Form) та Gerber (RS-274X) повністю підтримуються. Формати файлів LEF, DEF, Alliance, Lasi можуть бути імпортовані. LayoutEditor – це багатоплатформовий продукт на Windows, Linux, Mac OS X та інших операційних платформах [106].

Структура площинного компоунання та функціональний склад керуючих інтерфейсних елементів (меню, панелі інструментів, кнопки тощо) на рис. 2.4 є типовою для більшості ПЗ САПР різних галузей.

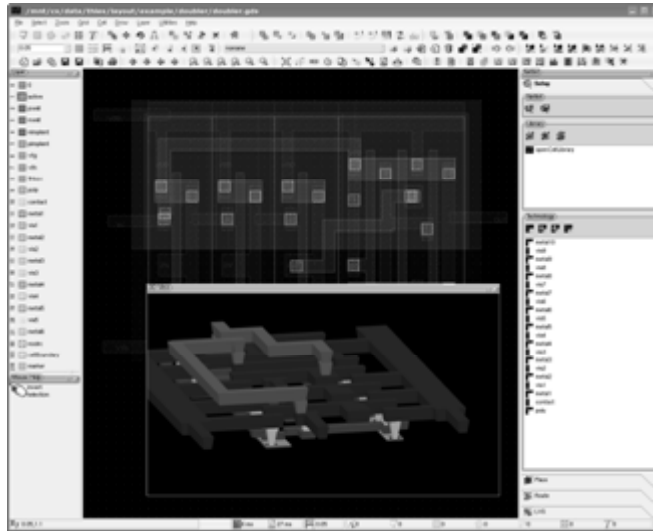


Рис. 2.4. Віконні інтерфейси LayoutEdit of MEMS/IC CAD Tool

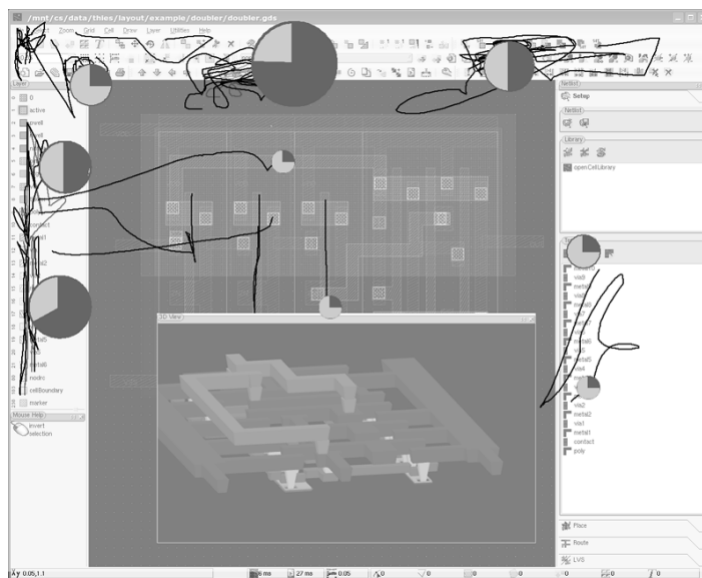


Рис. 2.5. Графічна візуалізація траєкторії «миші», часові записи користувацької активності дій з вікном LayoutEdit of MEMS/IC CAD Tool.

Усі сучасні САПР створено з використанням програмування користувацьких інтерфейсів на базі стандарту інтерфейсних елементів

операційної системи. Тому інтерфейсні елементи є типові, що дає змогу говорити про пошук і формалізацію закономірностей.

На рис. 2.5 лініями відображаються траєкторії переміщення вказівника маніпулятора «миші», а колами із заповненими секторами умовно позначаються часові затримки над певними областями екранного поля інтерфейсу користувача.

На рис. 2.6 показано співставлення графічної візуалізації траєкторій «миші» та «температурної карти» активності дій користувача з елементами інтерфейсу програмного забезпечення. Таке графічне інтерпретування дозволяє значно підвищити сприйняття даних та кількісно отримувати дані про різні зони поля густини взаємодії користувача з інтерфейсом.

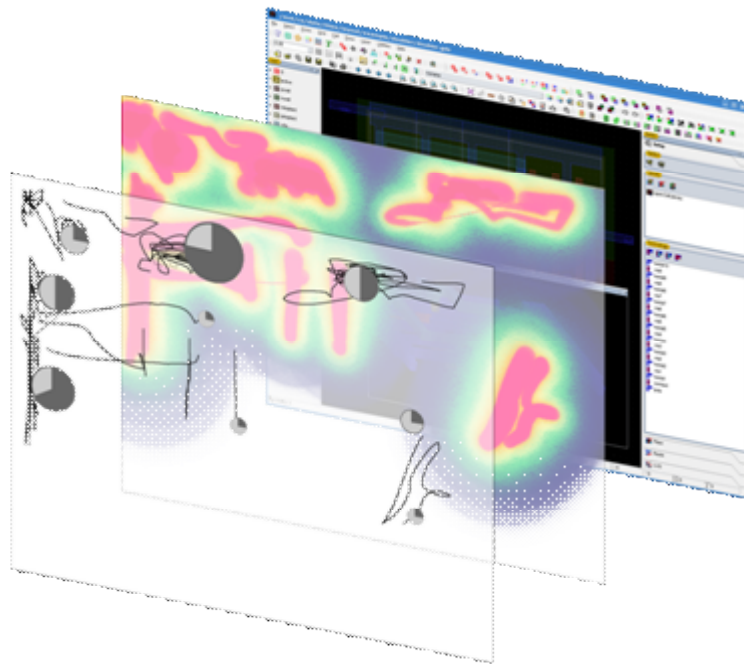


Рис. 2.6. Узагальнена «температурна карта» користувацької активності дій з інтерфейсами LayoutEdit of MEMS/IC CAD Tool.

2.3. Систематизація візуальних конструктивів інтерфейсу


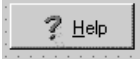
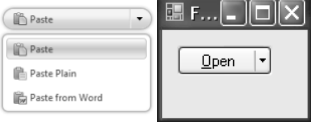
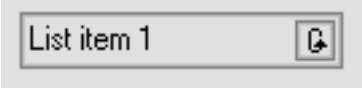
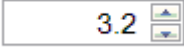


Інформаційне поле екранного інтерфейсу користувача ПЗ наповнене різноманітними візуальними компонентами. Ці компоненти виконують

функціональні дії та реакції на дії користувачів. Після ознайомлення із різними ПЗ, засобами прототипування інтерфейсів та середовищами програмування, було систематизовано візуальні конструктиви (табл. 2.1).

2.3.1. Типізація конструктивів графічного інтерфейсу

Таблиця 2.1.

Типи конструктивів графічного інтерфейсу користувача

№	Назва	Умовне позначення. Приклад вигляду.	Командний ввід / вибір	Контейнер	Ввід-вибір даних	Навігація	Інформаційні	Widget / Gadget
1	2	3	4	5	6	7	8	9
	Button		X					
1.	Icon button		X					
2.	Split button		X		X			
3.	Cycle button				X			
4.	Checkbox	Unchecked: <input type="checkbox"/> Checked: <input checked="" type="checkbox"/> Disabled: <input type="checkbox"/>			X			
5.	Radio button	<input type="radio"/> Unchecked <input checked="" type="radio"/> Checked <input type="radio"/> Disabled			X			
6.	Spinner (Stepper input)				X			
7.	Picker				X			
8.	Switcher (Toggles)		X		X	X		

Таблиця 2.1. (продовження)

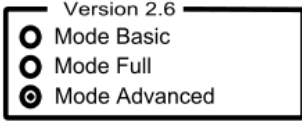
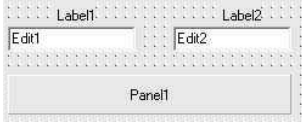
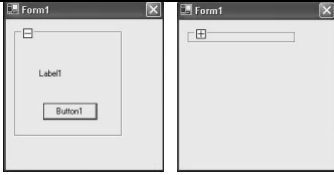

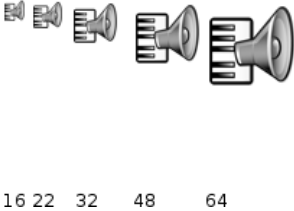
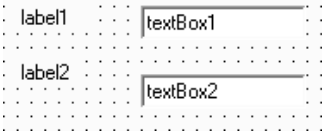
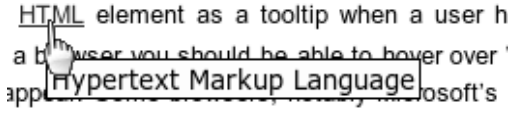
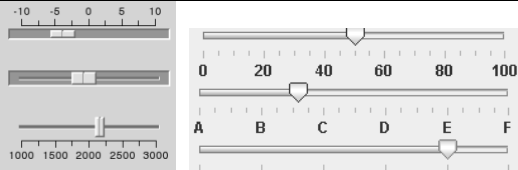
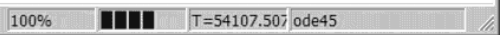
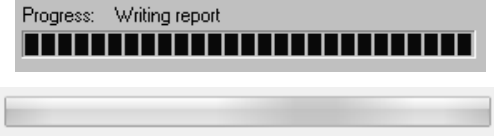
1	2	3	4	5	6	7	8	9
9.	Frame, Fieldset, Group box			X				
10.	Panel			X				
11.	Collapsible panel			X				
12.	Text edit		X		X			
13.	Icon						X	
14.	Label						X	
15.	Hyperlink	http://www.markelov.com.ua				X		
16.						X		
17.	Slider				X			
18.	Status bar			X			X	
19.	Progress bar						X	

Таблица 2.1. (продовження)


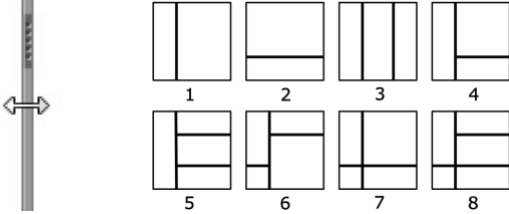
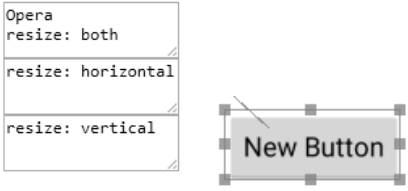
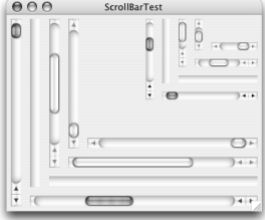
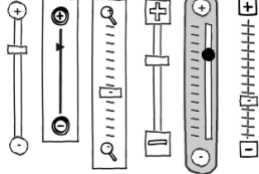

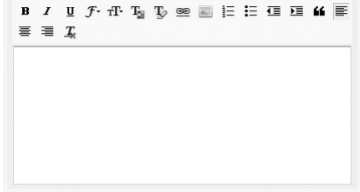
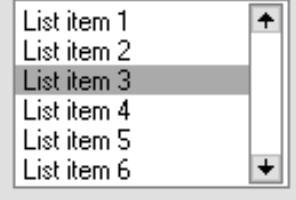
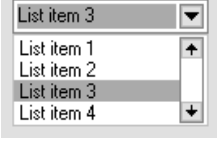
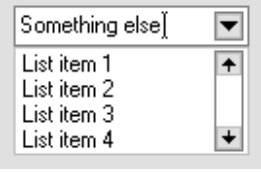


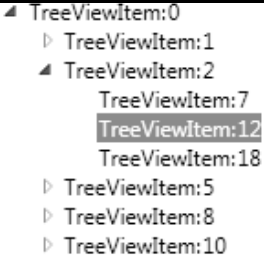
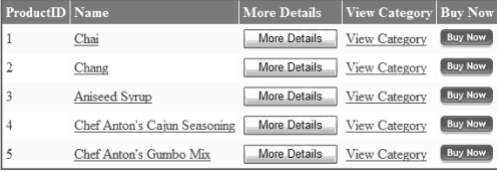
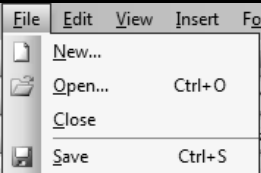
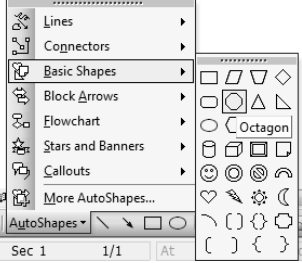
1	2	3	4	5	6	7	8	9
20.	Progress indicator (Throbber)						X	
21.	Splitter			X				
22.	Resizer			X				
23.	Scrollbar				X	X		
24.	Zoomer			X	X	X	X	
25.	Tabs (Pages)			X		X		
26.	Text Box (Memo, Rich Text)			X	X			
27.	List box				X			

Таблица 2.1. (продовження)

1	2	3	4	5	6	7	8	9
28.	Drop-down list		X		X	X		
29.	Combo box				X			
30.	Rating (Stars)				X		X	
31.	Tag cloud		X			X	X	X
32.	Tree view				X	X		
33.	Grid view			X				
34.	Menu		X	X		X		
35.	Hierarchical menus		X	X		X		

Таблиця 2.1. (продовження)

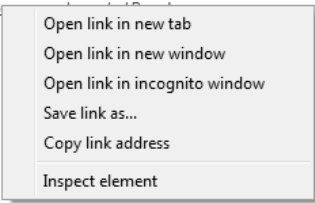

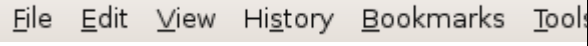


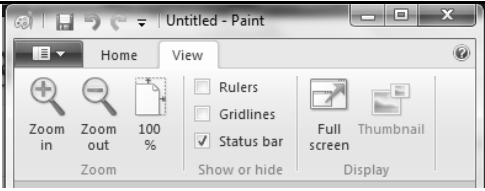


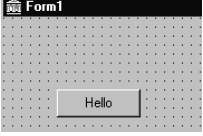
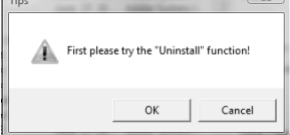

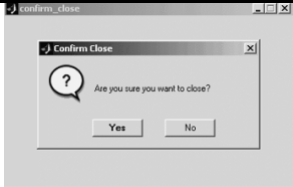
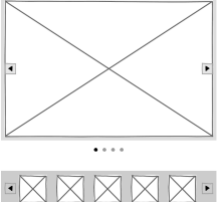
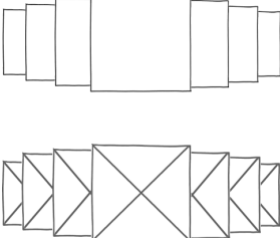
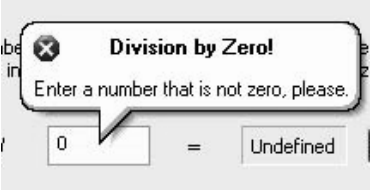
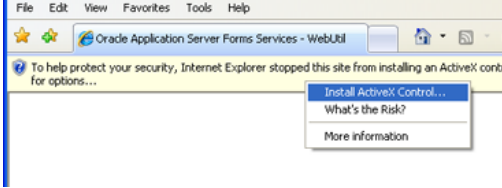

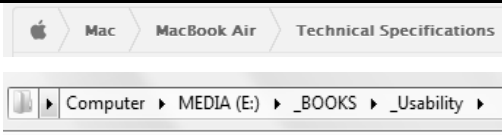


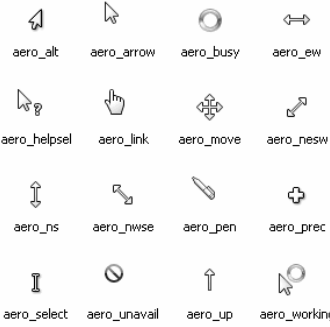
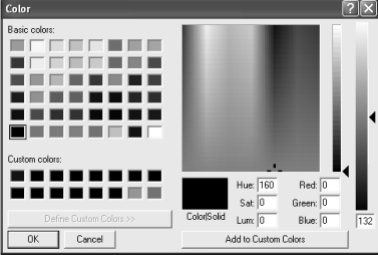
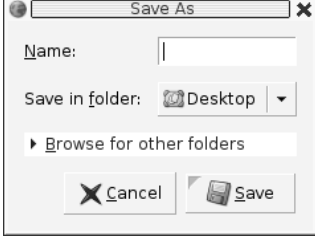

1	2	3	4	5	6	7	8	9
36.	Context menu		X					
37.	Pie menu (circle menu)		X					
38.	Menu bar			X				
39.	Menu extra (Dock)		X	X		X		X
40.	Toolbar			X				
41.	Ribbon			X				
42.	Accordion			X				
43.	Window			X				
44.	Modal window			X				
45.	Pop-up window			X				

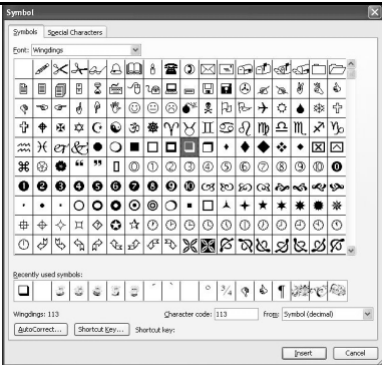

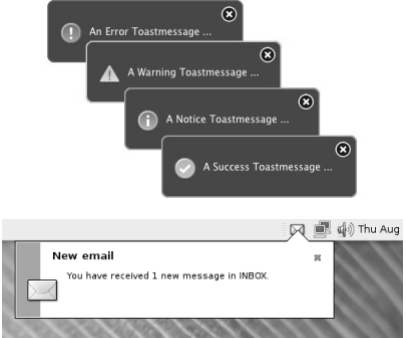
Таблица 2.1. (продовження)

1	2	3	4	5	6	7	8	9
46.	Dialog box			X				
47.	Alert box			X			X	
48.	Carousels			X		X		
49.	Coverflow carousel			X		X		
50.	Balloon help						X	
51.	Infobar			X			X	
52.	Address bar					X		
53.	PathViewer (Breadcrumb)					X	X	

Таблиця 2.1. (продовження)

1	2	3	4	5	6	7	8	9
54.	List nav bar			X		X		
55.	Pagination							
56.	Pointer (Cursor)					X	X	
57.	Color picker			X				X
58.	Disclosure widget			X				X
59.	Palette window (Utility window)			X	X			X

Таблиця 2.1. (продовження)

1	2	3	4	5	6	7	8	9	
	Palette window (Utility window)				X	X			X
60.	Inspector window				X				
61.	Toast (Notifier)				X			X	X

2.3.2. Спеціальні елементи GUI

Кнопка (англ. Button) – це графічне представлення електричних кнопок з'являється як частина графічного інтерфейсу користувача. Переміщення курсору миші, пера чи жесту руки над графічною кнопкою і натискання однієї з фізичних кнопок пристрою керування починає деякі програмні дій, такі, як виклик функції, закриття вікна або видалення файлів тощо.

Кільцеве меню (англ. Pie menu) – компонент меню забезпечує круговий вигляд команд меню з поділом на сектори. Як правило, спрацьовує,

як контекстне меню, а також дозволяє навігації спрямованості, жесту і традиційні рухи миші [44]

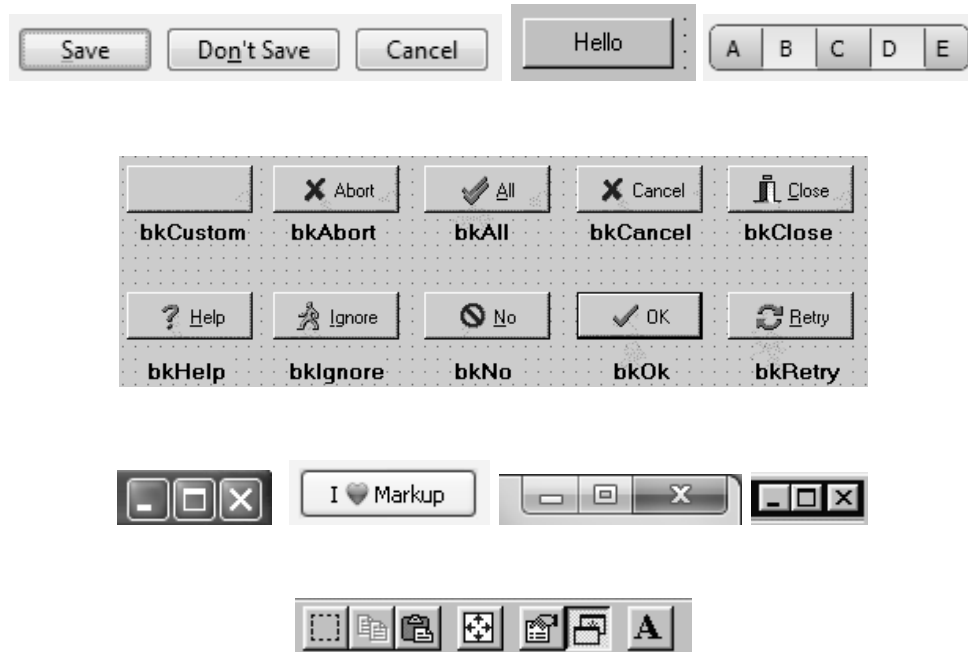


Рис. 2.7. Приклади графічних кнопок

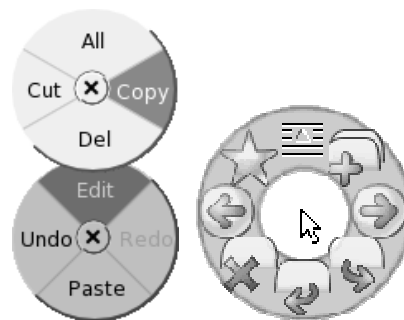


Рис. 2.8. Приклади кільцевого меню (Pie menu)

Циклічна кнопка (англ. – Cycle button) – графічний елемент користувацького інтерфейсу, який дозволяє користувачеві вибрати один з попередньо визначеного набору опцій. Він використовується як кнопка, зміст якої змінюється з кожним натисненням і циклів між двома або більше значень і відображається значення вибору користувача.

Перевага циклічної кнопки на відміну від радіо-кнопки або списку є те, що вона займає менше місця на екрані; її перевага перед розкритим списку

в тому, що одного клацання «миші» достатньо, щоб переключатися між двома варіантами вибору параметрів.

Однак недоліком є те, що за наявності багатьох варіантів значень, необхідно здійснити багато кліків, щоб перейти на шуканий. Ось чому деякі GUI середовища мають можливість відображення конструктива ІК у вигляді або циклічної кнопки, або випадаючим списком, в залежності від кількості варіантів значень.

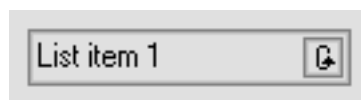


Рис. 2.9. Приклад циклічної кнопки (Cycle Button)

Підчас розроблення програмного забезпечення, «тости» (англ. Toast) – це невелике інформаційне вікно, у якому відображаються деякі види миттєвих повідомлень. «Тости» повідомляють користувачеві про різні події, такі як отримання нового повідомлення електронної пошти, завершення процесу завантаження, поява оновлення або зміни в мережі. Таку назву «тости» пов'язують із подібністю як на кухні хлівні скибочки тостів вистрибують з тостера для обсмажування. А на екрані до виринання міні-вікна слайдом вгору в полі зору.



Рис. 2.10. Приклади «тостових» виринаючих нагадувань (Toast)

2.4. Процес поділу поля макетів каркасів компоновання ГІК

Ефективність та зручність використання користувачем програмного забезпечення для розв'язання певних задач залежить від площинного

компонування інтерфейсних візуальних об'єктів функціонального керування, таких як меню, кнопок керування, панелей інструментів, полів введення даних тощо. Постійно збільшується кількість різноманітних мобільних пристроїв з різними розмірами дисплеїв (1.5", 3.2", 3.5", 3.7", 3.8", 4", 4.3", 4.5", 4.7", 5", 5.3", 5.7", 6", 7", 7.7", 7.9", 8", 8.9", 9.7", 10.1" та інші), роздільною здатністю (QVGA, HVGA, VGA, SVGA, qHD, WSVGA, XGA+, HD 720p, WXGA+, UXGA, Full HD, 2K, Ultra HD, Super Hi-Vision тощо), співвідношеннями сторін екрану (3:2, 4:3, 5:3, 8:5, 16:9, 16:10, 71:40 та інше) та щільністю зображення [74, 75].

Відповідно постає питання щодо можливості перенесення програмного засобу з одного пристрою чи з персонального комп'ютера на інший. Невеликий за розмірами дисплей по відношенню до кількості даних, які будуть відображатися, є реальною проблемою під час розроблення інформаційних систем. Відповідно витрачається багато часу для розроблення нових зручних для конкретного пристрою форм програми в залежності від розмірів віконного інтерфейсу.

Важливим фактором графічного інтерфейсу є групування компонент на графічному полі, яке формує в користувача чітке сприйняття візуальної ієрархії програмного засобу. Групування елементів відносно границь розміщення елементів в макеті (англ. Layout) графічного поля з однією основною робочою зоною показано на рис. 2.11, а). Так макет можна розділити на 5 частин: NORTH (північ), WEST (захід), CENTER (центр), EAST (схід), SOUTH (південь). Якщо у програмі неможливо явно виділити певну зону групування компонент відносно даного поділу, то це означає, що така зона в макеті відсутня і розділена між іншими зонами (рис. 2.11, б, в, г). Є чотири випадки: а) загальний; б) відсутня зона CENTER; в) відсутні зони WEST та EAST; г) відсутні зони WEST, CENTER, EAST.

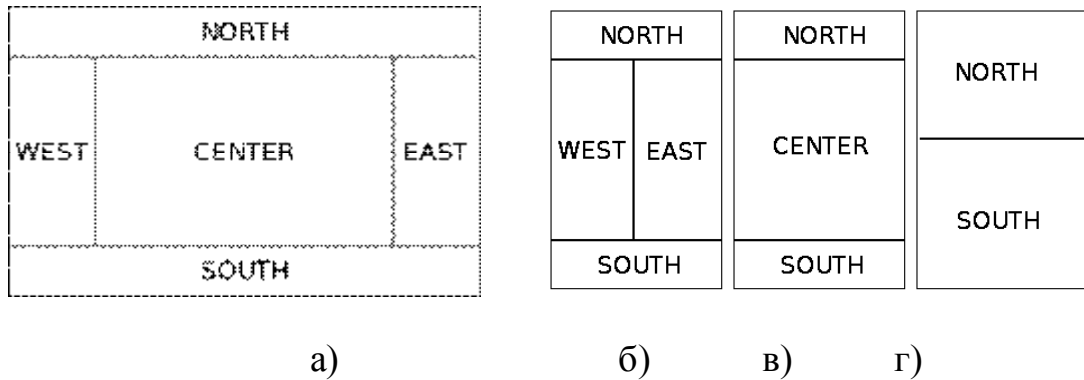


Рис. 2.11. Поділ розміщення компонент на графічному макеті програми

При цьому потрібно звернути увагу на те, що зони в свою чергу можуть бути поділені на підзони. Поділ на зони та підзони відбувається за використанням контейнерів, зокрема, `GroupBox` та `Panel`, використання розділювача `Splitter`. У програмах, які містять вкладки (`TabControl`), кожна вкладка розглядається як окремий макет і відповідно поділ відбувається окремо для кожної вкладки або якщо вкладки можна подати як набір таких контейнерів, як `GroupBox` та `Panel`, інакше програма не береться до уваги. Якщо зображення (`PictureBox`), текстова стрічка (`Label`) чи інші компоненти, що візуально перебувають у певній зоні, але не є основними компонентами, без яких програма втратить свою функціональність, то їм окрема зона може і не виділятися. Прокрутка вертикально та горизонтально не вважається окремою зоною, а є частиною відповідної зони, прокрутка в якій відбувається.

Якщо в прототипі програми є неосновні компоненти, яких не існує в пристрої, під який проектується програма і які не можна ефективно замінити існуючими, то вони вилучаються з поділу на зони та не розглядаються взагалі. Таким чином нове проектування вигляду програми відбуватиметься з втратою певного функціоналу. Тепер необхідно визначити зони за їх призначенням: 1) робоча зона – зона, в якій безпосередньо відбуваються зміни чи маніпулювання з об'єктом чи об'єктами, його відображення на дисплеї тощо (поля редагування тексту, зображень тощо); 2) функціональна

зона – це зона вибору операцій для дій над об’єктом чи зміни його властивостей, збереження даних, друк тощо; 3) інформаційна зона – це зона виводу інформації без можливості її зміни.

Відповідно на рис. 2.12 подано приклад форми програми з поділом на певні області, а задача полягає в проектуванні програми під мобільний пристрій. Область №2 повторює функції, які можна вибрати в області №1, тому області №1 і №2 є однією зоною NORTH. Але за необхідності розділення функціоналу програми області №1 і №2 можна розглянути як окремі зони – NORTH (верхню) та CENTER (центральну), але при цьому функції не повинні повторюватись, тобто наявна функція в одній зоні, не повинна дублюватись в іншій.. Область №5 не є зоною, так як це стрічка подій (StatusStrip) і такого компонента не існує на мобільному пристрої. Область №4 є робочою зоною CENTER при умові, що область №1 і №2 формують одну зону NORTH, або зоною SOUTH, якщо області №1 і №2 розглядаються як окремі зони. Область №3 в даному випадку не будемо розглядати як окрему зону. Таким чином в програмі є від 2 до 5 зон в залежності від умов розбиття, є 1 робоча зона та мінімум 1 функціональна.

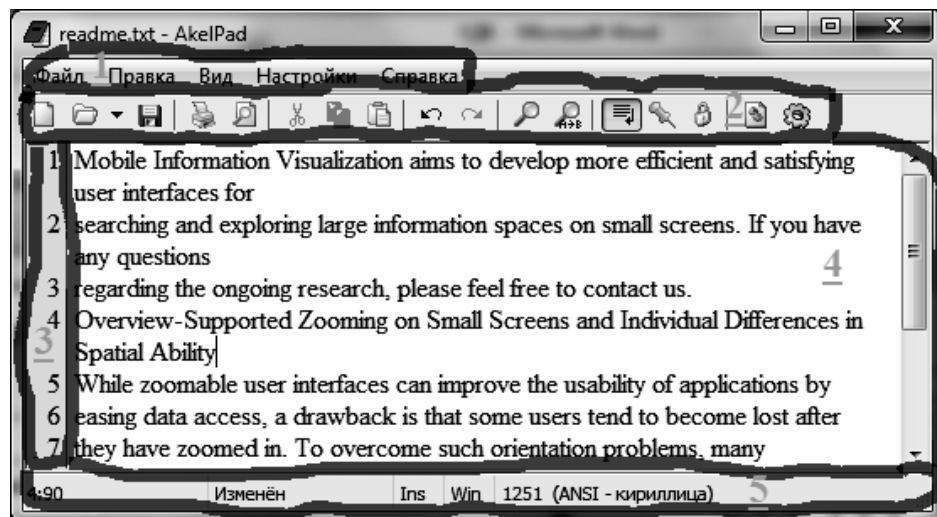


Рис. 2.12. Приклад поділу макету форми для програми AkelPad

Під час перетворення desktop-програми на мобільний пристрій, постає питання щодо можливості таких перетворень. Для цього необхідно розглянути [76] класифікацію програм щодо наявності певних зон та розміщенням робочої області (рис. 2.13). Безперечно доцільно розглянути й ті програми, які не мають явно виділених зон, мають декілька основних робочих полів та використовуються як форми заповнення певної інформації, зокрема, деякі програми документообігу, програми опитування, програми занесення контактів тощо. Такий тип буде називатись FormDocument і поділу на підзони не матиме.

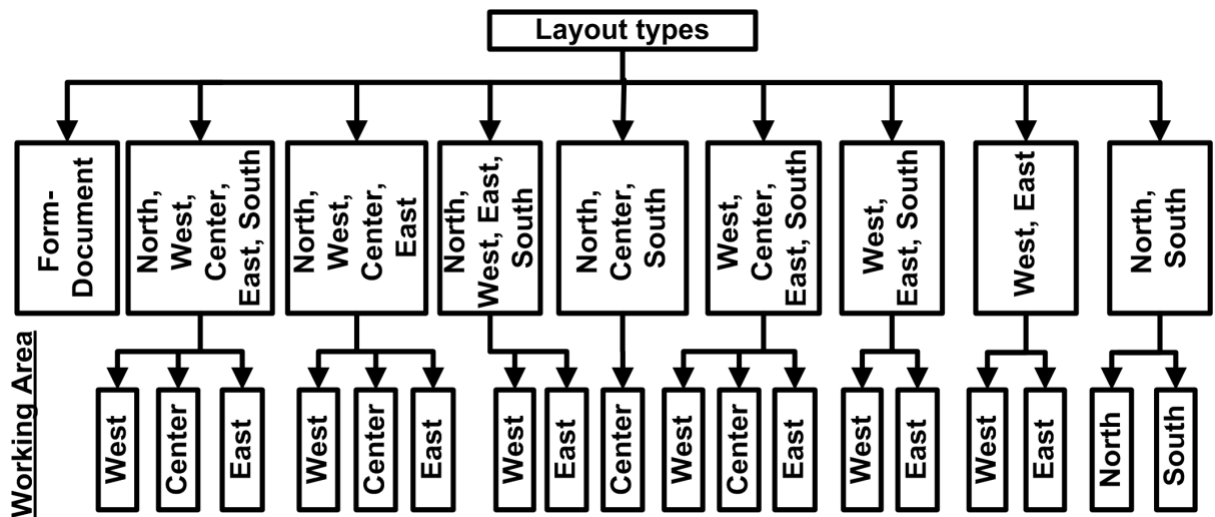


Рис. 2.13. Класифікаційний поділ макетів інтерфейсу користувача

Важливим фактором для програм з 1-ю основною робочою зоною також є висота (HEIGHT) та ширина (WIDTH) робочої зони відносно макету всієї програми. Враховуючи, що деякі програми з одною робочою областю дозволяють користувачеві змінювати компонування форми програми та розміри вікна програми, то необхідно розглядати макети форм програми тільки, якщо вікно програми має максимальні розміри щодо екрану відповідного пристрою. Класифікація подана на рис. 2.14.

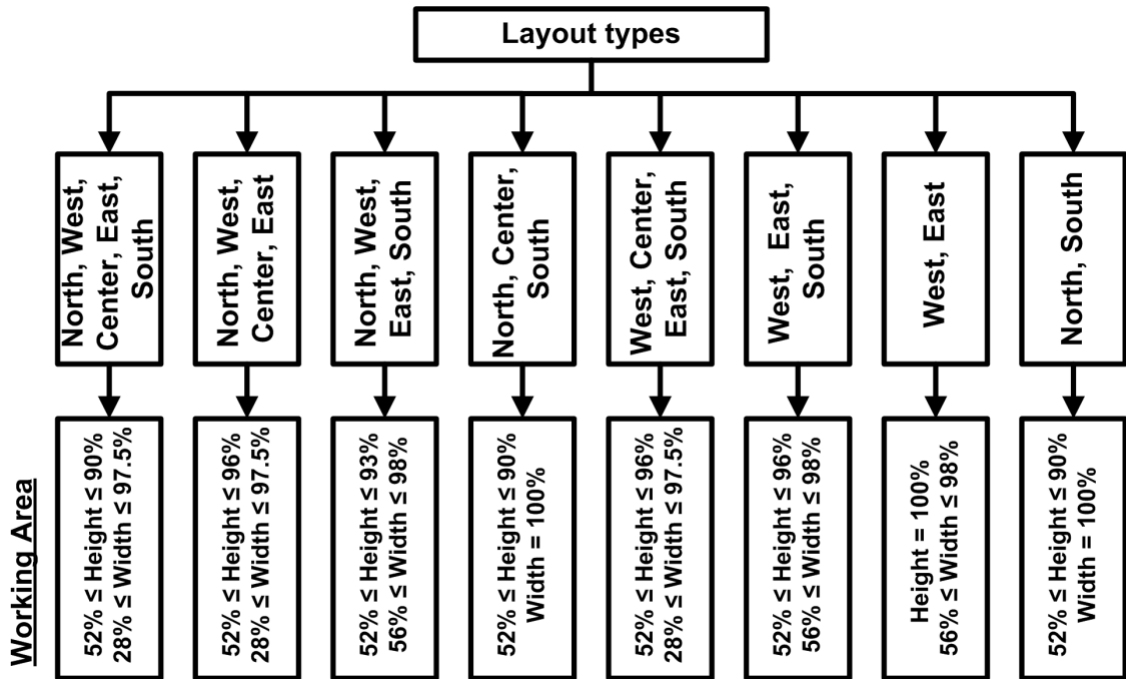


Рис. 2.14. Класифікаційний поділ макетів програм з пропорціями максимальних розмірів

Отже, подані класифікації дозволять розробити уніфіковані алгоритми перетворення інтерфейсу програмних засобів під час перенесення програмного продукту під інший екранний пристрій.

2.5. Метод оціночних характеристик засобів прототипування ГІК

Запропонований метод [77] оціночних характеристик для прийняття рішення при виборі програмного засобу прототипування інтерфейсів користувача. Базується на статистичних даних [127] застосування типів конструктивів користувацького інтерфейсу в існуючих САПР та на даних аналітичного огляду функціональних можливостей програмних продуктів прототипування інтерфейсів та настільних комп'ютерів та веб-систем.

Прототипи допомагають створювати високоякісні користувацькі інтерфейси. Для того, щоб швидко створювати прототипи зручного користувацького інтерфейсу, необхідно оцінити інструменти та методи

прототипування. Більшість оглядових статей про інструменти прототипування є описовими із загальними враженнями від першого використання, і не містять багатокритеріальності оцінок. Пропонований підхід [77] повинен допомогти здійснити вибір в ситуації обрання засобу ескізного інтерфейсного прототипування для конкретної програмістської організації з певною командою проектувальників інтерфейсів.

Дослідження фокусується на визначення переліку критеріїв для розгляду функціональності інструментів прототипування користувацького інтерфейсу. Критерії згруповані за категоріями. Ваги значення встановлюється для категорій і критеріїв. Ці вагові коефіцієнти встановлюють розподіл вплив на прийняття рішення. Зважений підхід до вирішення проблеми вибору інструментів для розробки користувацького інтерфейсу прототипів при досягненні умов: 1) різноманітність компонентів (C) \rightarrow max, 2) різноманітність генерації вихідного коду і форматів експорту (S) \rightarrow max, 3) різноманітність редагування прототипів і маніпуляції (E) \rightarrow max, 4) різноманітність платформ програмного забезпечення і використанню в команді (U) \rightarrow max, 5) Витрати, пов'язані з програмним забезпеченням (M) \rightarrow min.

За результатами статистичного опрацювання [127] GUI компонентів існуючих САПР (15 шт. в галузі машинобудування) було обрано коефіцієнти ваги (у відсотках до вартості табл. 2.2) для кореляції в категорії "Різноманітність компонентів (C)" (рис. 4.11). У підсумку було використано 98 різних типів елементів управління GUI (прості, як "кнопки", "іконки", "меню" тощо, та комбіновані, як "вибір кольору", "календар", "вибір дати", "калькулятор" тощо), які використовуються в дизайні і створенні прототипів користувацького інтерфейсу. Співвідношення кількостей появ елементів управління GUI представлені на рис. 4.11. Критерії категорії описані як:

$$C = [c_1, c_2, \dots, c_k], \text{ де } k = 98 \quad (2.13)$$

Таблиця 2.2.

Співвідношення ваг критеріїв для оцінки прототипування

№ <i>i</i>	Назва критерію	Умове позначення	Вага у категорії, %	Група категорій, %					
				Компоненти (C) → max	Програмний код та експорт (S) → max	Редагування (E) → max	Поширення, у команді (U) → max	Вартість ПЗ (M) → min	
1	Одноразова оплата	OP	5						70
2	Абонплата	PP							30
3	Локалізація	Lc	40						2
4	Платформа Windows Desktop	Dt							16
5	Платформа Web	Wb							16
6	Мобільна платформа	Mb							2
7	Платформа Apple Mac	Am							9
8	Платформа Linux/ Unix	LU							5
9	Платформа Java	Jv							2
10	Платформа Adobe AIR	AA							2
11	Клієнт-Сервер	CS							3
12	Надання доступу	Sh							10
13	Offline синхронізації	OS							4
14	Командне співробітництво	T							22
15	макет магазину	St							2
16	Версії	Ver							5
17	Прототипи для робочого столу	PD	15						10
18	Прототипи для веб	PW							10
19	Прототипи для мобільного	PM							7
20	Посилання екранів	Lnk							2
21	Інтерактивний прототип	IP							15
22	Компоненти ОС	N							3
23	Кросплатформні	Cr							7
24	Безпосереднє редагування	DD							10
25	Властивості редагування	EP							5
26	Активні зони	Hs							2

табл. 2.2 (продовження)

№ <i>i</i>	Назва критерію	Умове позначення	Вага у категорії, %	Група категорій, %				
				Компоненти (C) → max	Програмний код та експорт (S) → max	Редагування (E) → max	Поширення, у команді (U) → max	Компоненти (C) → max
27	Стиль як олівець	Pen	10			2		
28	Шари	Lr				3		
29	Шаблони	Tm				2		
30	Історія редагування	H				5		
31	Форматування тексту	Txt				2		
32	Вирівнювати	Al				2		
33	Самовирівнювання	Di				2		
34	Прив'язка до сітки	Sg				3		
35	Прив'язати до направляючих	Sgl				3		
36	Групування	Gr				3		
37	Редактор зображень	Ie				2		
38	Динамічний прото тип без кодування	Dc	10			10		
39	Створення вихідного коду	Src				48		
40	Експорт HTML5	htm				20		
41	Експорт PNG	png				5		
42	Експорт PDF	pdf				2		
43	Експорт JPG	jpg				2		
44	Створення специфікації	doc				7		
45	Створення карти сайту (Excel таблиця)	SM				2		
46	Створення презентацій	pre				2		
47	Друк	prt	2					
$j=$ 1.. ..98	Різноманітність компонентів $C = [c_1, c_2, \dots, c_{98}]$	Ci	30	\sum 100				
	Усього:		100	100	100	100	100	100

Для визначення загальної оцінки використовувалося середнє геометричне зважене:

$$\bar{x} = \left(\prod_{i=1}^n x_i^{\omega_i} \right)^{1/\sum_{i=1}^n \omega_i} = \exp \left(\frac{1}{\sum_{i=1}^n \omega_i} \sum_{i=1}^n \omega_i \ln x_i \right) \quad (2.14)$$

Тоді, для категорій наведених параметрів (табл. 2.2) та формули (2.14):
 $i = 1 \dots 47$; $n = 47$; $j = 1 \dots 98$;

$$X = [x_i] \cup [c_j] \quad (2.15)$$

підставивши ваги отримаємо формулу саме для визначення значень у категорії «М» (з врахуванням процентних співвідношень):

$$\bar{M} = \sqrt[50]{OP^{35} + PP^{15}} \quad (2.16)$$

$$\bar{C} = \sqrt[50]{\prod_{j=1}^{98} c_j^{\varphi_j}}, \text{ де } i = 1 \dots 98 ; \sum_{j=1}^{98} \varphi_j = \frac{100}{\min(\varphi_j)} \quad (2.17)$$

Аналогічний крок для категорії «Вихідний програмний код та формати експортування» (S):

$$\bar{S} = \sqrt[50]{Dc^5 + Src^{24} + htm^{10} + png^{2.5} + pdf + jpg + doc^{3.5} + SM + pre + prt} \quad (2.18)$$

Категорія «Редагування та маніпулювання (E)» :

$$\bar{E} = \sqrt[50]{\prod_{i=1}^{21} e_i^{f_i}} \quad (2.19)$$

де: $e_i = [PD, PW, PM, Lnk, IP, N, Cr, DD, EP, Hs, Pen, Lr, Tm, H, Txt, Al, Di, Sg, Sgl, Gr, Ie]$;

$$f_i = [5, 5, 3.5, 1, 7.5, 1.5, 3.5, 5, 2.5, 1, 1, 1.5, 1, \\ 2.5, 1, 1, 1, 1.5, 1.5, 1.5, 1]$$

$$i = 1 \dots 21;$$

Категорія «Поширення і використання в команді (U)» :

$$\bar{U} = \sqrt[50]{\prod_{i=1}^{14} u_i^{r_i}} \quad (2.20)$$

де: $u_i = [Lc, Dt, Wb, Mb, Am, LU, Jv, AA, CS, Sh, OS, T, St, Ver]$

$$r_i = [1, 8, 8, 1, 4.5, 2.5, 1, 1, 1.5, 5, 2, 11, 1, 2.5];$$

$$i = 1 \dots 14.$$

Таким чином, середнє арифметичне зважене для комплексного індексу, що можна застосувати для обрання програмного інструмента прототипування ескізів графічних інтерфейсів користувача визначається за формулою:

$$\bar{X} = \frac{\sum_{i=1}^n \omega_i \cdot x_i}{\sum_{i=1}^n \omega_i} \quad (2.21)$$

де $\omega_i = [0.3, 0.1, 0.15, 0.14, 0.05]$ – вагові коефіцієнти складових суми за категоріями. Функція зведеної оцінки тоді буде:

$$\bar{X} = \bar{C} \cdot 0.3 + \bar{S} \cdot 0.1 + \bar{E} \cdot 0.15 + \bar{U} \cdot 0.4 + \bar{M} \cdot 0.05 \quad (2.22)$$

Огляд функціональних можливостей понад 30 сучасних програмних засобів для створення ескізу, каркасу, макету, прототипу інтерфейсу користувача; огляд більше 98 типів конструктивних елементів графічного інтерфейсу користувача ПЗ – дозволили створити метод для прийняття рішення у виборі з альтернатив серед набору програмних інструментів для проектування користувацького інтерфейсу. Він охоплює ваги критеріїв у групах категорій факторів.

2.6. Метод оціночних характеристик мов програмування ІК

Метою цього дослідження [99, 100] є: розгляд програмних концепцій і специфікацій користувацьких інтерфейсів, які найбільше використовуються; визначення рівня їхньої готовності застосування в ПЗ САПР і аналізу їхніх можливостей; виокремлення та опис критеріїв для експертної оцінки мов декларування користувацьких інтерфейсів; створення порівняльної таблиці даних про функціональні та концептуальні можливості; визначення найвірогідніших технологій програмування користувацьких інтерфейсів для ПЗ САПР на основі створених вагових критеріїв для експертної оцінки.

Для проведення дослідження слід зосередити увагу на визначенні списку критеріїв [100] для експертної оцінки мов декларування. Кожний з критеріїв буде мати назву, яка дублюється скороченим поняттям та латинськими літерами. Критерії будуть згруповані за змістом у категорії. Для категорій та критеріїв будуть вказані вагові коефіцієнти важливості та розподілу впливу на прийняття рішення. При підборі списку спеціалізованих мов опису інтерфейсу користувача є зміст поділити їх на дві групи: 1) ті, що отримали широке застосування і продовжують розвиватися розробниками; 2) ті, що мають вузьку прикладну спрямованість або втратили актуальність з розвитку.

Поняття, які вкладаються у критерії експертної оцінки [100]:

1. Розширювані елементи користувацького інтерфейсу, навігації та контролю даних (eXtendable elements, X) – визначає можливість створення складніших піделементів візуальних елементів користувацького інтерфейсу з наслідуванням компонентних властивостей візуалізації та реакції на дії користувача.
2. Спецефекти елементів користувацького інтерфейсу (eFfects, F) – здатність до розширення базового відображення в залежності від визначених станів.

3. Візуальна деталізація (**V**isual detailing, V) – рівень графічної повноти та відповідності психологічній асоціативності при спостереженні користувачем.
4. Векторна графіка (**G**raphics, G) - наявність підтримки векторного формування графічного відображення елементів користувацьких інтерфейсів.
5. Самотестованість (**T**est, T) – визначається програмною методологією написання коду з можливістю автоматичного тестування поведінки ПЗ користувацьких інтерфейсів.
6. Автоналаштуваність (**U**totune, U) – внутрішня алгоритмічно закладена модель визначення параметрів відображення компонентів користувацького інтерфейсу в залежності від характеристик роздільної здатності, вирівнювання на формі, «притягування» до згруповано-взаємодіючих елементів користувацького інтерфейсу.
7. Ліцензія (**L**icense, L) – накладає умови щодо застосування, використання та розповсюдження технології опису користувацьких інтерфейсів.
8. Доступність платформи (**A**vaila**B**ility, B) – визначається рівнем розвитку і часом виходу технології в світ.
9. Серверна частина (**S**erver, S) – вказує на необхідність окремого стороннього ядра процесора обробки.
10. Об'єктно-орієнтована мова програмування (**O**OP**L**, O) – вказує на список мов, які можуть бути застосовані для програмування реакції на події від компонент інтерфейсу. Фактично визначає рівень повноти охоплення програмних інтерфейсів (Application Program Interface, API).
11. Середовище розробки (**E**nvironment, N) – визначає спеціалізовану прив'язку до інтегрованого середовища програмування та компіляції (при необхідності). Фактично визначає свободу вибору програміста з вибору інструмента для програмування.

12. Готовність впровадження технології (**Deployment, D**) – показник, який визначає долю інтеграції технології в існуючі середовища програмних розробок та бібліотек для ПЗ САПР.
13. Технічна сумісність (**Compatibility, C**) – відповідність стандартам, спосіб обробки даних (компільованість, інтерпретованість) тощо.
14. Макровизначення (**Macro, M**) – можливість створення наперед визначених контекстів для контентних адаптацій під користувацьку ситуацію.
15. Асоціативний рівень сприйняття програмної моделі; явна семантика (**Associative, A**) – рівень належності до парадигми мов програмування, інформаційної розмітки та семантики природних мов.
16. Модель подій інтерфейсу (діапазон контрольованих подій, ефективність моделі, **Events, E**) – кількісний параметр, який визначає діапазон подій, спричинених керуючими діями користувачем інтерфейсу, до яких здійснюється програмна прив'язка проектної обробки.
17. Документація (**specIfication, I**) – показник, який визначає рівень наявності описів лінгвістичних структур мови опису інтерфейсів користувача, різноманіття прикладів застосування, покрокових взірців описів тощо.

При оптимальному підході до розв'язання задач вибору інструментарію для проектування ПЗ САПР на рівні описів користувацьких інтерфейсів є зміст досягати чотирьох напрямків:

1. Якість користувацьких інтерфейсів проекту ПЗ САПР (**Quality, Q**) $\rightarrow \max$
2. Витрати на реалізацію користувацьких інтерфейсів ПЗ САПР (**exPenses, P**) $\rightarrow \min$
3. Час на реалізацію/модифікацію інтерфейсів ПЗ САПР (**pRoductive time, R**) $\rightarrow \min$
4. Складність реалізації /модифікації інтерфейсів ПЗ САПР (**complexitY, Y**) $\rightarrow \min$



Рис. 2.15. Вагові бали загальної оцінки у часткових категоріях

Таблиця 2.3.

Вагові розподіли у групових категоріях показників оцінки мов розмітки користувацьких інтерфейсів у застосуванні розробки ПЗ САПР

Показник, Умовне позначення		Категорії				Ваговий коефіцієнт суми в категорії
		(Quality, Q) → max	(exPenses, P) → min	(pRoductive time, R) → min	(complexitY, Y) → min	
1) eXtendable elements	(X)	25 %				0.45
2) eFfects	(F)	10 %				
3) Visual detailing	(V)	10 %				
4) vector Graphics	(G)	30 %				
5) self-Test	(T)	20 %				
6) aUtotune	(U)	5 %				
7) Licence	(L)		5 %			0.17
8) availaBility	(B)		20 %			
9) Server	(S)		20 %			
10) OOPL	(O)		35 %			
11) eNvironment	(N)		20 %			
12) Deployment	(D)		35 %			0.23
13) Compatibility	(C)		35 %			
14) Macro	(M)		30 %			
15) Associative	(A)			40 %		0.15
16) Events	(E)			15 %		
17) specIfication	(I)			45 %		
Загальна показник в категорії:		100%	100%	100%	100%	1

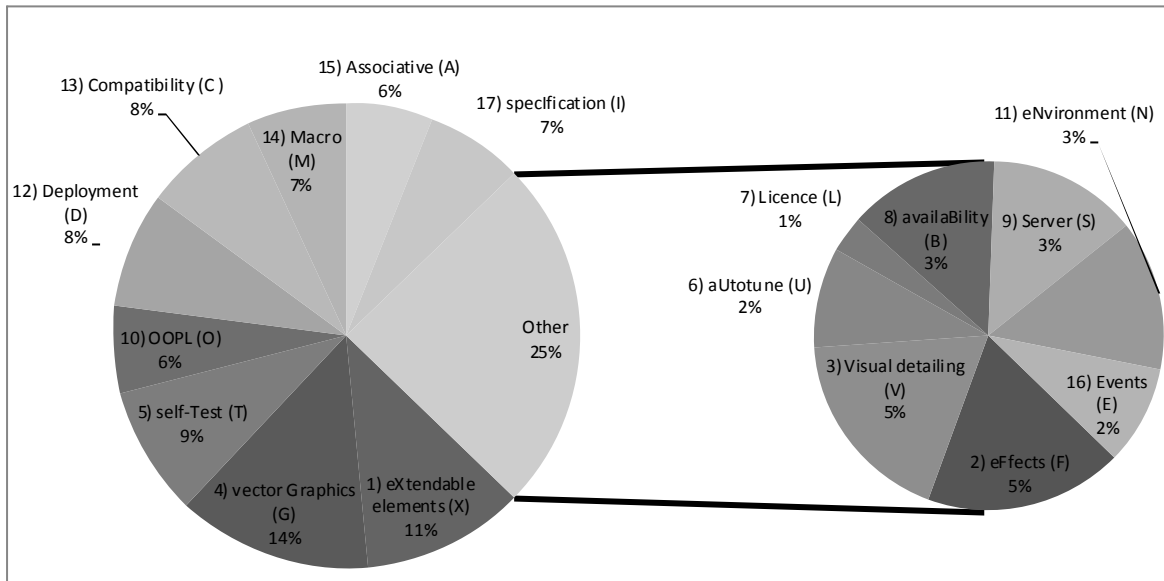


Рис. 2.16. Діаграми співвідношення показників із врахуванням ваг категорій оцінки мов декларування користувацьких інтерфейсів

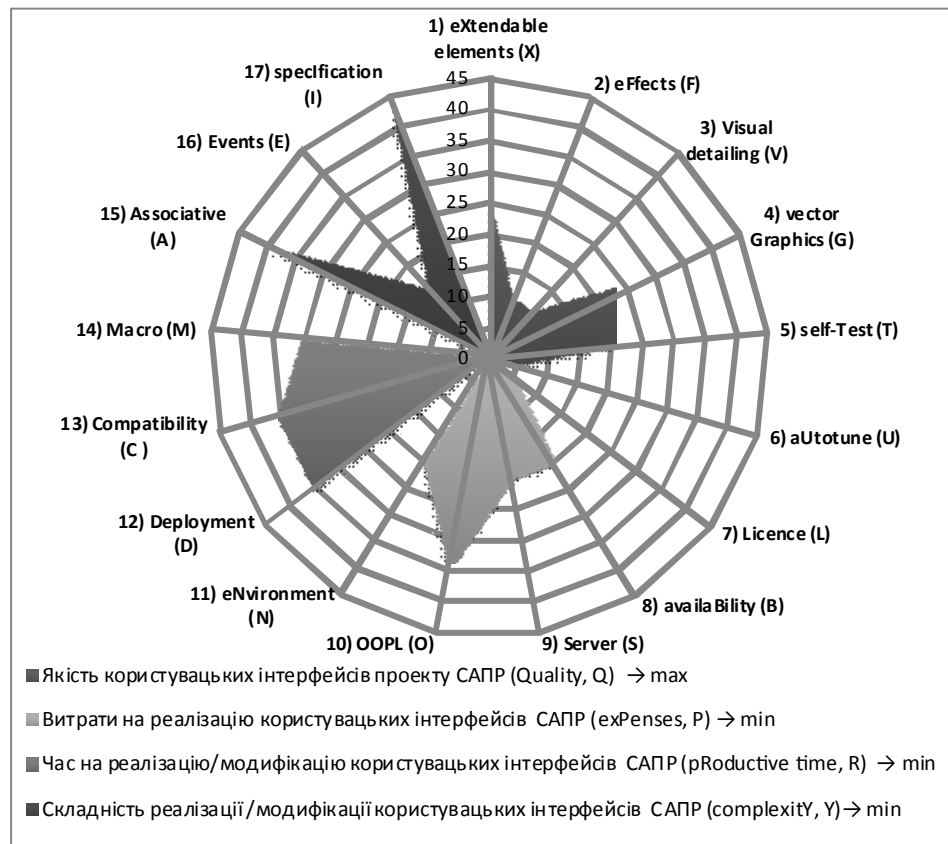


Рис. 2.17. Пелюсткова діаграма співвідношень показників у категоріях оцінки мов користувацьких інтерфейсів

Для визначення підсумкових показників [99, 100] варто скористатись середньо геометричним зваженням:

$$\bar{x} = \left(\prod_{i=1}^n x_i^{\omega_i} \right)^{1/\sum_{i=1}^n \omega_i} = \exp \left(\frac{1}{\sum_{i=1}^n \omega_i} \sum_{i=1}^n \omega_i \ln x_i \right) \quad (2.23)$$

Тоді для категорій з параметрами на основі табл. 2.3:

$i = 1 \dots 17$; $n = 17$ – перелік критеріїв оцінювання;

$$x_i = [X, F, V, G, T, U, L, B, S, O, N, D, C, M, A, E, I]; \quad (2.24)$$

Тоді підставивши значення у (2.23) отримуємо безпосередню формулу для визначення значення у категорії (Quality, Q), з врахуванням процентних співвідношень:

$$\bar{Q} = \sqrt[20]{X^5 + F^2 + V^2 + G^6 + T^4 + U} \quad (2.25)$$

та само і для категорії (exPenses, P) буде наступна формула:

$$\bar{P} = \sqrt[20]{L + B^4 + S^4 + O^7 + N^4} \quad (2.26)$$

аналогічно підставивши у формулу (2.23) отримуємо для категорії (pRoductive time, R) скорочену формулу:

$$\bar{R} = \sqrt[20]{D^7 + C^7 + M^6} \quad (2.27)$$

аналогічно для категорії (complexitY, Y):

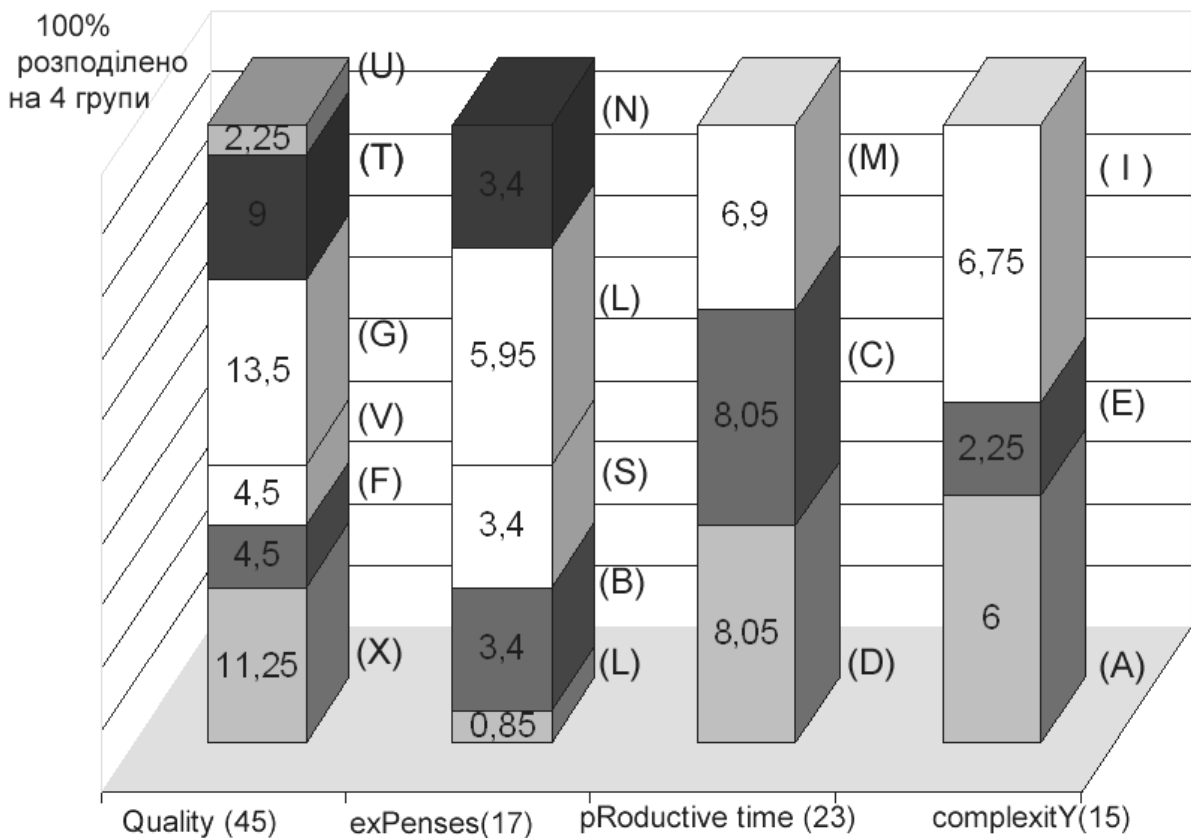
$$\bar{Y} = \sqrt[20]{A^8 + E^3 + I^9} \quad (2.28)$$

Отже середньо арифметичним зваженням для комплексного показника готовності мови програмування опису користувацького інтерфейсу буде:

$$\bar{x} = \frac{\sum_{i=1}^n \omega_i \cdot x_i}{\sum_{i=1}^n \omega_i} \quad (2.29)$$

де $\omega_i = [0.45, 0.17, 0.23, 0.15]$ – ваговий коефіцієнт суми в категорії

$$\bar{Z} = \bar{Q} \cdot 0.45 + \bar{P} \cdot 0.17 + \bar{R} \cdot 0.23 + \bar{Y} \cdot 0.15 \quad (2.30)$$



- 1) Якість користувацьких інтерфейсів проекту САПР (Quality, Q) → max
- 2) Витрати на реалізацію користувацьких інтерфейсів САПР (exPenses, P) → min
- 3) Час на реалізацію/модифікацію користувацьких інтерфейсів САПР (pRoductive time, R) → min
- 4) Складність реалізації /модифікації користувацьких інтерфейсів САПР (complexitY, Y)→ min

Рис. 2.18. Стовпцева діаграма процентних співвідношень показників у категоріях оцінки мов користувацьких інтерфейсів

Загальне враження про використання тих чи інших мов опису користувацького інтерфейсу з точки зору програмного впровадження у розробку ПЗ САПР виглядає наступним чином. Застосування, що декларуються провідними корпораціями, як Microsoft, Adobe, є найбільш розвинутими і за рівнями функціональності, і за рівнями якості отримуваних інтерфейсів, як для впровадження для локальних, так і для веб-програмних продуктів. Привабливість для розробників ПЗ САПР є високою. Але слід відзначити значну вартість серверних рішень для технологій та середовищ розробки. Крім цього, запропоновані технології трудомісткими при програмуванні. Інструмент XUL від групи Mozilla дає значні можливості програмістам писати сервіси близькі за функціональністю до настільних програм. Кожне з представлених рішень викликало створення програмної мови опису розмітки користувацького інтерфейсу, як окремої галузі. Однак, у цих випадках ці мови є частково платформно-середовищно орієнтовані, на відміну від необхідності концептуального опису інтерфейсів користувача. Тому саме мова UIML, яка не прив'язана до жодного з середовищ, але чітко розділяє абстрактні поняття структури інтерфейсів є найбільш перспективною до застосування. Конструкції мови UIML можуть використовуватися для трансляції в довільні інші описи, інші програмні середовища. І програміст-розробник ПЗ САПР може, як і раніше, спиратися на звичні технології програмування, доповнивши їх новими спеціалізованими описами користувацьких інтерфейсів.

2.7. Метод оцінювання інформаційних інструментів моніторингу функціонування ГІК ПЗ

Слід звернути увагу на визначенні списку критеріїв [115] для експертного оцінювання інструментів [109, 110, 111, 112, 113, 114] моніторингу функціонування ПЗ. Критерії групують за змістом у категорії.

Для категорій та критеріїв вказують вагові коефіцієнти важливості та розподілу впливу на прийняття рішення (Таблиця 2.2).

Таблиця 2.4.

Вагові розподіли у групових категоріях показників оцінки інструментів для моніторингу по функціональності

№ <i>i</i>	Показник	Категорії			Ваговий коефіцієнт суми категорії
		Можливості моніторингу (М)	Корисні функції (Ф)	Побічні показники (П)	
1	Монітор Подій	15%			0.5
2	Монітор API ОС	30%			
3	Монітор Кастомних .dll	20%			
4	Монітор MFC	10%			
5	Монітор .NET	15%			
6	Монітор ActiveX, COM	10%			
7	Ін'єкція .dll		20%		0.35
8	Моніторинг Кількох Процесів		15%		
9	Можливість зупинення процесу		5%		
10	Підтримка Сервісів		25%		
11	Підтримка Unicode		15%		
12	Вбудована Мова Сценаріїв		10%		
13	Підтримка Плагінів		10%		
14	Стабільність			40%	0.15
15	Швидко дія			35%	
16	Зручність			25%	
	Загальний показник у категорії	100%	100%	100%	1

Для визначення підсумкових показників варто скористатись середньо геометричним зваженим:

$$\bar{x} = \left(\frac{\sum_{i=1}^n \omega_i \cdot x_i}{\sum_{i=1}^n \omega_i} \right) \quad (2.31)$$

де $\omega_i = [0.5, 0.35, 0.15]$ – вагові коефіцієнти суми в категорії, $i = 1 \dots 17$;

Кількісні та якісні результати застосування цього методу наведені в розділі 4 у табл. 4.2, 4.3 та на рис. 4.12.

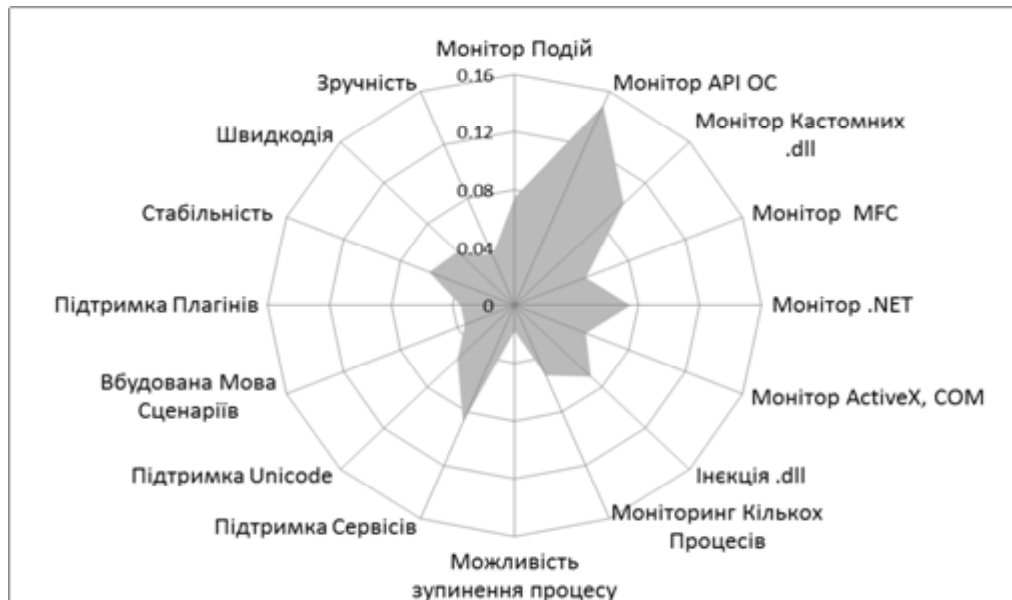


Рис. 2.19. Пелюсткова діаграма співвідношень показників у категоріях оцінювання інструментів для моніторингу

Запропонований підхід в методі містить критерії та показники за якими можна зважено визначити сумарну оціночну вагу для обрання програмістом чи тестером програмного забезпечення програмного засобу серед набору запропонованих засобів, які автоматизують процес моніторингу подій в інтерфейсних рівнях програмних продуктів. Це дає змогу скоротити час працівника на прийняття рішення.

2.8. Процес моніторингу інтерактивних взаємодій користувача із ГІК

На основі результатів (табл. 2.4, 4.2, 4.3; рис. 4.12) дослідження існуючих засобів моніторингу інформаційних подій в ПЗ та ОС, запропоновані схеми процесу виявлення та запису полій інтерфейсу користувача (рис. 2.20, 2.21):

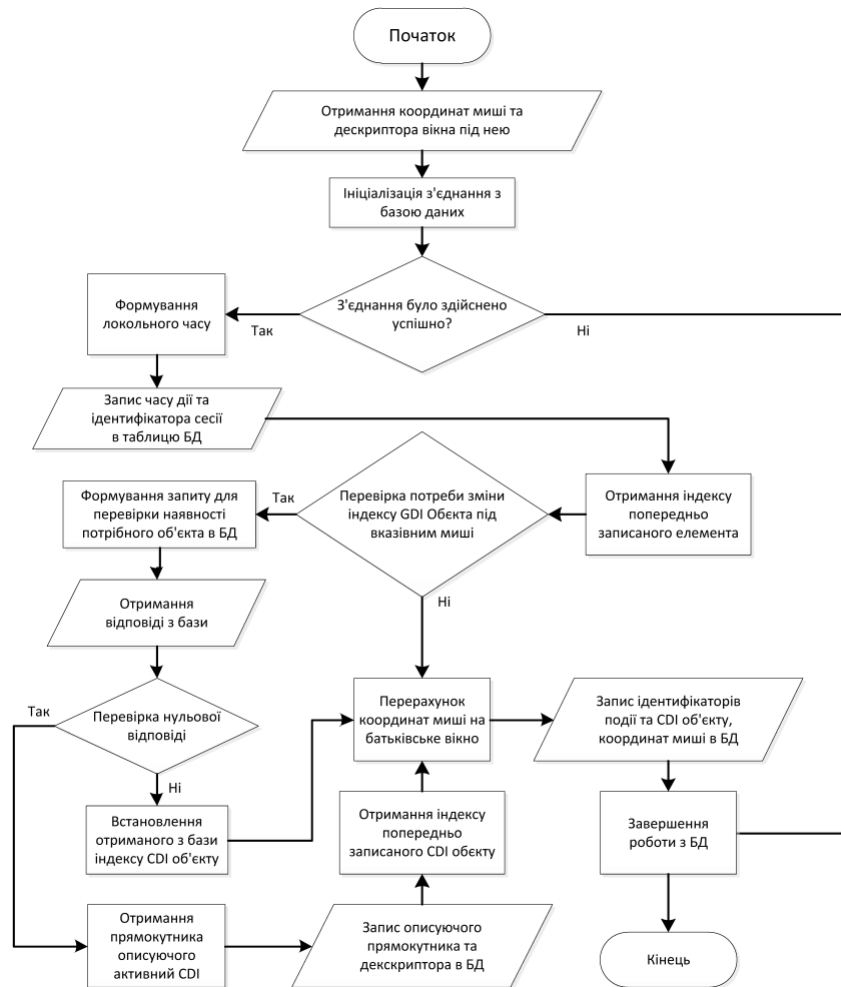


Рис. 2.20. Схема процесу ідентифікації та запису подій курсору «миші»

Для обчислення активності користувача над об'єктом ГІК використовується час, який «миша» провела на об'єктом та кількість активних дій. Адитивна формула має вигляд:

$$a = \omega_{move} \cdot \sum_{i=1}^k t_i + \omega_{action} \cdot \sum_{j=1}^n type_j \quad (2.32)$$

де ω_{move} та ω_{action} – коефіцієнти вагомості дій;

k та n – кількість відповідних дій; t_i – час дії;

$type_j$ – тип важливості дії.

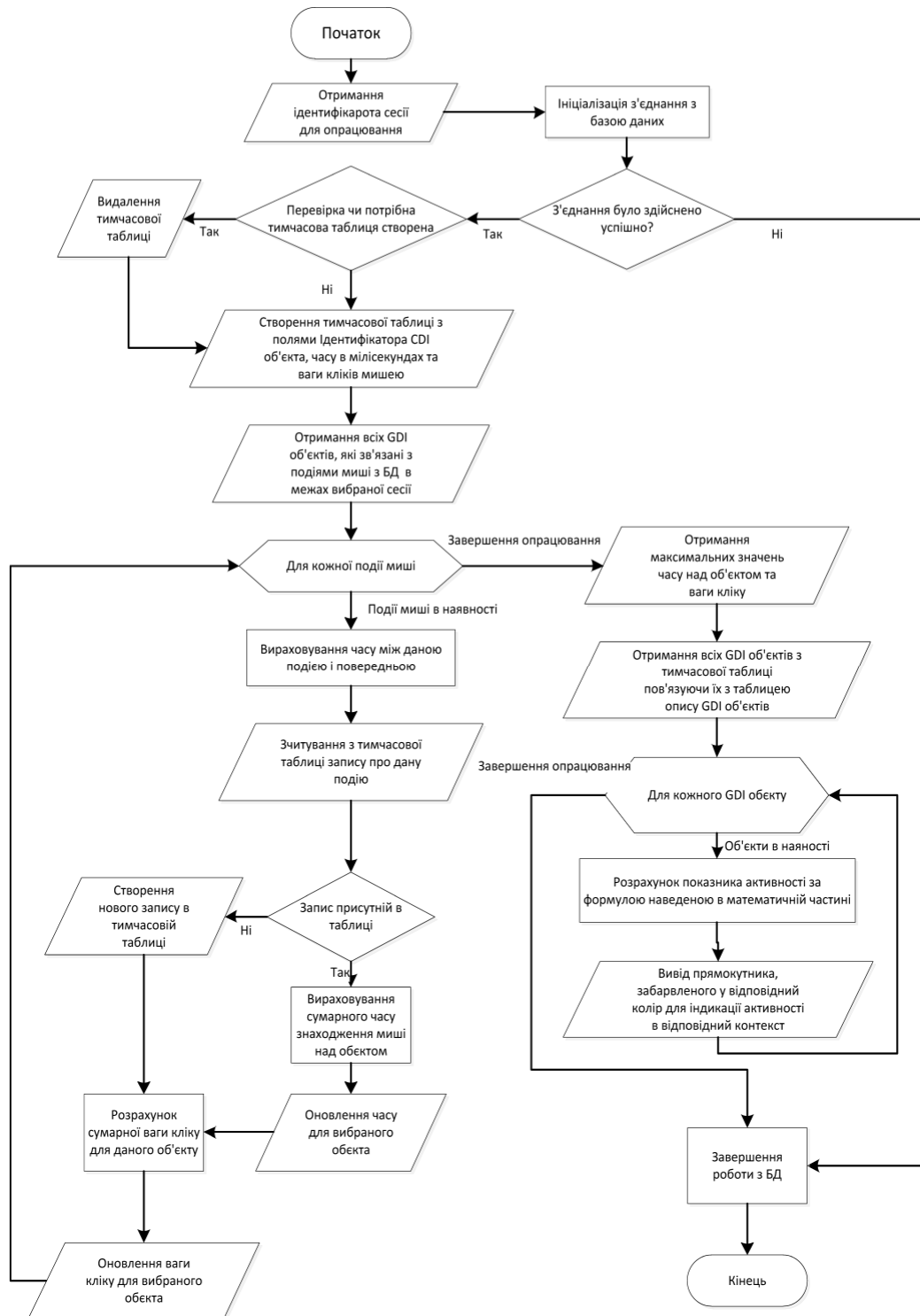


Рис. 2.21. Схема процесу аналізування подій від «миші»

Мультиплікативний спосіб оцінки, який зменшує ефект від хаотичних кліки та простоювання «миші» можна обчислити за формулою:

$$a = \left(\left(\omega_{move} \cdot \sum_{i=1}^k t_i \right) + d_{move} \right) \cdot \left(\left(\omega_{action} \cdot \sum_{j=1}^n type_j \right) + d_{action} \right) \quad (2.33)$$

де d_{move} та d_{action} – компенсаційні коефіцієнти вагомості дій.

Запропоновані схеми застосовуються підчас автоматизації процесів накопичення даних, які описані в підрозділах 2.2.2, 2.2.3, про інтерактивні взаємодії користувачів із інтерфейсами ПЗ. Така автоматизація підвищує точність обробки програмних подій ІК,

2.9. Процес перетворення графічних прототипів віконних форм інтерфейсу користувача в програмний код

Ескізне проектування прототипів віконних форм (в англ. термінології – prototype, mock-ups, wireframe, sketching) графічного інтерфейсу користувача (англ. GUI) як правило здійснюється або вручну, або в графічних редакторах растрових чи векторних зображень. Часто такі прототипи називаються каркасними прототипами, коли вони абстрагуються від деталей графічного дизайну. Кінцевим результатом роботи дизайнера GUI є растрове площинне зображення. Надалі таке зображення необхідно інформаційними технологіями опису, програмними кодами описати програмістам та зверстати компонування розміщення – це кропіткий та тривалий процес. Виникає потреба автоматизувати цей процес.

Інструментальний засіб реалізації цього процесу дозволяє розпізнавати чи ідентифікувати контури типових візуальних графічних конструктивних елементів (кнопки, меню, текстові поля...) як показано на рис. 2.22 та в табл. 2.5 користувацького інтерфейсу ПЗ з подальшим експортуванням у програмний код. Своєрідне розпізнавання зображення і переведення в електронно редаговані формати вихідних кодів.

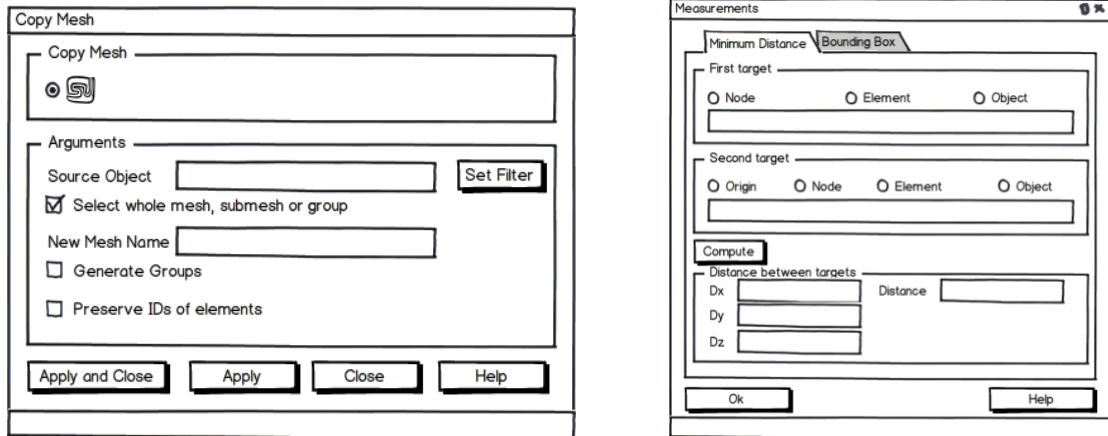


Рис. 2.22. Приклади ескізів графічних прототипів віконних форм для ідентифікації шаблонних конструктивних елементів ГІК

2.9.1. Основні кроки процесу ідентифікації конструктивних прототипів ГІК

Процес ідентифікації шаблонних компонентів графічного інтерфейсу користувача, дозволяє забезпечити автоматизацію програмування інтерфейсної складової ПЗ. У таких засобах особливо зацікавлені підприємства, що розробляють програмне забезпечення, які можуть в такий спосіб зменшити витрати на процес дизайну та верстання площинних віконних форм інтерфейсу за рахунок скорочення часу роботи програміста та збільшення його продуктивності.

Цей процес ідентифікації компонентів із графічного файлу є у складі комплексу інформаційних засобів автоматизованого генерування файлів розмітки і описів стилів користувацького інтерфейсу.

Процес складається з основних етапів:

- 1) накопичення репозиторію шаблонних компонентів графічного інтерфейсу користувача шляхом поступового виокремлення із віконних форм існуючих програмних продуктів для функціональних галузей; визначення кольорних спектрів візуальних форм, визначення графіків контурів;

2) накопичення репозиторію компонування площинного розміщення елементів інтерфейсу; геометричні моделі описів площинного поділу;

3) визначення характеристик візуальних геометричних параметрів шаблонних елементів інтерфейсів (габаритні розміри, позиції, параметри відображення тощо);

4) виконати якісний та кількісний аналіз взаємозв'язків між елементами інтерфейсу;

Усі елементи інтерфейсу, які необхідно розпізнати [116], відрізняються кольором свого забарвлення, від кольору фона віконної форми інтерфейсу, на якому вони розташовуються. На основі різниці кольорів сусідніх пікселів зображення, визначаються образи шаблонних елементів графічного інтерфейсу ПЗ. Для визначення різниці кольорів використовуються формули колірної відмінності. Контур елемента отриманий за допомогою розпізнавання різниці кольорів містить всю необхідну інформацію про форму об'єкта. Контур кодується послідовністю [118], що складається з комплексних чисел. На контурі фіксується точка, яка називається початковою точкою. Потім, контур обходиться (наприклад за годинниковою стрілкою), і кожен вектор зміщення записується комплексним числом $a + i \cdot b$, де a – це зсув точки по осі X, b – це зсув по осі Y. Зсув береться щодо попередньої точки.





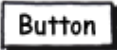


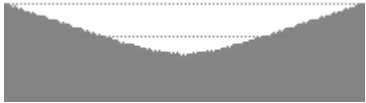
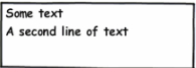
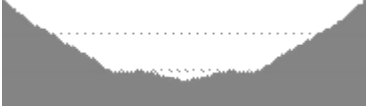
Також слід застосовувати автокореляційну функцію (АКФ) контуру, яка вважається характеристикою форми контура. Так, форми, близькі до кола мають рівномірні значення модуля АКФ. Сильно витягнуті в одному напрямку форми – мають провал в центральній частині АКФ. Форми, що переходять у самих себе при повороті, мають максимум АКФ у відповідному місці.

Спектр двовимірної автокореляційної функції фрагментів (табл. 2.5) зображення рівний енергетичному спектру зображення (спектральній густині) дає можливість працювати з контурами. Властивості АКФ: 1) АКФ

не залежить від вибору початкової точки контуру, зміна початкової точки призведе просто до зміни порядку сумарних елементів і не змінить суму; 2) модуль АКФ симетричний щодо центрального відліку; 3) якщо контур має будь-яку симетрію щодо повороту, то аналогічну симетрію має його АКФ; 4) АКФ контуру вважається характеристикою форми контуру. Нормована АКФ не залежить від масштабу, положення, обертання і вибору початкової точки контуру. [117]

Таблиця 2.5.

Приклади графіків автокореляційної функції для елементів графічного інтерфейсу користувача

Вигляд елемента інтерфейсу	Назва елемента інтерфейсу	Графік АКФ контуру елемента
	Checkbox (Чекбокс, прапорець)	
	Radio button (Радіокнопка)	
	Button (Кнопка)	
	Text Edit (Текстове поле)	
	Мемо (Текстова область)	

Існують методи, що вирішують завдання фільтрації контурів: оператор Кенні; оператор Собеля; оператор Лапласа; оператор Прюїтт; оператор Робертса.

Найчастіше використовується саме оператор Кенні, реалізований в програмній бібліотеці OpenCV.

2.9.2. Алгоритмізація процесу ідентифікації конструктивів КІ та синтезу

Схема алгоритмізації [119] процесу поетапної обробки графічного ескізу прототипу (растрове зображення) віконної форми графічного інтерфейсу користувача для виокремлення та ідентифікації змістовного конструктиву з подальшим синтезом текстового програмного коду для виокремлених конструктивів (рис. 2.23).

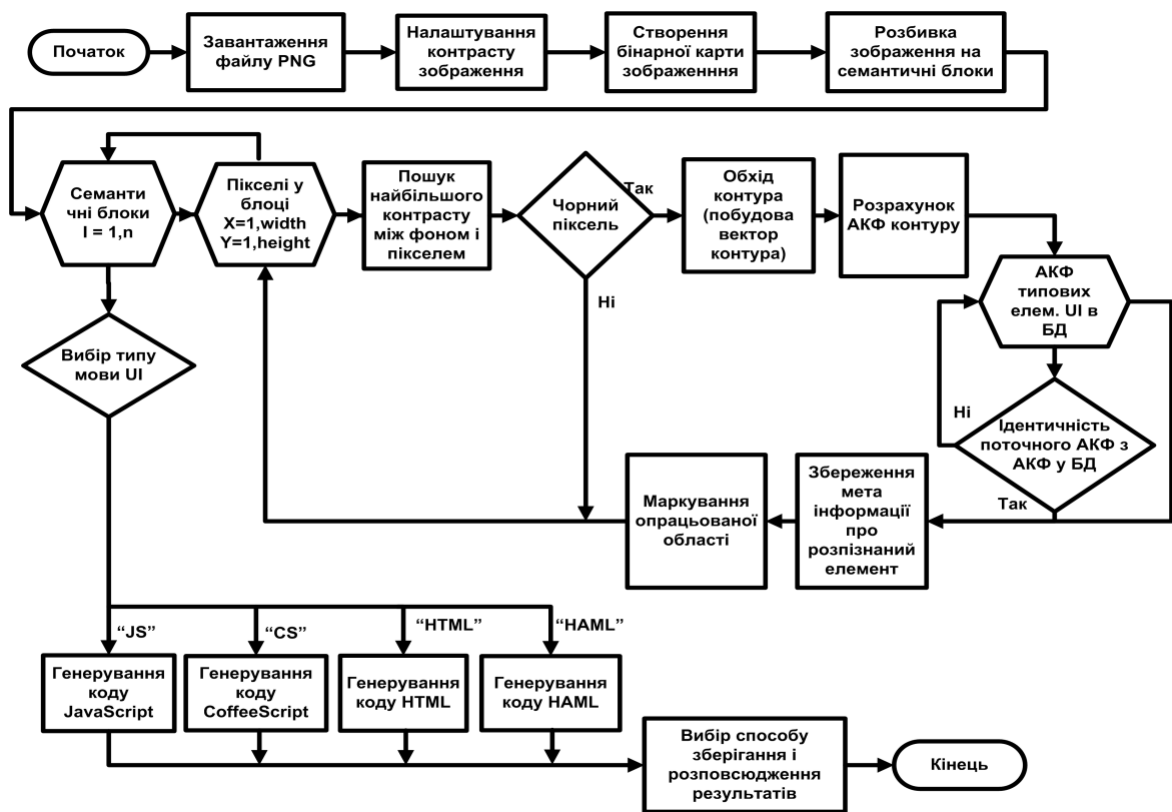


Рис. 2.23. Схема алгоритму процесу ідентифікації змістовних конструктивів ГІК та синтезу програмного коду

Запропонована схема дає змогу автоматизувати процес первинного ескізного програмування патернів ІК шляхом візуального розпізнавання контурів інтерфейсних конструктивів. Результати наведені у підрозділі 4.5.

2.10. Процес перепроєктування інтерфейсу

Схема процесу перепроєктування інтерфейсу користувача, складається з наступних етапів (рис. 2.24):

Етап 1. Для вхідного програмного продукту, який має графічний інтерфейс, необхідно виділити та розібрати атрибути, відокремити фрагменти програмного коду інтерфейсу. Аналізується синтаксис, щоб отримати ієрархічні дерева абстракції вхідного коду, структури та залежностей об'єктів ядра системи та інтерфейсу користувача в межах досліджуваної системи;

Етап 2. Відокремити фрагменти інтерфейсу користувача з решти системи, а також проаналізувати потоки даних від інтерфейсу користувача до функціоналу системи. Фрагменти отримуються виокремленням інтерфейс-коду від решти системи і представлення цих фрагментів коду у вигляді дерев структури конструктивів інтерфейсу (патерни програмного коду окремих конструктивів ІК). Якщо функції інтерфейсу у початковій системі одразу були програмно відокремлені в окремі модульні структури, то швидкість аналізування збільшується. Розгалуження дерева структур інтерфейсу створюють обмеження у вузлах. Ці обмеження враховуються при генеруванні інтерфейсу користувача у вихідний програмний код. Обмеження містять попередні залежності, узгодженості функцій звернення до структури даних, тощо.

Етап 3. Створення абстракції специфікації інтерфейсу користувача. Цей етап перетворює отримані фрагменти коду інтерфейсу в об'єкти з графічними характеристиками. Абстракції з потоків даних функціоналу системи інтегруються з даними про поведінку користувачів. З інтерфейс-фрагментів та інформації про потоки, абстракції відображають основні технічні характеристики візуального інтерфейсу на внутрішній мові опису абстракцій графічного інтерфейсу, як мови уявлення абстракцій

характеристик вхідного інтерфейсу інтерфейсу. Базова текстова структура абстракцій специфікації перетворюється у об'єктну специфікацію, яка використовує графічні конструкції для опису об'єктів, та їхні послідовності на екрані пристрою, яку користувач сприймає візуально.

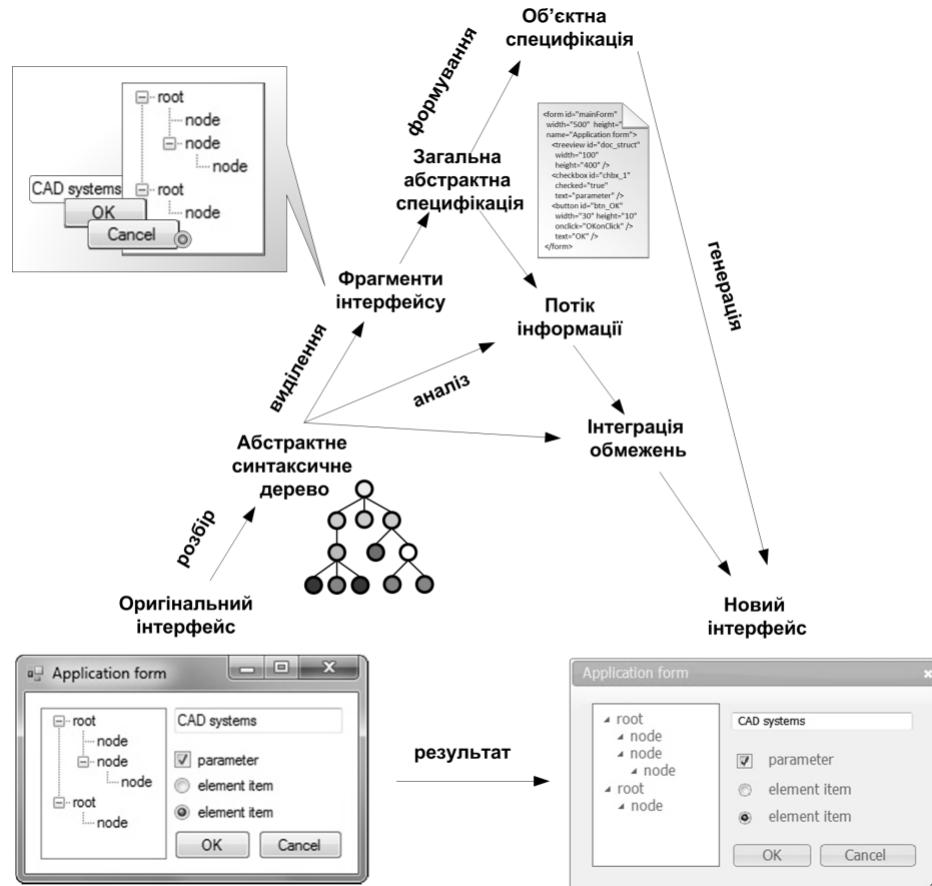


Рис. 2.24. Схема процесу автоматизації перепроєктування інтерфейсу з врахуванням метрик досвіду користування і програмним патернів

Перепроєктування розпочинається з вхідного інтерфейсу системи (лівий нижній кут на рис. 2.24) і триває до реалізації нового інтерфейсу на основі патернів програмних кодів із репозиторію. Важливим моментом цього процесу є виокремлення існуючої специфікації інтерфейсу з існуючої системи, поєднання її із накопиченими даними інтерактивних взаємодій користувачів і перетворення її у базову специфікацію опису абстракцій, а згодом перехід від базової специфікації до об'єктної специфікації, яку сприйматиме користувач на екрані монітору.

Етап 4. Генерування нового графічного інтерфейсу користувача. Автоматизовано генерується програмний код нового графічного інтерфейсу із специфікації опису абстракції з використанням шаблонних фрагментів генерування коду (патернів). Це особливо важливо для прототипування, для швидкого оцінювання майбутнього результату програмування.

2.11. Висновки до розділу

1. Визначено та формалізовано факторні параметри впливу користувачів на продуктивність роботи системи на основі тренуваності користувача на виконання сценаріїв взаємодій.
2. Показано формалізація процесу накопичення даних про досвід користувача та їхні кінематичні індекси, представлено геометричну інтерпретацію візуалізації накопичених даних. Надалі ці дані використовуються як кількісні коефіцієнти для синтезування чи перепроєктування поля макетного патерну компонування графічного інтерфейсу користувача.
3. Сформульовано критерії для оцінки якості засобів прототипування графічного інтерфейсу користувача та придатності спеціалізованих мов програмування та декларування інтерфейсів для зменшення часу впровадження у виробничий процес побудови інтерфейсів користувачів програмного забезпечення.
4. Запропоновано процес автоматизації перетворення візуальних графічних прототипів інтерфейсу користувача у програмний код, алгоритмізовано процес ідентифікації конструктивних елементів інтерфейсу користувача та синтезу програмного коду на основі патернів.
5. Модифіковано процес перепроєктування лінгвістичних описів інтерфейсу користувача з врахуванням метрик досвіду використання користувачами поля інтерфейсу.

Розділ 3. Реалізація засобів автоматизації процесів інтерфейсних побудов

У третьому розділі розкрито основні аспекти інформаційної, архітектурної, лінгвістичної, програмної реалізації засобів побудови (синтезу) інтерфейсів користувача на базі репозиторію патернів проектування програмних кодів. У процесі програмної реалізації застосовано сучасні засоби програмування та інформаційні технології. Показано інтеграційні засоби розподілених репозиторіїв статистичних даних інтерактивних взаємодій, патернів реінженірінгу кодів, макетів компонування компонентів ГІК та синтезу ІК із графічних прототипів.

Основні результати розділу опубліковано автором у працях [54, 77, 107, 108, 115, 119, 120, 124, 125, 126, 127, 143, 144, 145, 155]

3.1. Засоби для накопичення діалогових взаємодій

Підчас проектування інтерфейсів користувача програмного забезпечення та автоматизації цих процесів для програмістів, крім досліджень спектру елементів інтерфейсу [127] та візуальних і програмних характеристик [143] цих елементів, необхідно мати достовірну і різноманітну інформацію про часові та кількісні характеристики параметрів взаємодії кінцевих користувачів із вже готовими екранними інтерфейсами програмного забезпечення. Ці дані будуть використані для формування критеріїв підбору альтернативних форм елементів інтерфейсу для визначеного функціоналу програмного забезпечення на основі евристичних даних. Отже, розроблені інформаційні моделі репозиторію [143] елементів інтерфейсу необхідно доповнити статистичними даними їх використання [107, 108].

3.1.1. Підсистема програмного API доступу до інтерфейсних елементів та моніторингу подій

На рис. 3.1 наведено схему інформаційних зв'язків бази локального накопичення взаємодій користувача їх досліджуваним ПЗ.

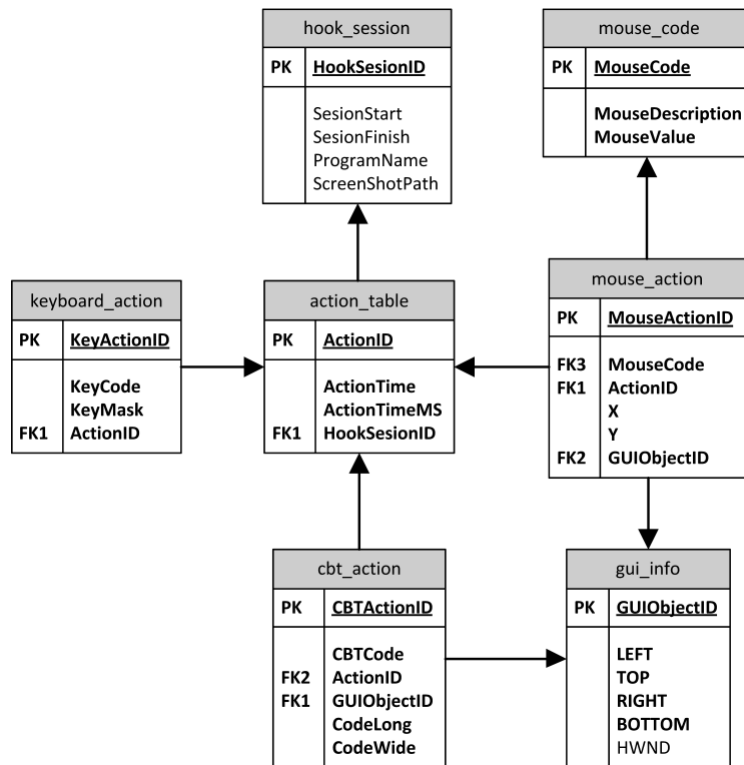


Рис. 3.1. Схема зв'язків локальної бази даних інтерактивних дій користувача

Схема містить інформаційну таблицю для зберігання даних про сесії моніторингу інтерактивних подій користувача (сутність `hook_session`), у якій вказуються: дані про час початку й завершення етапу моніторингу (`SessionStart`, `SessionFinish`), назву програмного засобу який підпадає моніторингу (`ProgramName`), унікальний ідентифікатор екранного відображення в операційній системі (`ScreenShotPath`). Це дозволяє одночасно спостерігати зі кількома програмними продуктами і як користувач їх по черзі активує. Окремо зберігаються дані подій від кнопок фізичної клавіатури, та маніпулятора «миші» у сутностях `keyboard_action`, `mouse_action`. Для кожної з подій зафіксується час, вид (сутність `cbt_action`)

та екранні координати й вид конструктивного елемента графічного інтерфейса користувача (сутність `gui_info`).

Ця схема локальної бази даних у подальшому інтегрується у програмне ядро підсистему накопичення інтерактивних дій користувача (рис. 3.9) і у підсистему візуалізації статистичних даних (рис. 3.10).

На рис. 3.2 показаний зовнішній вигляд віконної форми розробленого засобу для автоматизації процесу моніторингу подій користувача над полем інтерфейсу досліджуваної системи. Як видно з рис. 3.2 в операційній системі зараз завантажені 5 ПЗ і серед них є САПР AutoCAD. Можна розпочати процес моніторингу (пункт меню «Записати сценарій») за діями користувача під час його роботи із AutoCAD. Усі записані кліки «миші» потім можна візуально відобразити у режимі «Відтворити сценарій» (рис. 3.3). «Менеджер сценаріїв» (рис. 3.8) містить перелік доступних сесій записаних дій користувача із програмними продуктами.

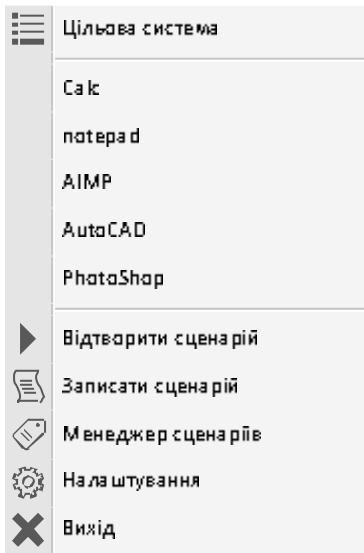


Рис. 3.2. Прототип вікна вибору досліджуваної системи

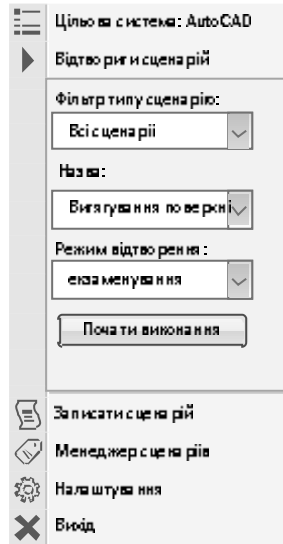


Рис. 3.3. Вибір сценарію для відтворення



Рис. 3.4. Інтерфейс програми під час відтворення

На рис. 3.5 показано меню «Налаштування», у якому можна вказати де будуть зберігатися дані інтерактивного моніторингу сценаріїв дій

користувача (рис. 3.4) та якими кольорами буде візуальне відтворення (рис. 3.6).



Рис. 3.5. Налаштування модулів моніторингу взаємодій

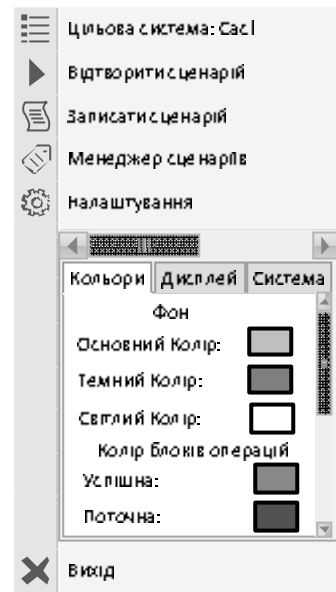


Рис. 3.6. Налаштування колірної палітри візуалізації

Створення сценаріїв автоматизованого моніторингу показано на рис. 3.7



Рис. 3.7. Вікно створення сценарію

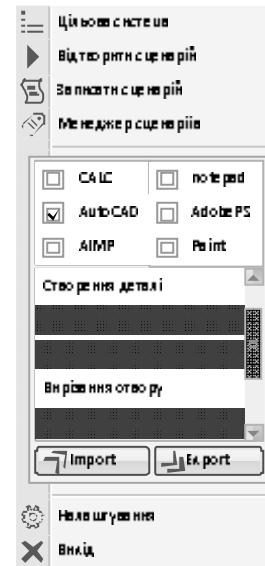


Рис. 3.8. Вікно менеджера сценаріїв

3.1.2. Підсистема локального накопичення даних

Підсистема збору та накопичення параметрів інформаційної діяльності [108] з користувацькими інтерфейсами програмного забезпечення виконує:

- автоматичне спостереження користувацьких взаємодій з інтерфейсами САПР (на схемі позначено Local CAD Tool);
- фіксування координат та вектору переміщень миші, пера планшету (на схемі позначено як hook);
- фіксування натискання, кліків, виділення, наведень активації тощо;
- фіксування часу на читання тексту, вивчення варіантів елементів керування інтерфейсом.

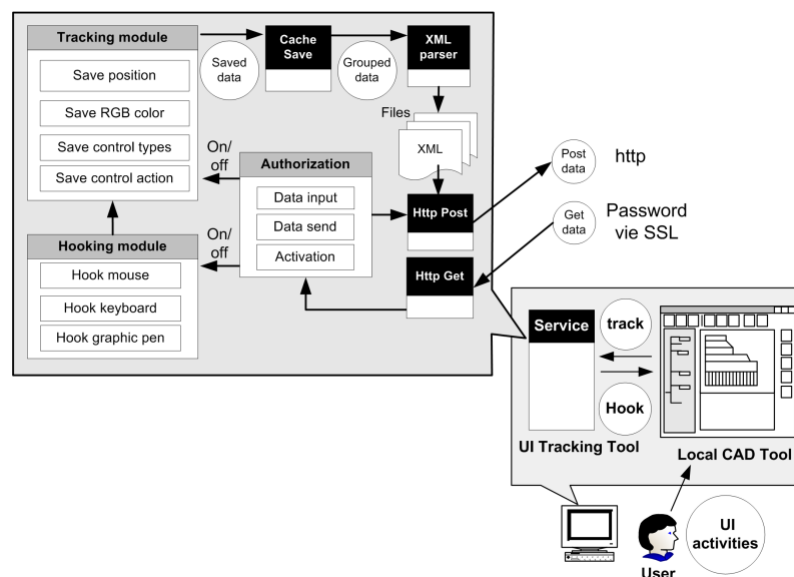


Рис. 3.9. Функціональна схема підсистеми відслідковування та накопичення даних про інтерактивну взаємодію користувача з ПЗ

Накопиченню підлягають і дані про колір пікселів над якими відбуваються інтерактивні дії користувача. Доступ до таких даних здійснюється через програмні дескриптори Application Program Interface (API) операційної системи. Дані зберігаються у локальному XML форматovanому кеші.

3.1.3. Підсистема візуалізації статистичної густини інтерактивної взаємодії

Схема (рис. 3.10) підсистеми візуалізації «температурних карт» густини інтерактивності дій користувача складається із модулів: оброблення даних з формату XML для подальшого перетворення патернами декларативного формату XSLT у формати веб-відображення мовами HTML, CSS, JavaScript. Графічні колірні карти густини активності користувача а також траєкторії переміщення «миші» візуалізуються мовою векторного малювання SVG.

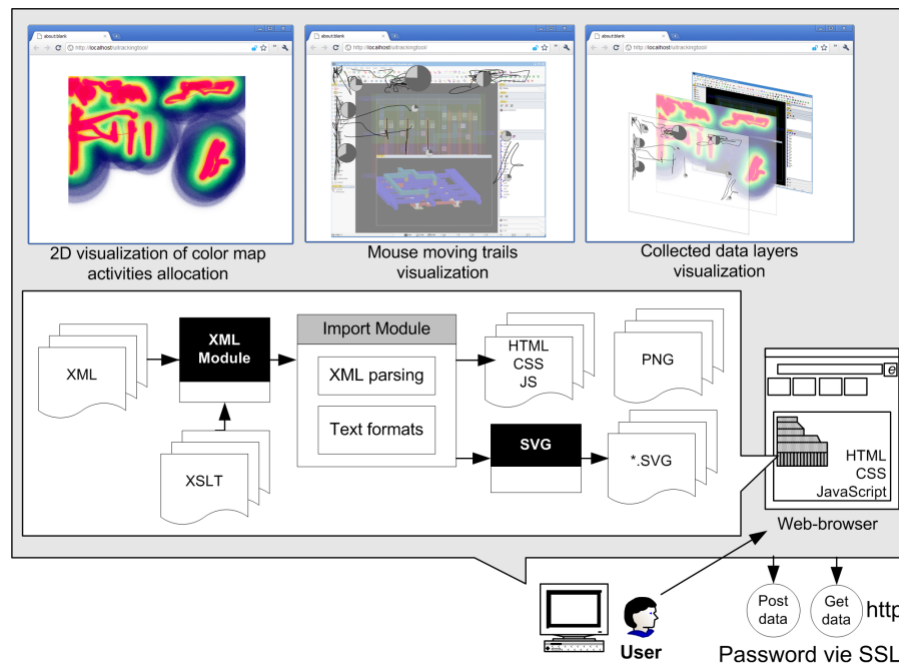


Рис. 3.10. Функціональна схема підсистеми візуалізації статистично оброблених даних інтерактивної взаємодії

3.1.4. Серверна частина

Головні можливості архітектури мережевого користувацького доступу:

- багатокористувацький доступ;
- стандартний веб-браузер та спеціальне програмне забезпечення на стороні користувача;

- міжплатформна сумісність.

Усі компоненти архітектури фізично розміщені на одному Internet-сервері, який працює як Web-сервер та обслуговує численних клієнтів системи. Для віддаленої роботи з відстежувачим процесором достатньо мати стандартний Web-браузер, як Internet Explorer, Firefox, Opera чи Google Chrome із Scalable Vector Graphics для кращої візуалізації. Увесь інформаційний обмін між користувачем та сервером реалізується через стандартний HTTP протокол та SSL.

Основні модулі:

- модуль адміністрування Web-порталу – забезпечує контроль доступу, управління обліковими даними користувачів та загальних компонентів порталу;
- модуль організації експериментів дозволяє керувати записами сесіями відслідковування користувацьких дій, керування записами сесій відслідковування користувацьких елементів інтерфейсу, статистичного моделювання;
- модуль захисту Web-системи проектування здійснює механізм ідентифікації усіх користувачів системою та ідентифікації відслідковувань;
- модуль експортування та імпортування проектних даних у систему та з системи передбачає функції підготовки проектної інформації у текстовому, XML, растровому, векторному представленні та поліграфічному форматуванні.

Архітектура системи [144] побудована на HTML5, Ajax, CSS, XML, PHP, MySQL, SVG, C# програмних технологіях. Загальна серверна архітектура запропонованої Інтернет/Інтранет-орієнтованого засобу представлена на рис. 3.11.

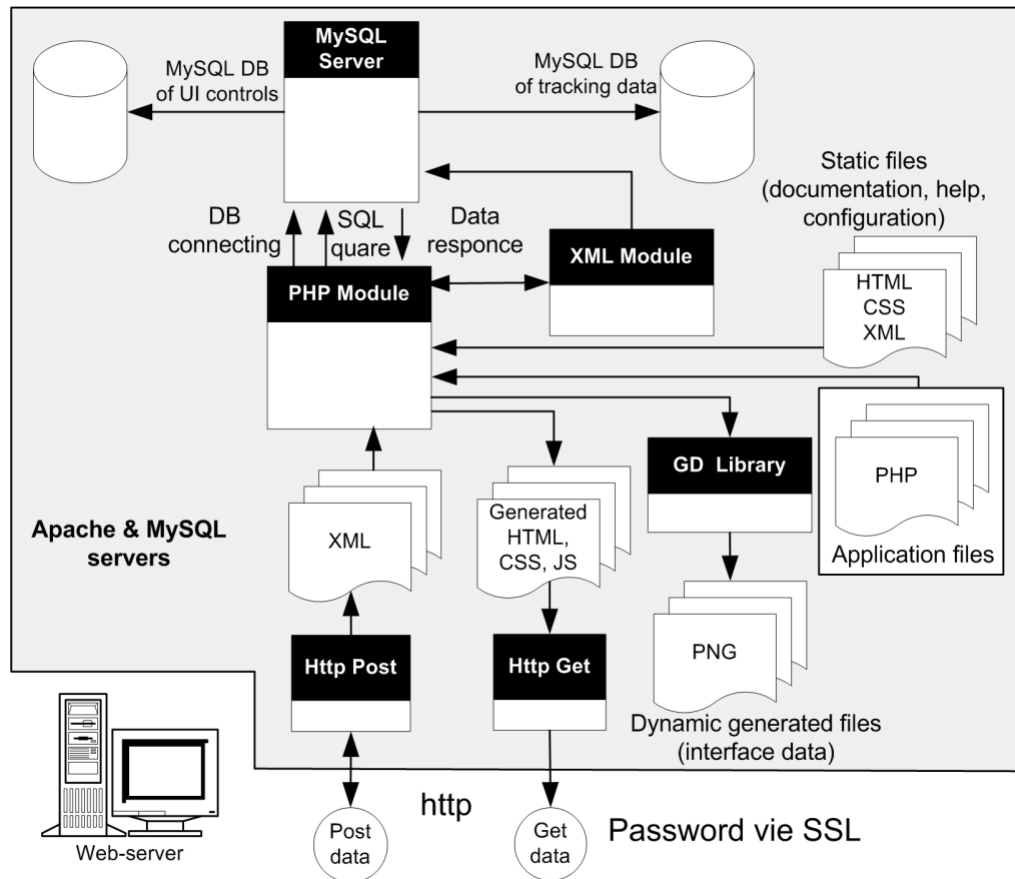


Рис. 3.11. Схема інформаційних складових серверної частини

Задачі та результати дослідження досвіду користування зберігаються у центральній реляційній базі даних (рис. 3.12). Записані локально в кеші дані відслідковування, пересилаються на сервер через POST-запит як XML дані. На сервері дані з XML розбираються в поля БД для подальшої обробки SQL (Structured Query Language) запитам.

Для керування базами даних застосована СУБД MySQL Server. Одна з баз містить дані про відслідковування подій в інтерфейсів (tracking data), друга база містить дані про патерни візуальних і програмних характеристик всього спектру конструктивних елементів ІК (UI controls). Серверний модуль PHP використаний для організації бізнес логіки (ярда сервера). Функції команд PHP вбудовуються у шаблони веб-сторінок HTML, CSS, JS для динамічного генерування відповіді сервера на запити. Бібліотека GD застосовується для динамічного генерування растрових зображень «колірних

карт» на основі даних із репозиторіїв UI controls та Tracking data. У подальшому для серверна складова інтегрована у інформаційну архітектуру, яка наведена на рис. 3.13.

3.1.5. Інформаційна модель збереження даних

Інформаційні сутності бази показані на рис. 3.12. Для відокремлення різних користувачів системи реалізована таблиця «Accounts», вона містить відомості про логіни, тип прав доступу користувача, паролі. Користувач засобу може групувати сесії простереження на інтерактивністю взаємодій, об'єднувати їх в «експерименти». Кожна сесія взаємодій здійснюється над певними віконними формами ПЗ, дані (розміри, тип, позиція на екрані, час відображення) про вікна зберігаються в таблиці «Window». Таблиця «Events» містить дані про тип, опис, патерн програмного відгуку на дії користувача ГІК. Таблиця «Action» містить часові, позиційні, колірні, конструктивні дані про інтерактивний вплив користувача на частину екрану.

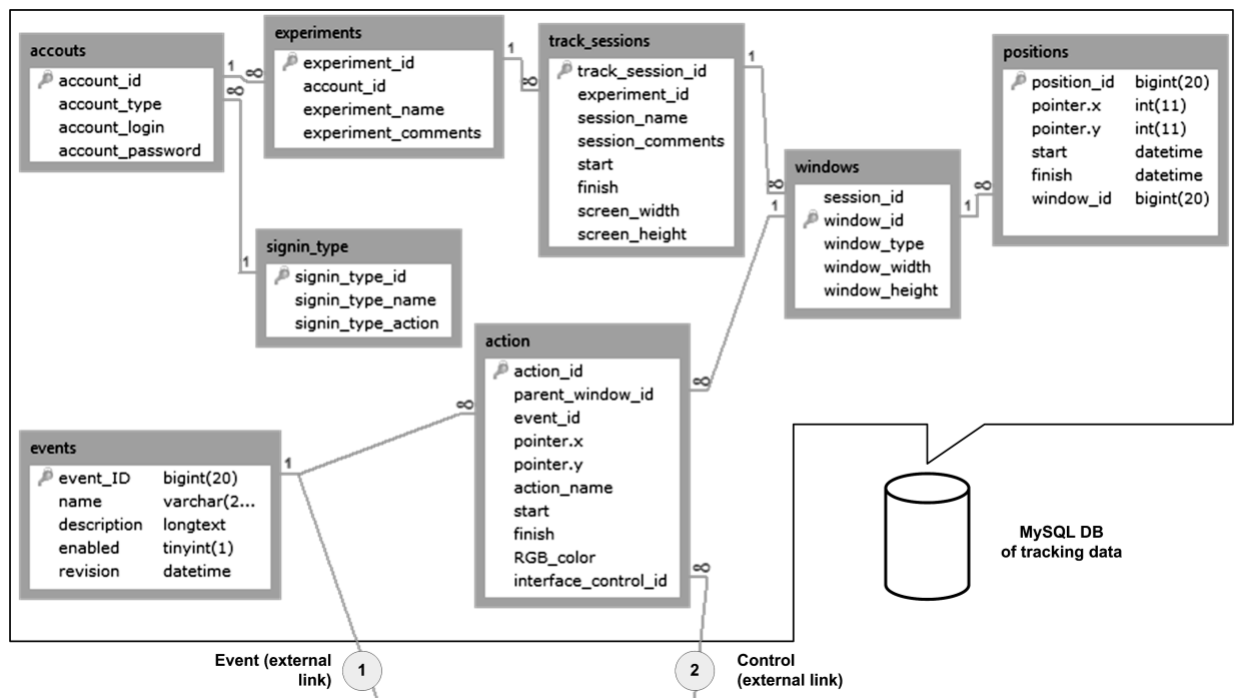


Рис. 3.12. Схема бази накопичення даних взаємодій користувача

3.1.6. Інформаційна інтеграція підсистем накопичення даних

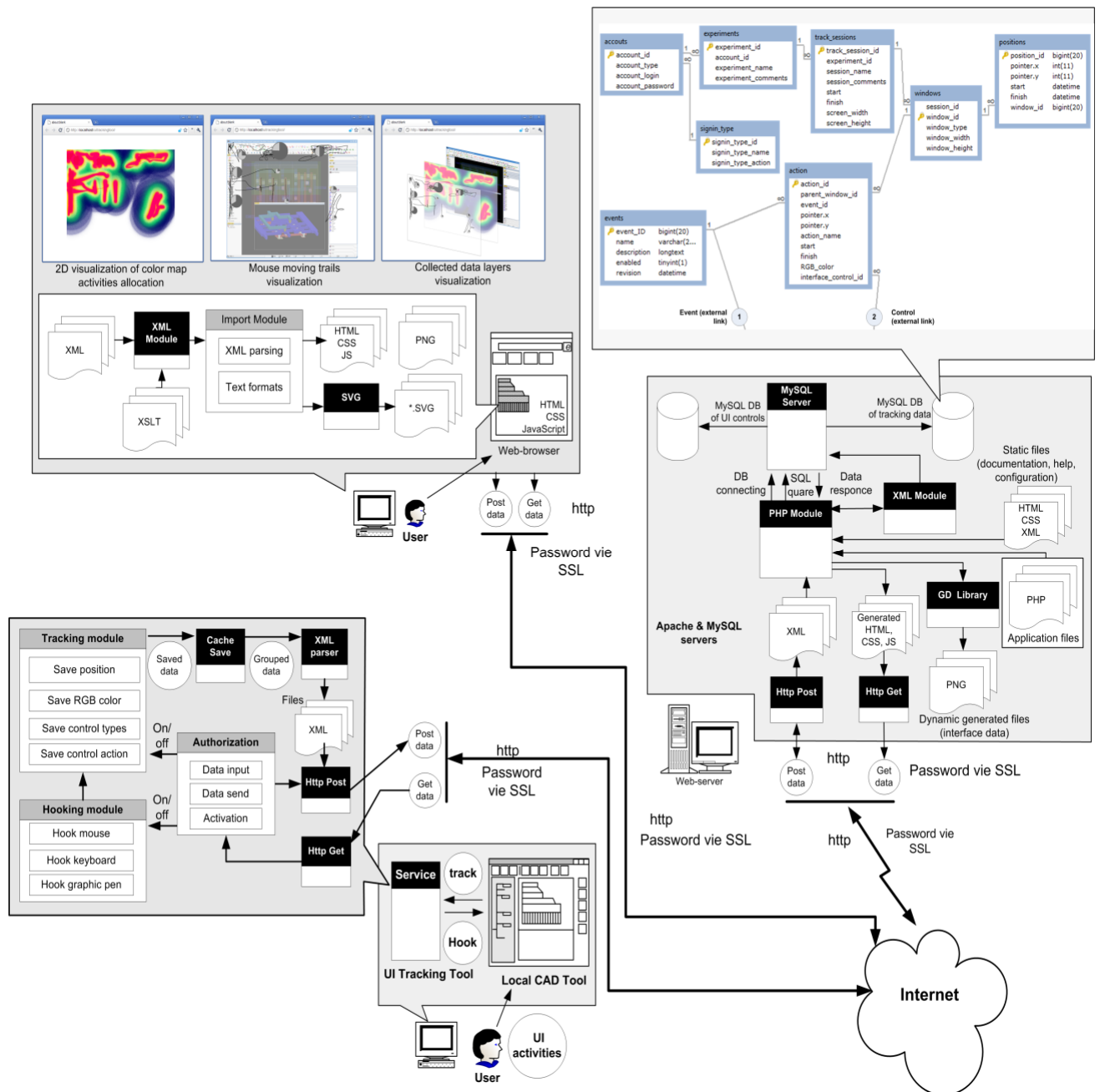


Рис. 3.13. Загальна схема архітектуру інформаційного засобу накопичення даних взаємодій користувача

Інформаційна архітектура (рис. 3.13) інтегрується за принципами розподілених систем, модулів й об'єднання відбувається мережевими технологіями та протоколами передавання даних.

3.2. Засоби формування репозиторію конструктивних елементів ІК

Загальна структура [126, 143, 145] інформаційного репозиторію для зберігання даних про конструктивні елементи інтерфейсу користувача програмного забезпечення показана на рис. 3.14.

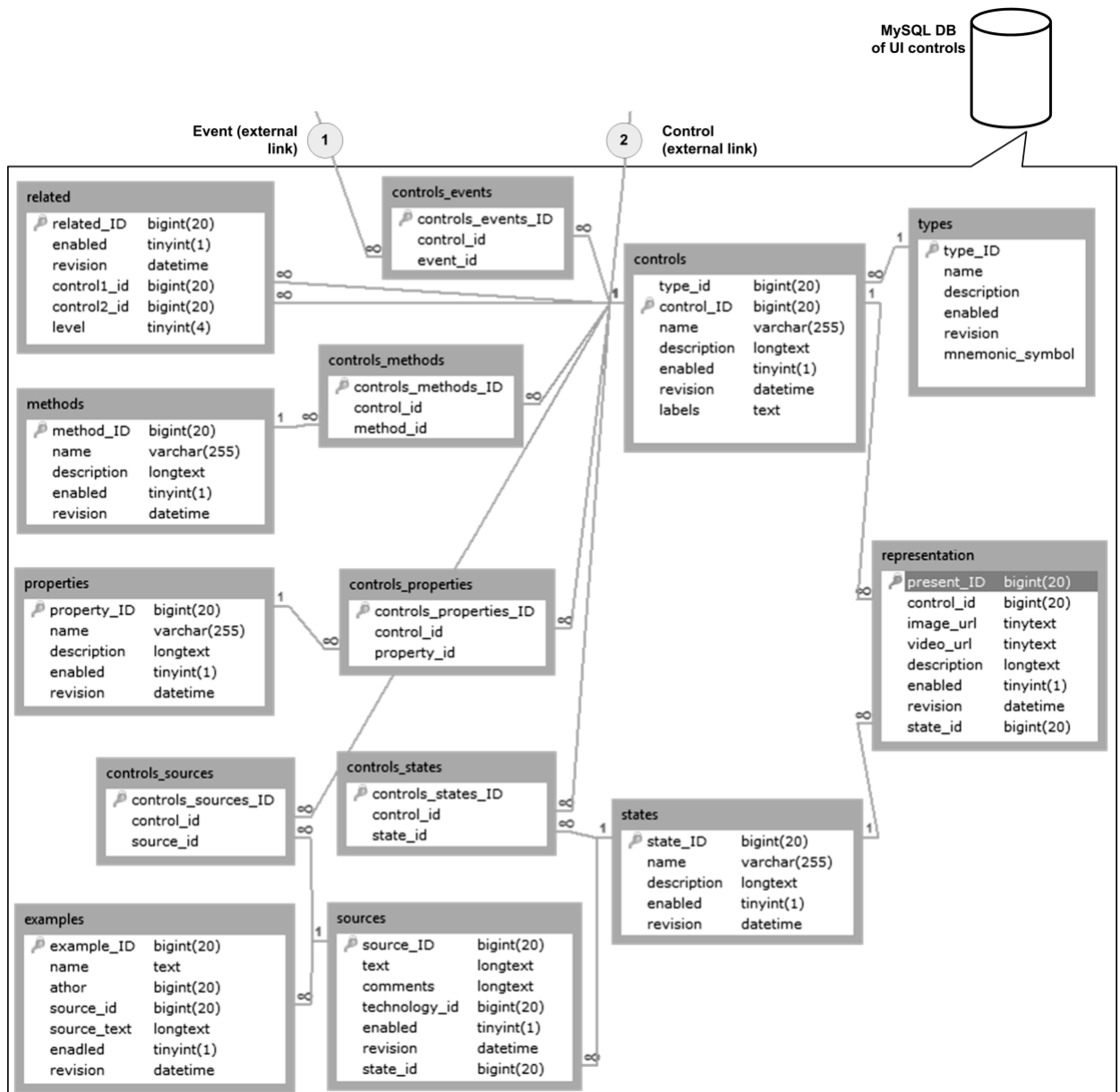


Рис. 3.14. Схема бази репозиторію конструктивних елементів ІК

Сховище [126, 143] компонентів GUI містить візуальне подання і функціональний зручний пошук альтернатив інтерфейсних реалізацій. Сховище забезпечує можливість вибору альтернатив вихідний код програми, порівняння варіантів вихідного коду на різних мовах програмування.

Інформаційна структура репозиторію містить наступні таблиці: «controls» – елементи управління (для списку всіх наявних модифікацій одного або декількох елементів управління); «control_types» – типізація (групування елементів управління в узагальнюючі підписки); «related_controls» взаємно пов'язані компоненти (вказання елементів управління, які можуть бути схожими один на одного, або як функціональна альтернатива, але з різних типів); «properties» – описує перелік усіх властивостей, які задають зовнішній вигляд елемента, його характеристики; «events» – описує перелік реакцій елемента керування на дії користувача чи системні події; «methods» – описує список всіх функцій, які можуть бути активовані елементом інтерфейсу чи реакції на події; «states» – стан (описує інформацію про візуальний стан управління до, під час, після взаємодії з користувачем); «control_presentation» – візуалізація презентація компонента (візуальна графіка, динамічні характеристики управління в різних станах або альтернативної графічної модифікації); «source» – вихідний код (код програми в різних доступних мовах програмування); «source_examples» – приклади програмного коду (готовий до використання, реалізації програмного коду у різних версіях програмного забезпечення). Усі таблиці мають поле «revision» («редакція») – версія даних, яке містить значення дати й часу. Це дозволяє зберігати кілька версій одних і тих самих даних. За необхідності, можна повернутися до більш ранніх версій даних або визначити, хто і коли зробив зміни у конкретній специфікації. Поле «Enabled» («доступний») у всіх таблицях визначає, чи можлива для використання будь-яка із складових інформаційної моделі бази даних. Поле

«Description» («опис») у таблицях містить технічні зауваження чи коментарі до складових інформаційної моделі.

3.3. Засіб автоматизації візуальної ідентифікації конструктивів прототипів ГІК

3.3.1. Архітектурне рішення функціонування засобу

Архітектура системи має клієнт-серверний підхід (рис. 3.15). Такий підхід є домінуючим патерном архітектурних шаблонів ПЗ у створенні розподілених мережових застосунків і передбачає взаємодію та обмін даними між ними.

Клієнтська частина відповідає за вибір графічного файлу на жорсткому диску комп'ютера, відправлення зображення на сервер, отримання даних про розпізнані контури елементів користувацького інтерфейсу та розпізнаний текст, представлення меню для вибору типу контуру користувачем, генерування програмного коду користувацького інтерфейсу та завантаження отриманого коду у вигляді файлу з розширенням HTML. Клієнтська частина системи реалізована на основі мови програмування JavaScript з використанням фреймворку AngularJS.

Серверна частина відповідає за отримання графічного файлу з ескізом графічного інтерфейсу користувача, розпізнавання контурів елементів інтерфейсу користувача, розпізнавання тексту та пересилки інформації з результатами розпізнавання на клієнтську частину. Серверна частина реалізована на основі програмної платформи NodeJS з використанням фреймворку Express. Взаємодія клієнтської та серверної частини відбувається за допомогою JavaScript бібліотеки Socket.IO, що підключається на обох частинах системи. Спрощена схема алгоритму функціонування серверної частина на рис. 3.16

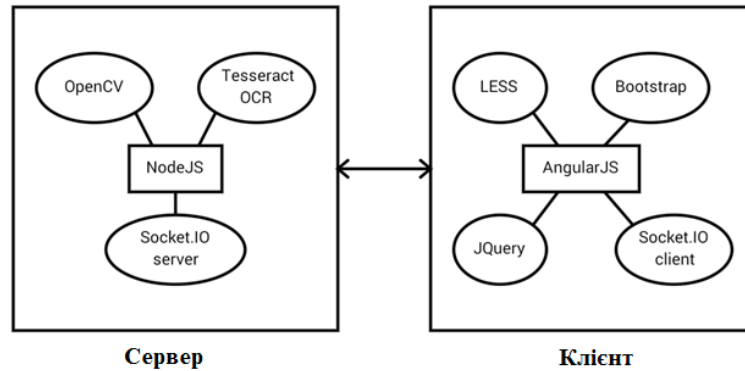


Рис. 3.15. Схема технологічних зв'язків інформаційних технологій

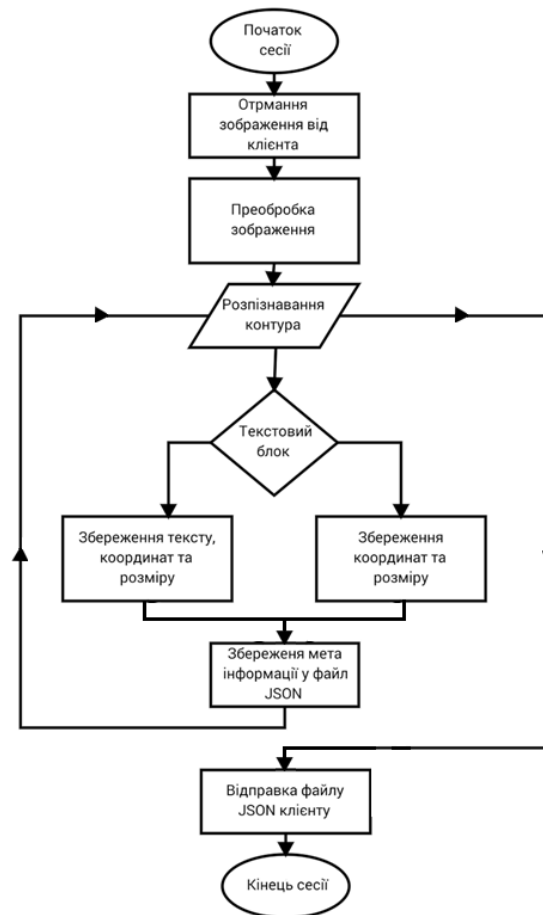


Рис. 3.16. Схема алгоритму функціонування серверного модуля

Схема (рис. 3.17) об'єктно-орієнтованих класів містить сутності про контур конструктиву «ElementContour» ІК (координати позиції та габаритні розміри), сутності про контури «TextContour» текстових зон (координати позиції, габаритні розміри, стрічки символів), патерн програмного коду

«Template» для ідентифікованого конструктиву ІК. Клас «UI_Element» відповідає за сутність опрацьованих під час розпізнавання конструктивів ІК. Клас «Canvas» візуалізовує візуальне підсвічування відокремлених зон з графічного ескізу ІК. Клас «Generator» для продукування програмного коду опису ІК під розпізнані площинно-габаритні розміри з растрового ескізу.

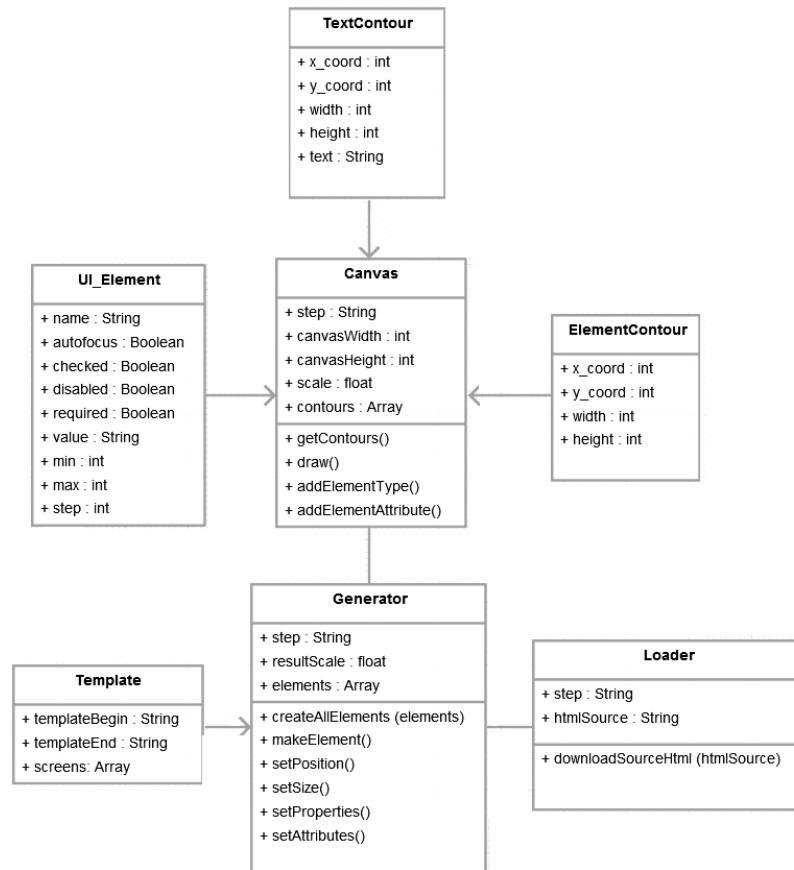


Рис. 3.17. Схема класів ООП клієнтського модуля

3.3.2. Ролі користувачів системи ідентифікації компонентів ГІК

У межах організаційного забезпечення система передбачає дві ролі користувачів: користувач та адміністратор. Можливості взаємодії користувачів в залежності від ролі розподілені наступним чином (рис. 3.18):

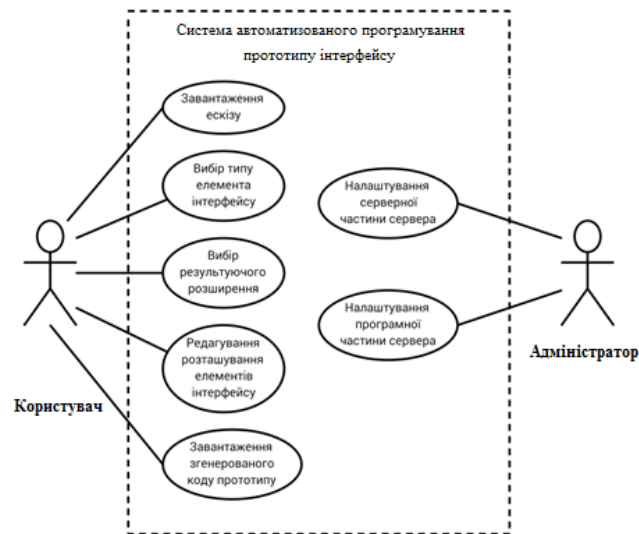


Рис. 3.18. Use case діаграма ролей користувачів системи

До можливостей користувача без прав адміністратора відносяться:

- Можливість завантаження зображення ескізу інтерфейсу
- Вибір типів елементів інтерфейсу
- Вибір результуючого розширення екрану прототипу
- Редагування розташування елементів інтерфейсу
- Завантаження згенерованого програмного коду прототипу інтерфейсу користувача

Користувач з правами адміністратора має аналогічні можливості взаємодії з системою, що й користувач без прав адміністратора, але з наступними розширеннями:

- Доступ до налаштувань апаратної частини сервера через адміністративну панель хостинг провайдера.
- Доступ до налаштувань програмної частини сервера через налаштування в операційній системі сервера.

Розмежування прав користувачів системи відбувається за допомогою надання користувачам з правами адміністратора імен користувачів та відповідних їм паролів до адміністративної панелі хостинг провайдера та операційної системи, що встановлена на серверному комп'ютері.

3.4. Інтеграція засобів накопичення і метрик діалогових взаємодій з даними репозиторіїв патернів

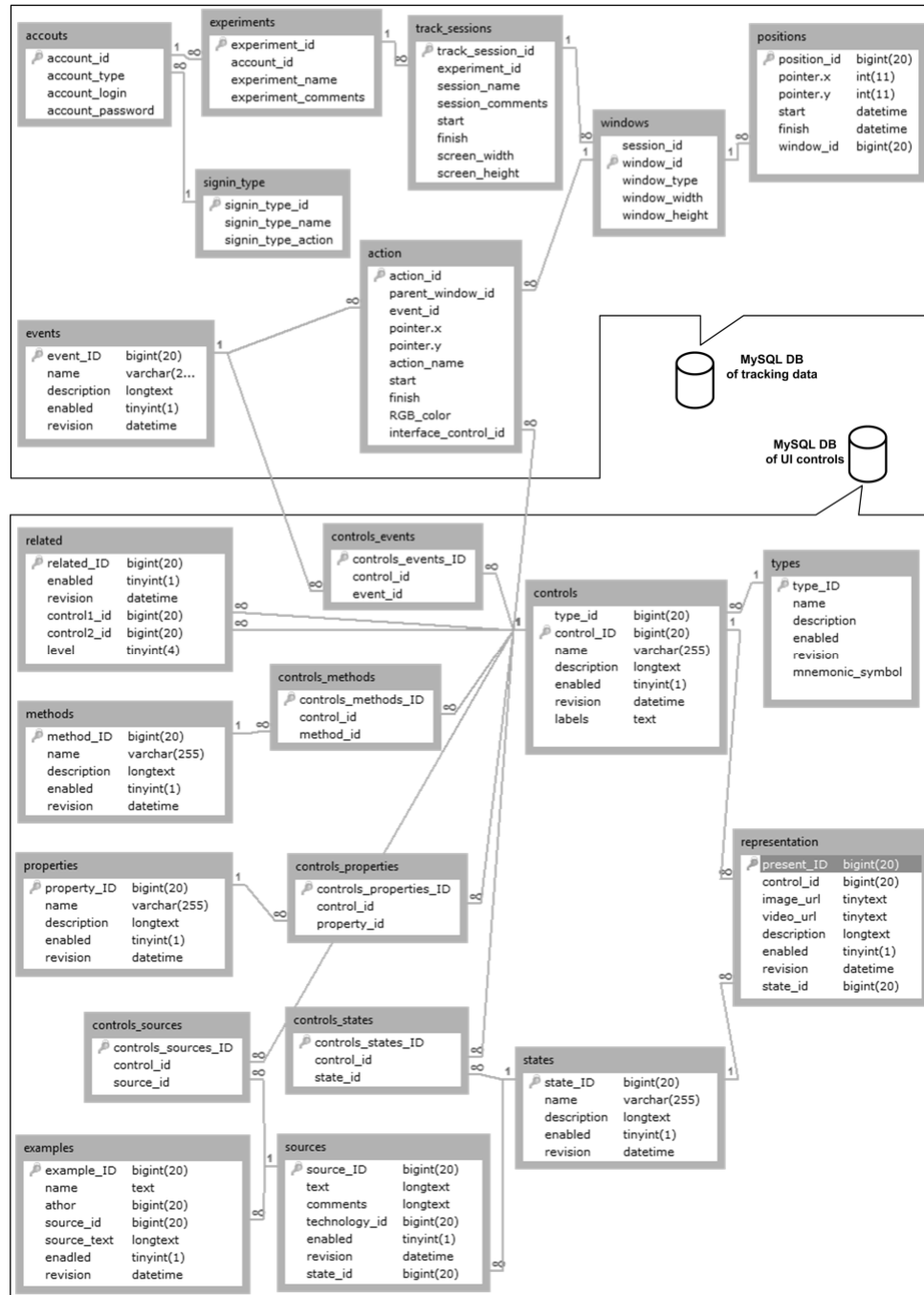


Рис. 3.19. Реляційна модель даних інтеграції даних з репозиторію конструктивних елементів інтерфейсу та даних накопичення інтерактивного використання інтерфейсу користувача

Запропонована інформаційна модель інтегрування даних (рис. 3.19) про використання інтерфейсів користувачами (база на рис. 3.12) із даними, які використовуються розробниками програмного забезпечення (база на рис. 3.14) дає змогу формувати додаткові критерії вибору конструктивних елементів інтерфейсу та створювати варіанти оформлення інтерфейсу для різних груп користувачів на основі статичної обробки даних інтерактивного використання цих інтерфейсів.

3.5. Засоби реінженірінгу програмного коду інтерфейсів користувача

Розвиток підгалузей у сфері програмування програмного забезпечення (ПЗ) сформував напрям автоматизації проектувальних робіт зі створення програмних інтерфейсів користувача і відповідного їх дослідження. Програмні продукти створюються тою чи іншою програмною технологією [79, 86, 87, 99, 100] або їх комбінацією. Але для розширення сфери впровадження такого ПЗ необхідно постійно адаптувати програмні коди під нові програмні платформи та середовища. Отже, для «еволюційного» програмного розвитку програмного продукту необхідний реінженірінг (перепроєктування) вже існуючого, щоби розширити сфери використання кінцевими користувачами. Інтерфейс користувача можна створювати різними спеціалізованими мовами програмування [99, 100]. Але немає універсального автоматизованого засобу для трансформації коду опису користувацького інтерфейсу (КІ) з одної мови програмування на іншу. Іншою проблемою може бути відсутність доступу до початкових кодів опису інтерфейсів користувача. У цьому випадку необхідне автоматичне або напівавтоматичне виявлення усіх інтерактивних взаємодій з елементами інтерфейсу користувача з подальшим автоматизованим перепроєктуванням у нову програмістську технологію чи бібліотеки (frameworks).

Проектування нового інтерфейсу для користувача або перепроектування вже існуючого – сьогодні, мабуть, найнеобхідна на ринку розробки ПЗ послуга. Регулярна модернізація існуючих інтерфейсів ПЗ здатна продовжити термін життя та використання програмне забезпечення. Інформаційна архітектура підсистеми автоматичного чи напівавтоматичного реінженірінгу користувацьких інтерфейсів формує семантичний аналізатор-синтезатор на основі технології XSLT та репозиторію еквівалентного співставлення локальних та веб-рішень графічних елементів користувацького інтерфейсу.

На рис. 3.20 представлена схема роботи [124] системи автоматизованого перепроектування інтерфейсу користувача:

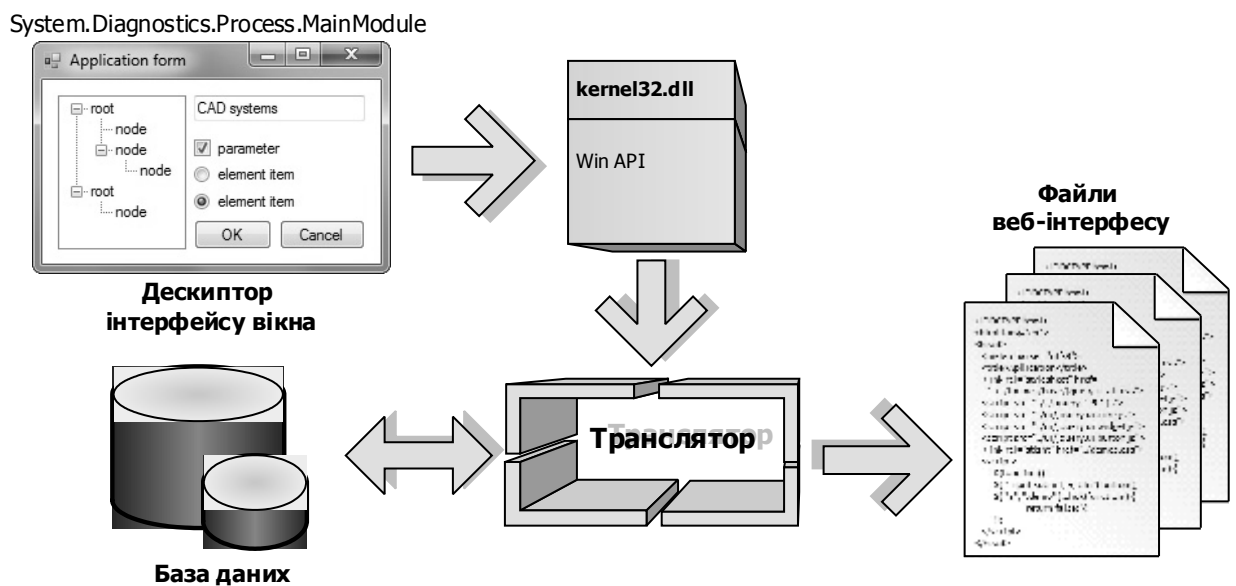


Рис. 3.20. Загальна схема процесу трансляції GUI

Попередньо система читає усі запущені процеси операційної системи через дескриптори форм GUI, але працює лише з тим, який вказує користувач. За допомогою функцій WinAPI [150] досліджується графічне представлення процесу в ОС: його меню, кнопки, допоміжні вікна, піктограми тощо. Як показано на рис. 3.20, представлення такого виду, у вигляді функцій, переходить у блок «Транслятор» для подальшого

опрацювання. Система використовує базу даних, до якої звертається транслятор при генерації нового інтерфейсу.

Результатом роботи системи є файл з програмним кодом, який у інтернет-браузері повинен відтворювати інтерфейс вхідного потоку.

На рис. 3.21 детальніше показано схему процесу [124, 125] роботу транслятора:

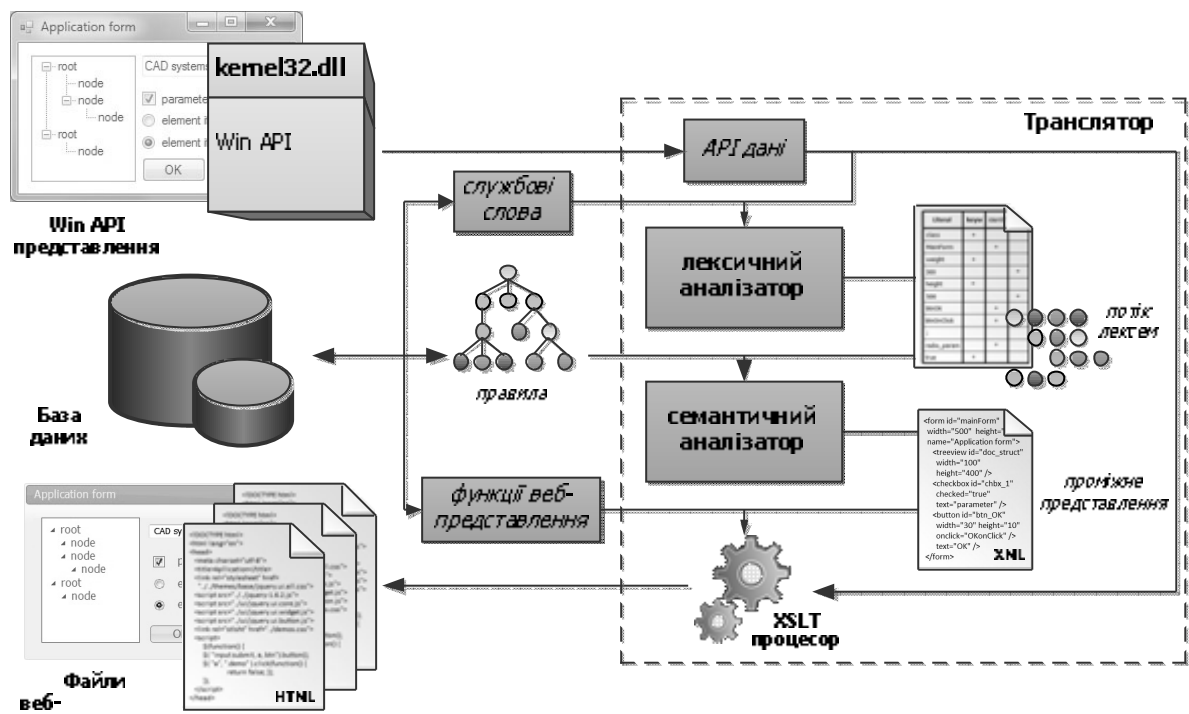


Рис. 3.21. Схема процесу роботи транслятора лексем мов опису інтерфейсу користувача

На вхід транслятора надходить програмний код у вигляді функцій, з яких можна виділити такі характеристики як: тип графічного компонента, назва, висота, ширина, піктограма (фонова картинка), активність тощо. Уся ця інформація подається у лексичний блок транслятора. Аналізатор, у свою чергу, розбиває вхідний код на лексичні складові – лексеми (службові слова, ідентифікатори, розділювачі). На цьому етапі роботи системи здійснюється оперування не конкретно програмним кодом, а функціями, які реалізують

графічне представлення, та залежностями між ними, для того, щоб майбутній інтерфейс повністю відтворював динаміку роботи вхідного процесу.

Наступним робочим блоком транслятора є семантичний аналізатор. За допомогою бази даних, яка містить граматику розбору вхідної інформації, виконується синтаксична перевірка. На цьому етапі генеруються та виводяться помилки вхідної інформації, якщо такі знайдено.

Після цього формується представлення інформації проміжною мовою, якою обрано нотації XML [151], для представлення вхідного інтерфейсу на мові опису абстракцій. XML дозволяє відокремити дані для повноцінного представлення ієрархії взаємодій компонентів інтерфейсу. При успішному виконанні даного етапу отримується структура інтерфейсу у вигляді ієрархії належності елементів вікон самим формам, структуру передачі керування між вікнами, їхню активність та активність їхніх елементів. На цьому етапі можна використати деяку підсистему візуалізації, – візуалізатор, який би перетворив дану об'єктну специфікацію у прототипне, графічне відображення. Оскільки процес реінженірінгу передбачає внесення змін, це надало б кінцевому користувачу можливість доопрацювати недоліки вхідного інтерфейсу.

Завершальним кроком роботи транслятора є генерація коду, який в системі реалізовується за допомогою XSLT [153] процесора. З бази даних співставляються функції веб-рішень локального представлення компонентів інтерфейсу, які подаються на вхід процесора разом із раніше відокремленими даними, та граматиною переводу XML на інші мови, такі як HTML [152], CSS, з використанням функцій JavaScript, чи jQuery [154].

3.5.1. Концептуальна модель БД

Концептуальна модель – це відображення предметної області, для якої розробляється база даних. Усі об'єкти, які відображають сутності, позначаються у вигляді прямокутника. Атрибути, що характеризують

сутність – у вигляді овалу, а зв'язки між сутностями – ромбами. Тип зв'язку позначаються так: у напрямку, де відношення дорівнює багатьом – обернена стрілка на кінці, а зі сторони, де відношення рівне одиниці – без стрілки).

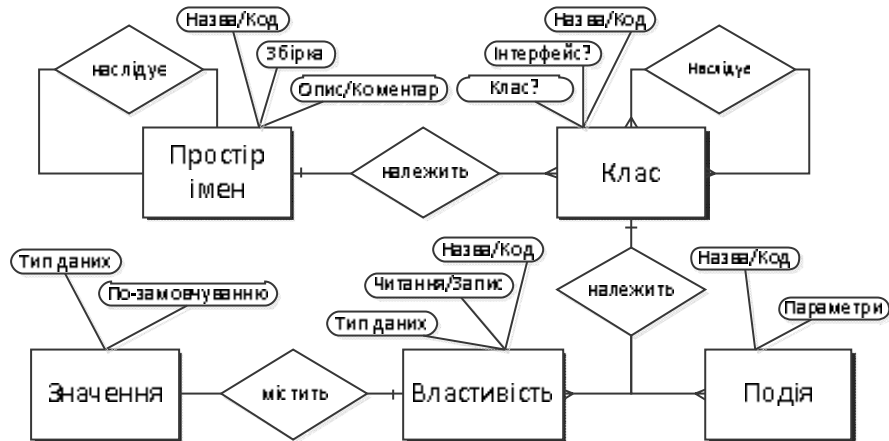


Рис. 3.22. Концептуальна модель ООП структури класу

Тому перед початком створення структури бази даних для проектованої системи була розроблена концептуальна модель предметної області.

Розроблені моделі дали змогу більш чіткіше зрозуміти відношення різних сутностей між собою, визначити типи даних, які будуть міститися у полях таблиць, побачити загальну структуру зберігання інформації проектованої системи.

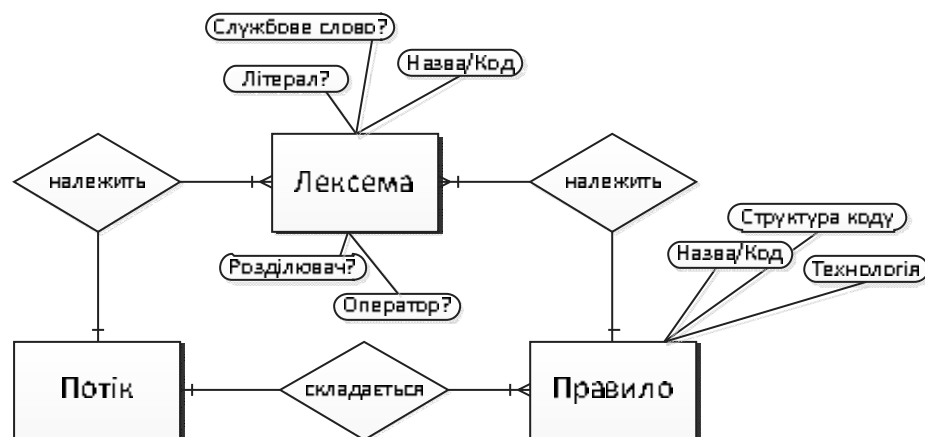


Рис. 3.23. Концептуальна модель семантичного аналізу

3.5.2. Реляційна модель бази даних

На рис. 3.24 представлена [125] реляційна модель бази даних. Перш за все, база даних повинна містити три основні категорії інформації:

- службові слова;
- правила розбору;
- альтернативні коди представлення елементів інтерфейсу користувача у веб-середовищі.

Перша частина (*службові слова*) в БД буде приймати образ прототипів класів бібліотек *.NET Frameworks*. Це реалізовано за допомогою таких таблиць:

- таблиця «*tbl_Type*» містить інформацію про назву типів (класів) та простір імен де власне можна знайти даний клас. Поле «*IsInterface*» вказує, чи являється даний тип інтерфейсом;

- таблиця «*tbl_Element*» містить інформацію про властивості (від англ. – *properties*) та події (від англ. – *events*), які може містити тип, їх назву та тип даних;

- таблиця «*tbl_Class*» містить інформацію, яка буде у повному розмірі представляти шаблон класу, його назву («*tbl_Type*») та властивості/події («*tbl_Element*»), які і будуть використовуватися при десеріалізації інтерфейсних модулів обраного процесу. Власне дані про події класу будуть вкрай необхідними на етапі синтаксичного аналізу, тому що програмні конструкції залежностей та обмежень користувацьких форм та контролів будуть міститися здебільшого у методах делегатів подій.

- таблиця «*tbl_Inherit*» містить відношення наслідувань класів платформи («*TypeID*» – ідентифікатор типу, який наслідує; «*ParentID*» – ідентифікатор наслідуваного класу). Дана таблиця являється проміжною при реалізації зв'язку «багато-до-багатьох». Такою структурою досягається максимальне зниження надмірності та дублювання даних.

- таблиця «tbl_Default» містить значення властивостей класу по-замовчуванню. За рахунок наслідування одні і ті ж властивості наслідуваних класів можуть приймати різні початкові значення в залежності від типу.

- таблиця «tbl_Namespace» містить інформацію про простір імен платформи .NET, має ієрархічну структуру, яка реалізує дерево залежностей між ними.

- таблиця «tbl_ValueType» містить інформацію про типи даних та технологію, до якої вона належить («tbl_Technology»).

Наступний блок бази даних (*правила розбору*) буде невід’ємною частиною на етапі семантичного аналізу. Можна реалізувати за допомогою наступних таблиць:

- таблиця «tbl_Rule» містить інформацію про правила для розбору структур коду. Поле «ConstructionData», у текстовому вигляді містить саме правило певної структури, а поле «TechnologyID» вказує до якої технології дане правило відноситься.

- таблиця «tbl_ConstructItem» містить інформацію про елементи, які використовуються у правилах. Правило являє собою певну послідовність конструктивних елементів. Ними можуть бути службові слова («class», «while», «function»), оператори («+», «/», «%»), розділювачі («.», «», «;») чи літерали – змінні. Дана таблиця містить назву, конструктивний тип, та належність до технології таких елементів.

Дана частина бази даних є важкою для реалізації в системі даного функціоналу, хоча і не завжди буде використовуватися. Основна проблема полягає в тому, що даний блок повинен містити правила для розбору структур коду, які в основному будуть знаходитися у методах подій елементів інтерфейсу. Ця інформація потрібна для того щоб як найточніше відтворити структуру (залежність) та динаміку форм інтерфейсу.

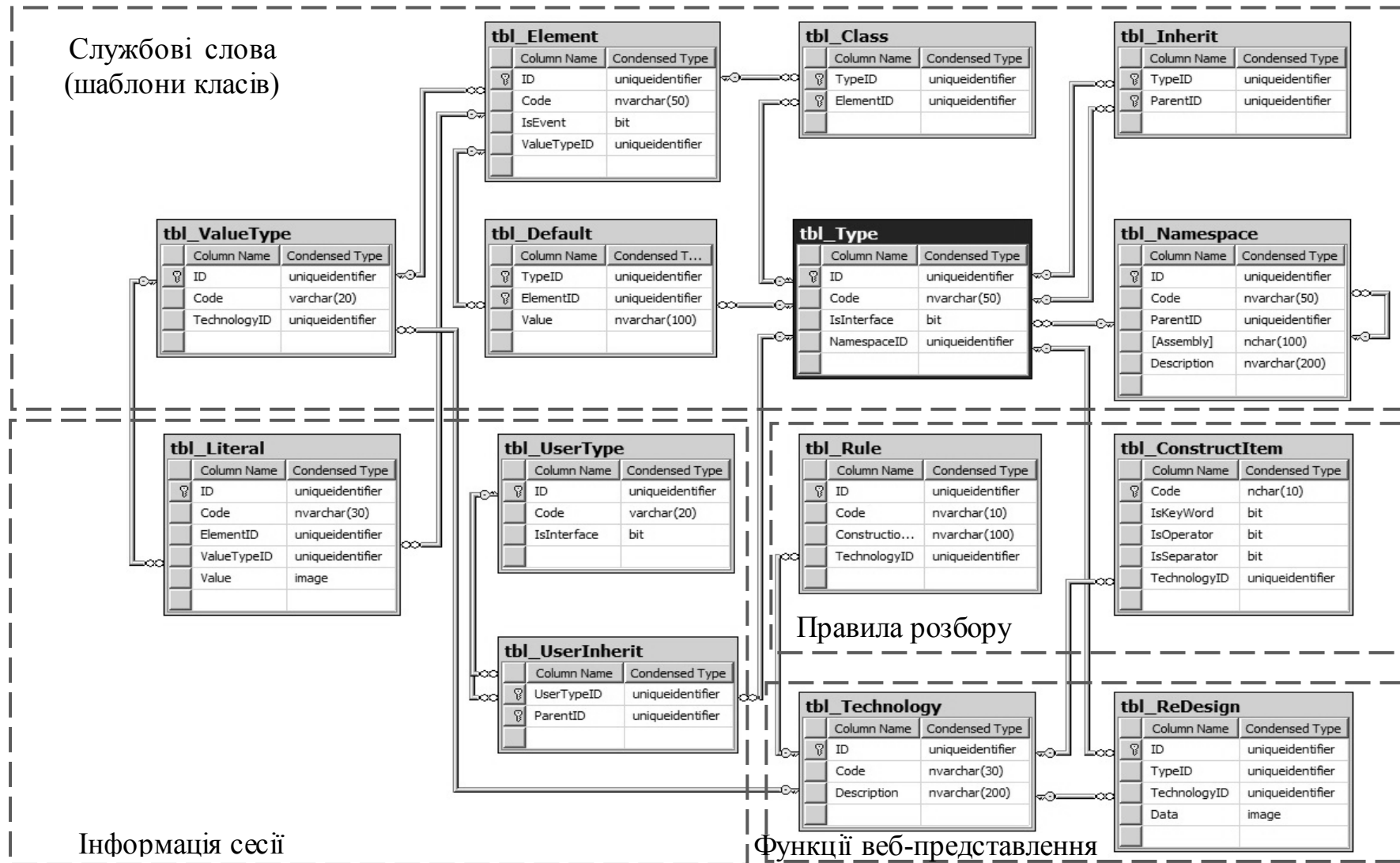


Рис. 3.24. Реляційна модель бази даних патернів перетворення програмних кодів інтерфейсів

Проблемою реалізації такого функціоналу для системи є те, що сучасні комерційні програми, і не тільки, містять розроблені коди у вигляді бібліотек (*.dll) бінарних команд, зрозумілих лише для ЕОМ, які при декомпіляції неможливо привести до початкового синтаксису, що дуже ускладнює процес переносу залежностей між UI та їхні обмеження. Проблемою реалізації такого функціоналу для бази даних є те, що навіть якщо програмні коди програмного продукту вхідного процесу представлені у відкритому тексті, то для універсальності БД системи повинна містити правила розбору структур коду найрізноманітніших технологій (*C#, Java, C++, Delphi*, тощо), на яких тільки можлива розробка інтерфейсу користувача, що знову ж таки значно ускладнює розробку даного блоку.

Останнім конструктивним блоком БД є таблиця, яка у бінарному вигляді буде містити тексти кодів альтернатив представлень класів (таблиця «tbl_ReDesign», поле «Data») у веб-середовищі. При генерації кінцевого результату представлення вхідного інтерфейсу дана таблиця буде приймати активну участь, співставляючи класи форм та елементів керування вхідного інтерфейсу із альтернативними структурами коду бази даних для певних технологій.

Клієнтською інформацією будуть доповнювати наступні таблиці:

- таблиця «tbl_Literal» міститиме інформацію про змінні, отримані на етапі синтаксичного аналізу, їх назву (поле «Code»), тип даних (поле «ValueTypeID») та значення (поле «Value»), якщо змінна носить в собі клієнтський характер, або ж належність значення до властивості певного типу (поле «ElementID»).

- таблиця «tbl_UserType» міститиме інформацію про клієнтські типи, їх назву (поле «Code»), інформацію про те, чи являється даний тип інтерфейсом (поле «IsInterface»).

- таблиця «tbl_UserInherit» міститиме інформацію про наслідування клієнтських типів у ієрархічній структурі. Поле «UserTypeID» вказує на

ідентифікатор типу, що наслідує, а поле «*ParentID*» вказує на ідентифікатор наслідуваного класу (може приймати значення як з таблиці «*tbl_Type*» так і з таблиці «*tbl_UserType*»).

При сучасному розвитку програмних технологій дана БД повинна постійно оновлюватися, щоб завжди бути актуальною і містити повну інформацію для коректної роботи всієї системи.

3.6. Інформаційна система параметризованого порівняння засобів прототипування інтерфейсів користувача

Представлена система [155] має на меті забезпечити проектувальників інтерфейсів користувачів інструментом для швидкого пошуку, порівняння та підбору за вказаними критеріями з різноманіття програмних продуктів прототипування інтерфейсів користувача. База даних розширюється як для критеріїв аналітичного підбору, так для даних про характеристики засобів прототипування.

Існує багато сайтів-агрегаторів (наприклад: [156, 157, 158, 159] та інші), де публікується інформація про програмні продукти класифіковані за категоріями призначення, операційними середовищами, основними функціями. Але в них нема механізмів детального параметричного пошуку за функціональними параметрами програмних продуктів, пошук і видача списків результатів здійснюється чи то за ключовими словами, чи то по словах коротких анотацій. І як правило засоби прототипування розмішені просто в загальній категорії програмних продуктів. І щоб проектувальник саме інтерфейсів для конкретного завдання чи під вимоги замовника зміг обрати саме потрібний інструмент, необхідно поступово переглянути більшість програмних продуктів із списку наявних у каталоги. І це зовсім не гарантує, що після ознайомлення з описами, з кількома вікнами-взірцями та після інсталювання програми прототипування і пробного періоду будуть наявні всі

необхідні функції. Усе це вимагає додаткових зусиль проектувальника та його робочого часу.

Недоліками існуючих систем є:

- Немає способи чи то методи отримання рекомендацій для обрання засобів прототипування.
- Не враховані метрики для проектів користувацьких інтерфейсів по функціоналу.
- Не враховані фактори варіацій елементів користувацьких інтерфейсів з однаковим функціональним навантаженням.

Гіпотези дослідження:

- “Популярність” завантаження ПЗ –це лише єдиний критерій «сайтів агрегаторів»? Можна покращити?!
- Чи потрібні всі можливі функції ПЗ для саме “5 хвилинного” прототипування інтерфейса користувача?
- Як підібрати ПЗ прототипування під свою інформаційну інфраструктуру (“інфоекосистему”)?
- Як зменшити час на перегляд описів і вагання чи встановлювати ПЗ чи ні?

Тому на основі попередніх досліджень [77, 127] було вирішено створити спеціалізований пошуково-інформаційний Інтернет сервіс для параметричного порівняння та підбору лише для галузі прототипування, проектування та програмування саме рівня інтерфейсів користувача програмного забезпечення. Система наповнюється метриками опису програмних продуктів, метриками інтерфейсів користувачів, метриками інтерфейсних компонентів при проектуванні. Ці описи також параметрично поповнюються рецензіями, відгуками, прикладами (текстуально, графічно, анімаціями, відео, графіками статистики тощо). Розвиток системи передбачає додатки для механізмів онлайн моніторингу появ нових продуктів прототипування інтерфейсу користувача з

подальшим перенаправленням зібраної інформації для команди на рецензію, що наповнює систему більш детальними параметризованими даними.

Головне вікно роботи системи представлено на рис. 3.25.

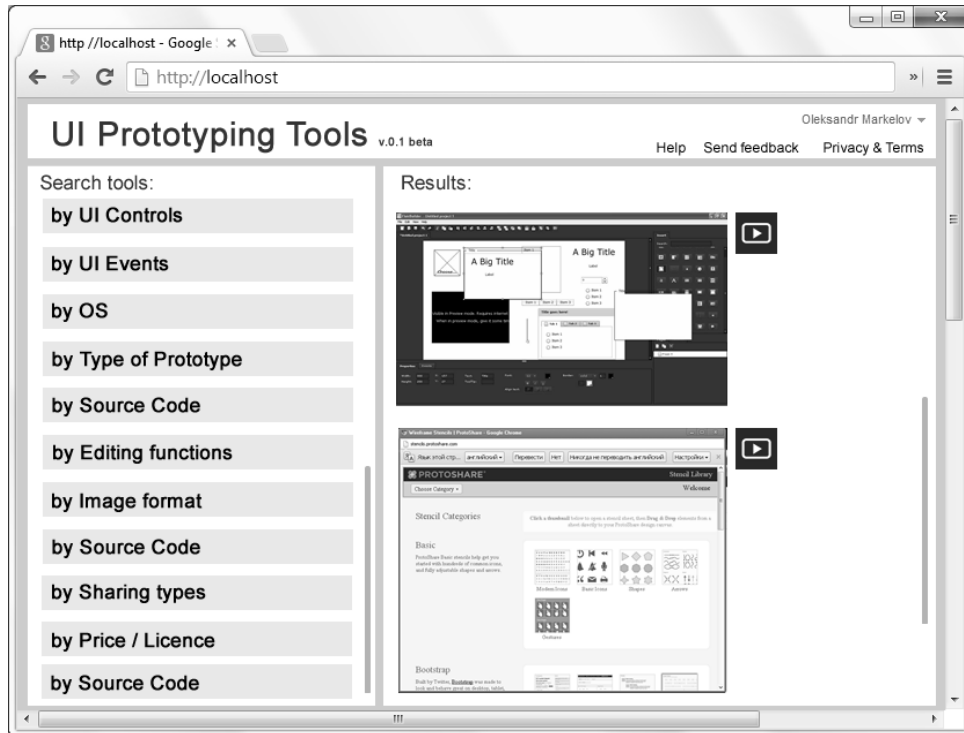


Рис. 3.25. Головне вікно системи аналітичного пошуку засобу прототипування інтерфейсу

Підчас програмної реалізації застосовані наступні технології: XML, XSLT, HTML5, JS, jQuery, CSS, MySQL Server, PHP.

Ефекти від впровадження:

- Пришвидшення вибору необхідного засобу прототипування ІК під вимоги замовника.
- Збільшення об'ємів продажів ліцензій на використання ПЗ саме на основі попередніх вимог ТЗ;
- Підвищення продуктивності та задоволення користувачів-проектантів ІК за рахунок ненадлишковості функціоналу ПЗ прототипування;

- Зменшення часу проєктантів та програмістів на створення та модифікацію ПЗ.

3.7. Висновки до розділу

1. Визначено та обґрунтовано концептуальні принципи організації новітньої системи автоматизації синтезу інтерфейсу користувача у вигляді комплексу інструментальних інформаційних засобів. До основних організаційних принципів віднесено автоматизація накопичення інтерактивної взаємодії користувачів з готовими інтерфейсами користувача, систематизація патернів конструктивних елементів ГІК з візуальними та програмними характеристиками, розподілена "клієнт-серверна" архітектура з центральним інтегрованим інформаційним фондом метрик, досвіду інтерактиву, патернів програмних кодів, патернів макетування площинних компоновок, інтеграцією з мережею Інтернет, автоматизації синтезу програмних кодів із візуального ідентифікування прототипів компонентів ГІК.

2. Розроблено розподілені архітектури мережевих засобів (підсистем) автоматизації побудови (синтезу) інтерфейсів користувача, які втілюють сформовані ключові принципи, формалізації, методи організації процесів побудови ІК на базі інтелектуалізації патернів ГІК.

4. Розроблено та реалізовано інформаційне, лінгвістичне, програмне забезпечення засобів накопичення діалогових користувацьких інтерактивів, засобу формування та наповнення репозиторію конструктивів ГІК, засобу візуальної ідентифікації конструктивів ІК, засобу реінженірінгу програмного коду ІК, засобу параметричного порівняння засобів прототипування ІК

Розділ 4. Результати дослідження та прототипування інтерфейсів користувача

У четвертому розділі відображені результати статистичного аналізу спектрального розподілу застосованих конструктивів графічного інтерфейсу в програмному забезпеченні САПР. Представлені результати синтезу сценаріїв діалогових інтерфейсів користувача на прикладі для графічного редактора електричних схем, наведена матриця отриманих зв'язків змістовних об'єктів сценаріїв діалогів інтерфейсів користувача, відображені результати візуальної ідентифікації в автоматизованому процесі синтезу коду декларування інтерфейсу користувача з графічних прототипів.

Основні результати розділу опубліковані автором у працях [12, 13, 119, 120, 127]








4.1. Статистичні оцінки аналізу використання конструктивів ГІК

4.1.1. Оцінка реалізації інтерфейсних компонентів у САПР схемотехніки

Щоб отримати кількісні дані про типи компонентів користувацького інтерфейсу програмних засобів були обрані [126, 146, 147, 148, 149] системи САПР схемотехнічного проектування для порівняння. Результати застосованих розробниками ПЗ САПР компонентів інтерфейсу користувача наведені в табл. 4.1. Групування й усереднення значень статистичного аналізу показало, що розробники використали лише до 10 основних конструктивних елементів графічного інтерфейсу користувача із цілого спектру (табл. 2.1), виявлених у дисертаційному дослідженні, конструктивів ГІК.

Таблиця 4.1.

Спектр застосування GUI компонентів в САП схемотехніки

Основні типи компонентів інтерфейсу	Кількість, %
 Button	70%
 Menu/ MenuItems	20%
 Common Box	5%
 Spin Edit	1%
 Text Edit	1%
 Tree View	0.5%
 ListView	0.5%
Інші	1%

4.1.2. Кількісна оцінка застосованих інтерфейсних компонентів у САПР машинобудування

Визначення кількісних характеристик, застосованих у сучасних САПР, компонентів інтерфейсу користувача програмного забезпечення здійснено [127] на прикладі САПР в галузі машинобудування. Ці характеристики дозволяють архітекторам ПЗ та програмістам зосередитися на поліпшенні існуючих інтерфейсів і вдосконалення методів реінженірінгу інтерфейсів, що використовують інновації та тенденції в компонентах користувача інтерфейсу.

Статистична інформація про спектр компонентів графічного інтерфейсу користувача (GUI) отриманий із репозиторію [143, 145] на показана у вигляді стовпцевої діаграми на рис. 4.10. Компоненти ПК згруповані в 6 груп: командні (опція), контейнера, вибір вхідних дані із варіантів, навігація, інформація, віджет / гаджет. Статистичні кількісні розрахунки були проведені для програмне забезпечення САПР машинобудівної галузі для 60 типів графічних інтерфейсних елементів як: Button, Icon Button, Split Button, Cycle button, Checkbox, Radio button, Spinner (StepperInput), Switcher (Toggles), Frame, Fieldset, Group Box, Panel, Text Edit, Icon, Label, Hyperlink, Tooltip (Hint), Slider, Status bar, Progress bar, Progress Indicator, Splitter, Scrollbar, Zoomer, Tabs

(Pages), Text Box (Memo), RichText, List box, Drop-down list, Combo box, Reyting (Stars), Tag Cloud, Tree view, Grid view, Menu, Hierarchical Menus, Context menu, Pie menu (circle menu), Menu bar, Menu extra (Dock), Toolbar, Ribbon, Accordion, Window, Modal window, Pop-up window, Dialog box, Alert box, Carousels, Coverflow Carousel, Balloon help, Infobar, Address bar, PathViewer (Breadcrumb), List Nav Bar, Pagination, Pointer (Cursor), ColorPicker, Disclosure widget, Palette window (Utility window), Inspector window, Toast (Notifier) та інші.

В якості досліджуваних САПР в експерименті обрані 15 САПР як: AutoCAD [128], BRL-CAD [129], BtoCAD [130], FreeCAD [131], IronCAD [132], Kompas [133], MeshLab [134], nanoCAD [135], Open CASCADE [136], progeCAD [137], qCAD [138], SALOME [139], SolidWorks [140], T-FLEX CAD [141], TurboCAD [142]. На наступних рисунках (рис. 4.2- 4.7) показані центральні віконні форми ГІК досліджуваних САПР.

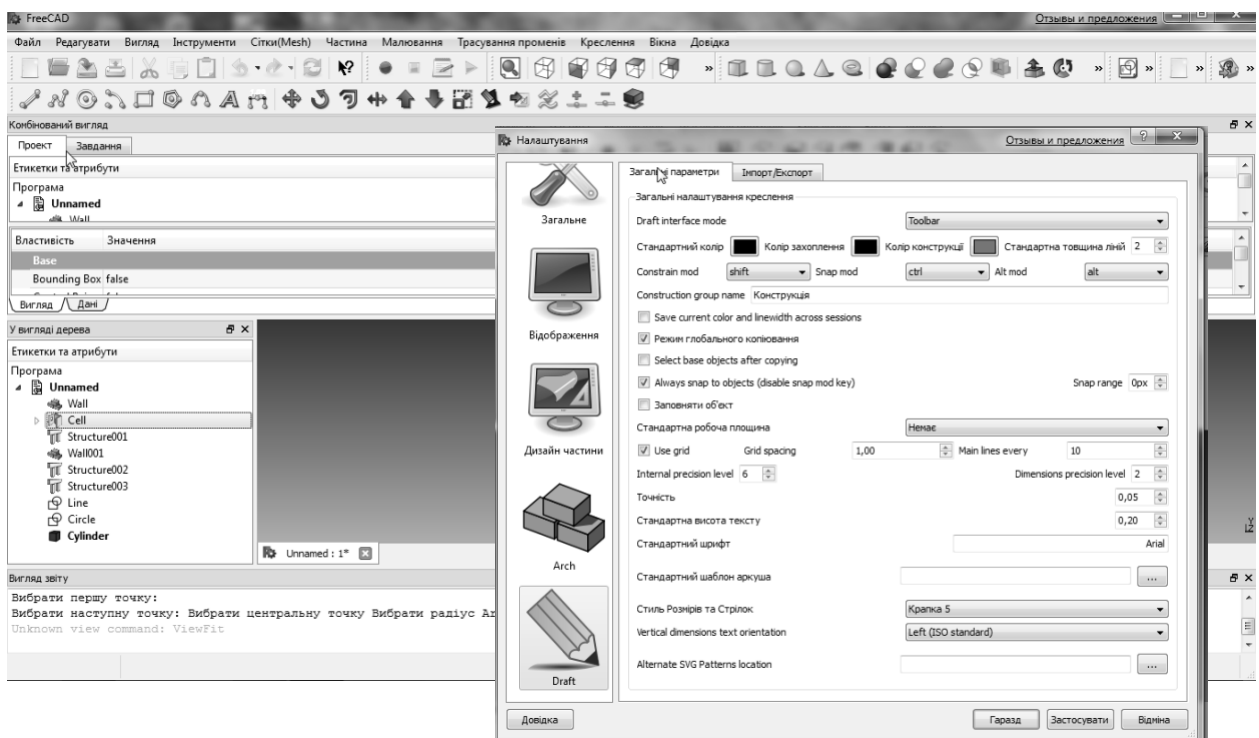


Рис. 4.1. Основні типові віконні форми ГІК FreeCAD

Спроектвані графічні прототипи інтерфейсів користувача для FreeCAD для вікна «Налаштування» та «Комбінований вигляд» показано на рис. 4.2.

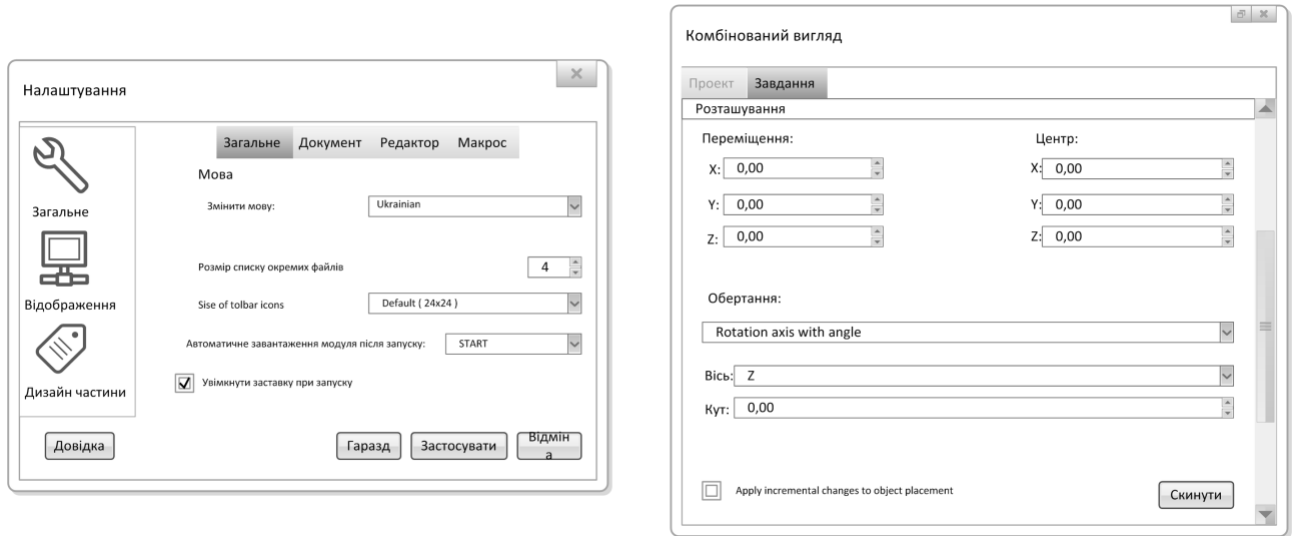


Рис. 4.2. Приклади спроектованих прототипів ГІК для на базі FreeCAD

На рис. 4.3 показано головне вікно САПР TurboCAD, як видно із зображення переважаючий спектр конструктивів ГІК – це Menu items, ToolBar Icon Button, EditBox.

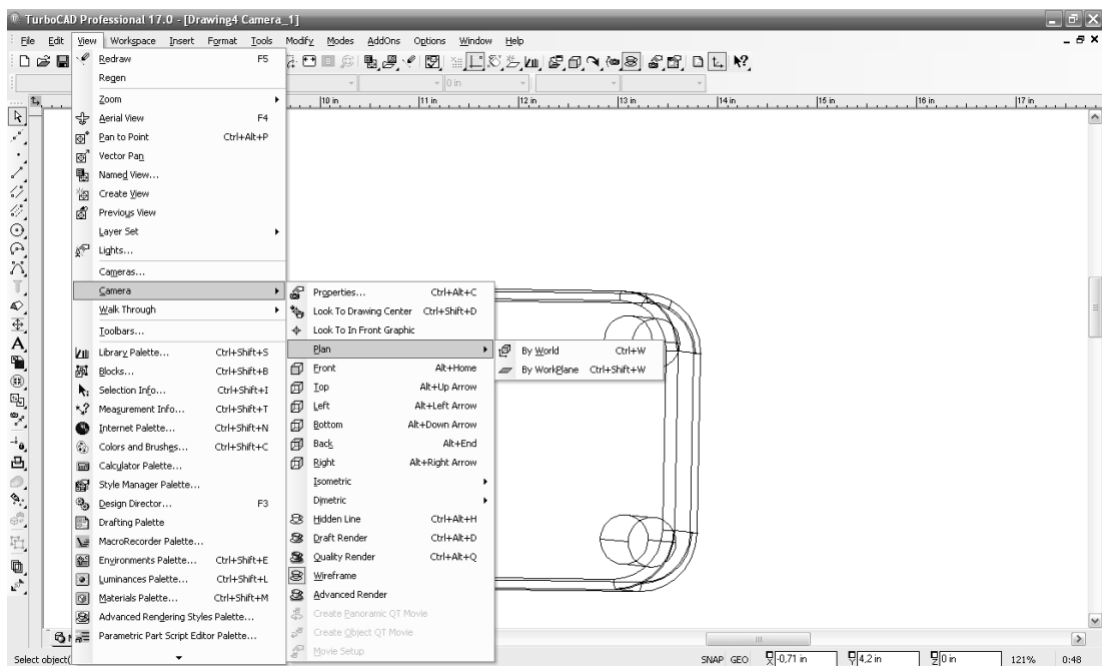


Рис. 4.3. Основні типові віконні форми ГІК TurboCAD

Результатом ескізного прототипування ГІК для TurboCAD є зображення на рис. 4.3.

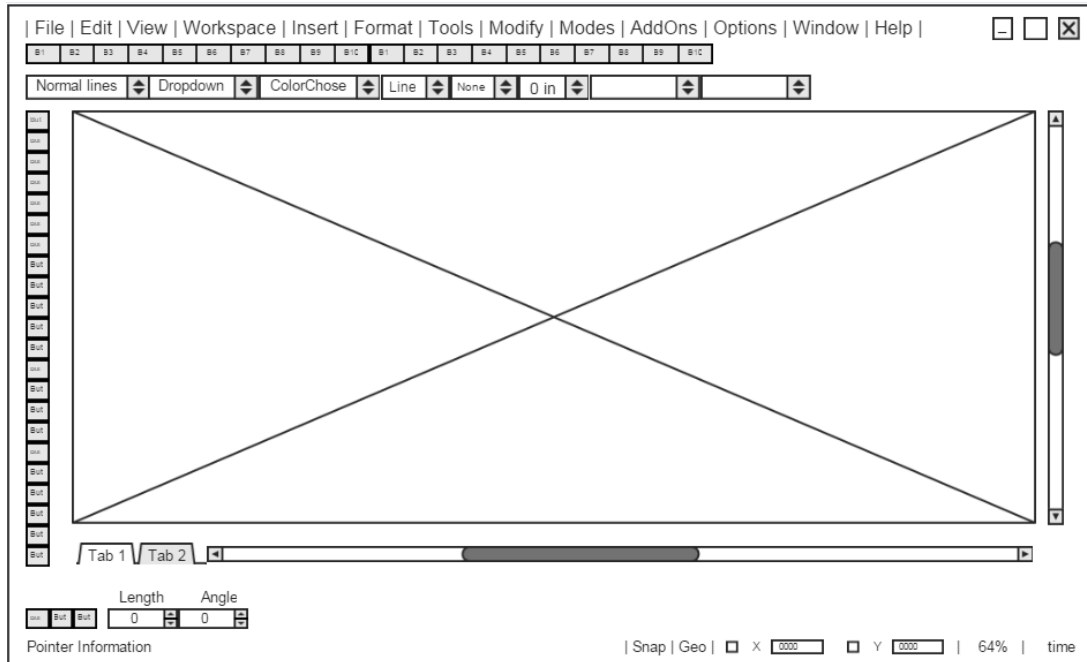


Рис. 4.4. Приклади спроектованих прототипів ГІК для на базі TurboCAD

Площинне компонування (рис. 4.5) графічного інтерфейсу користувача для вікон «Налаштування» у ПЗ CADwork 3D містить CheckBox, Tabs, RadioButton, DropDownList, EditBox.

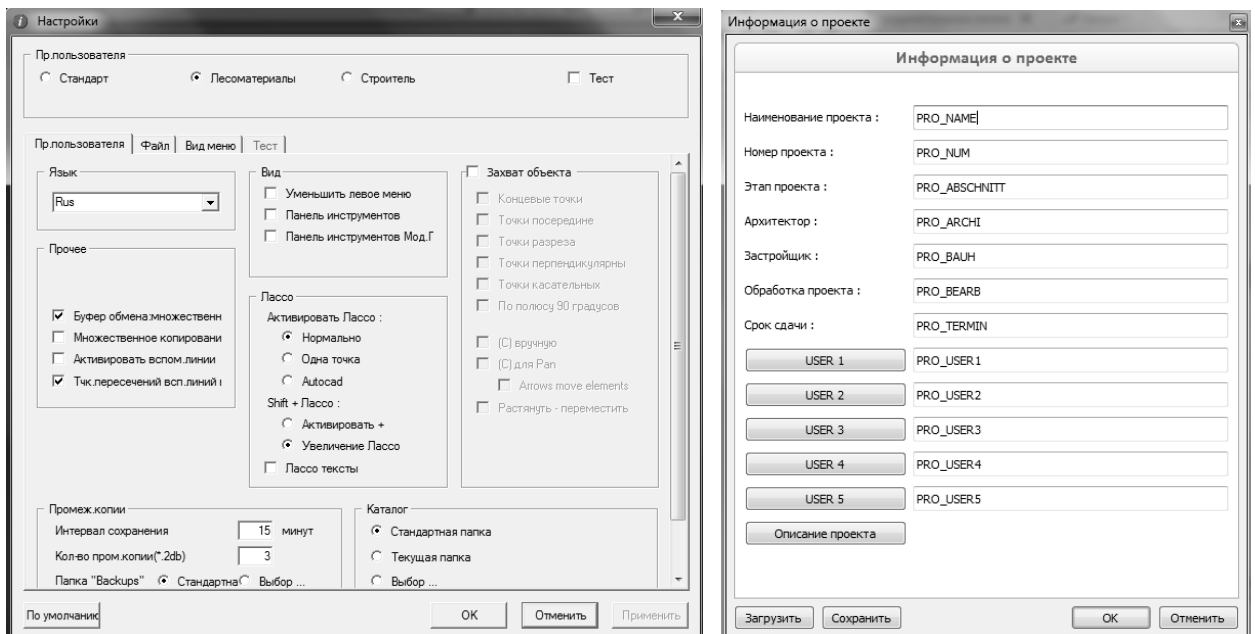


Рис. 4.5. Віконні форми ГІК «налаштування» проектів у CADwork 2D

Результат синтезу графічного прототипу на основі структури віконних

форм CADwork 2D показано на рис. 4.6

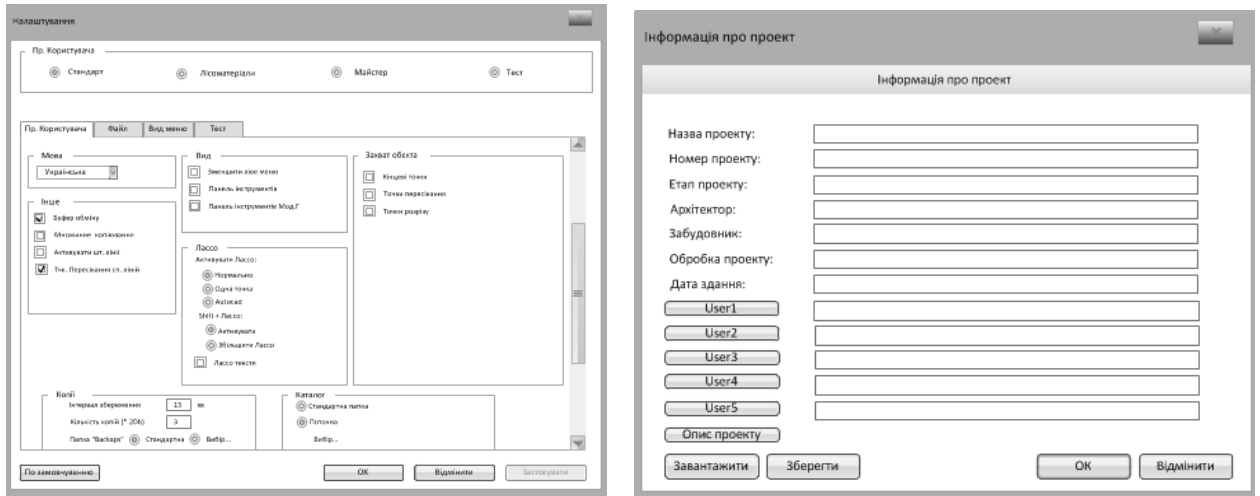


Рис. 4.6. Приклади спроектованих прототипів ГІК для progeCAD

У системі проектування SALOME (рис. 4.7) присутні елементи ГІК: меню, кнопки, дерево ієрархії, закладки, панелі інструментів тощо.

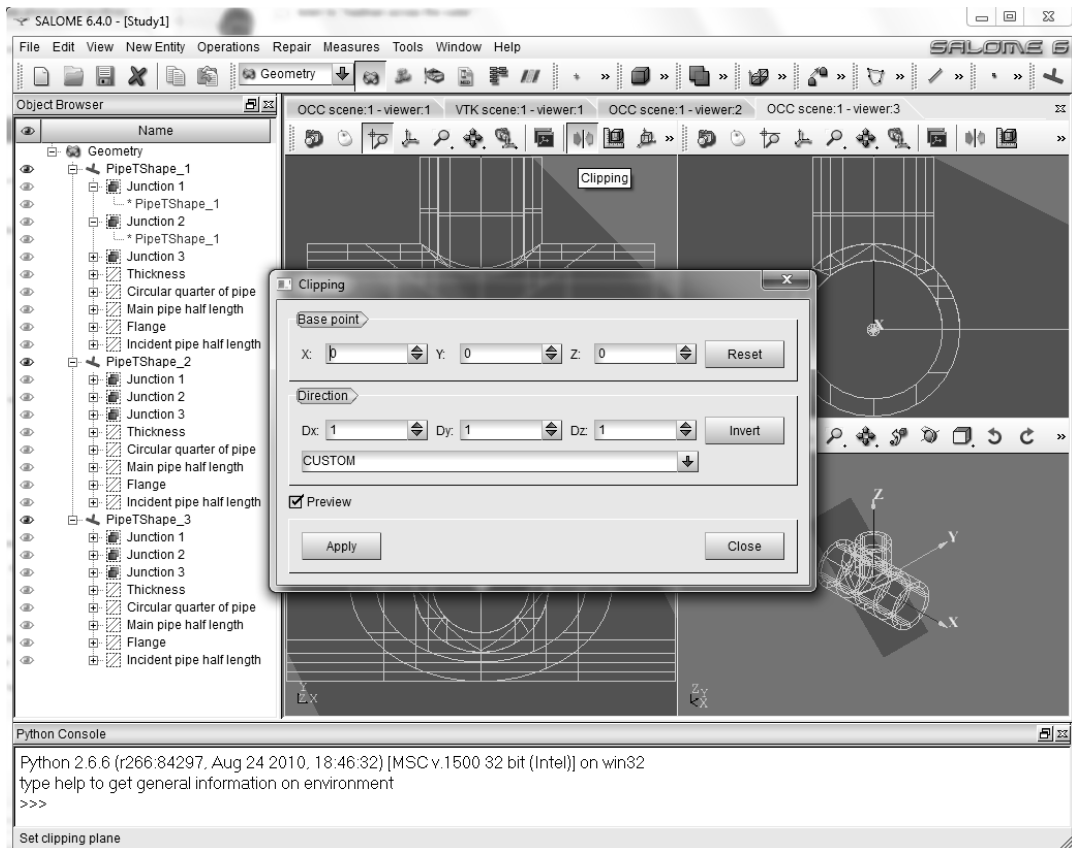
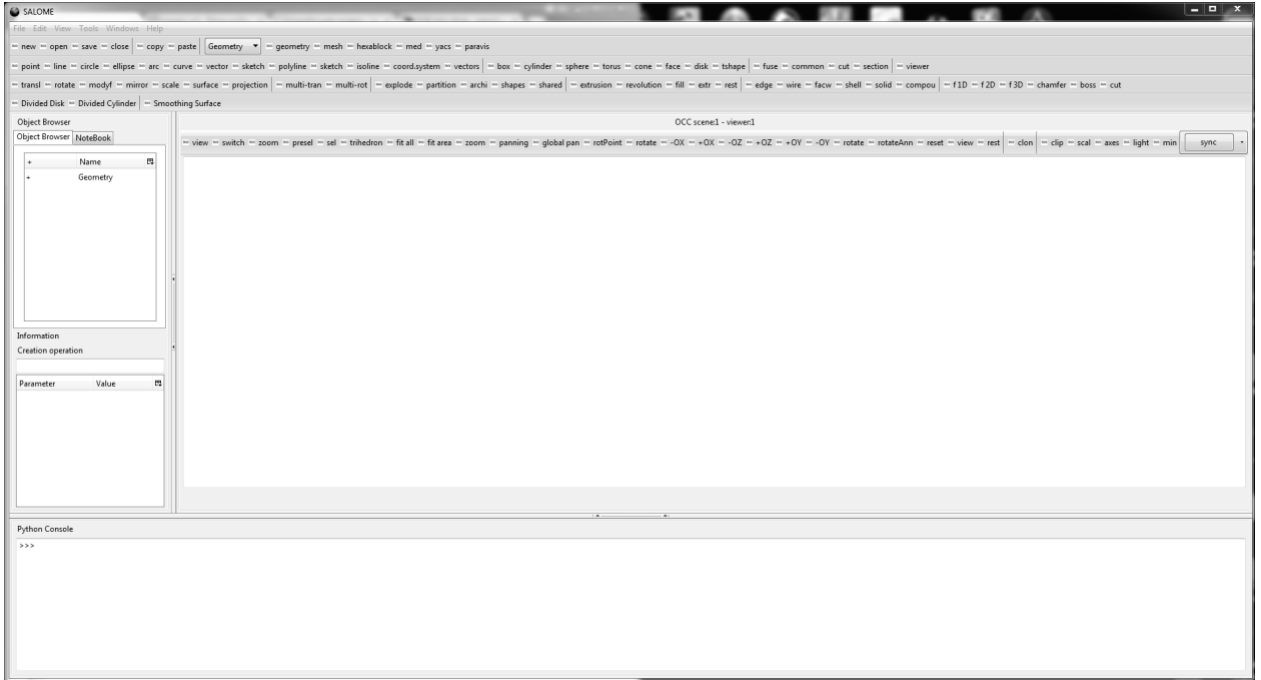
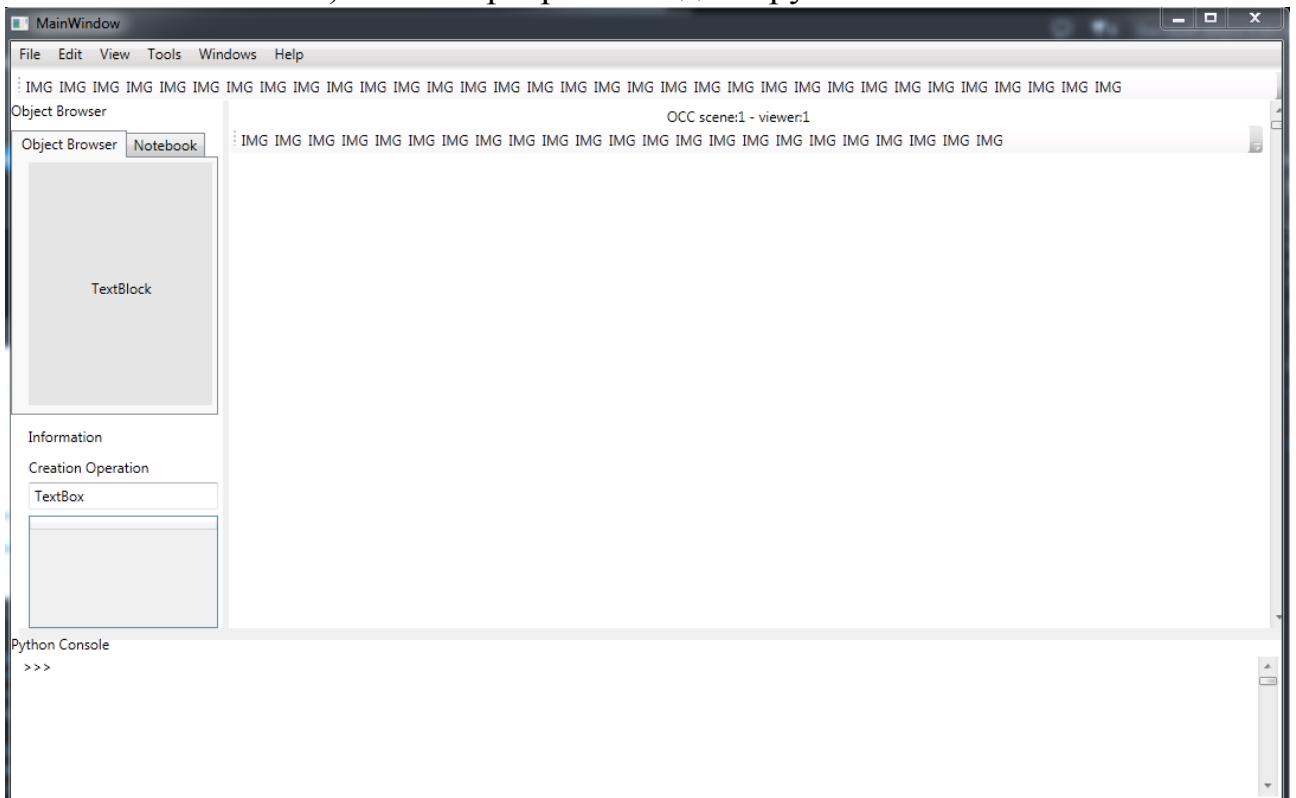


Рис. 4.7. Основні типові віконні форми ГІК SALOME



а) мовою програмного декларування XUL



б) мовою програмного декларування XAML

Рис. 4.8. Приклади синтезованого прототипу ГІК для SALOME

Графічна візуалізація результатів автоматизації синтезу програмних кодів мовами декларування інтерфейсів користувача показано на рис. 4.8. Приклади синтезованих програмних кодів на основі шаблонних фрагментів (патернів) на

мовах наведено на рис. 4.9. Лістинги містять заголовки XML-форматування. Мова XAML спрощує створення корисувачького інтерфейсу для прикладних програм у моделі програмування «.NET Framework».

<pre> <?xml version="1.0"?> <?xml-stylesheet href="chrome://global/skin/" type="text/css"> <window id="example-window" title="SALOME" xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul"> <script type="application/javascript" src="chrome://myapp/content/main.js"/> <toolbox id="toolboxid"> <menubar id="menubarid"> <menu id="menuid" label="File" accesskey=""> <menupopup> <menuitem id="menuid" label="New" oncommand="Ctrl+P"/> <menuitem id="menuid" label="Open.." oncommand="getOpenFile();"/> <menuitem id="menuid" label="Reopen" oncommand="{action}"/> <menuitem id="menuid" label="Save" oncommand="{action}"/> <menuitem id="menuid" label="Save as.." oncommand="{action}"/> <menuitem id="menuid" label="Close" oncommand="{action}"/> <menuseparator/> <menuitem id="menuid" label="Connect" oncommand="{action}"/> <menuitem id="menuid" label="Disconnect" oncommand="{action}"/> <menuseparator/> <menuitem id="menuid" label="Dump Study.." oncommand="{action}"/> <menuitem id="menuid" label="Load Script.." oncommand="{action}"/> <menuseparator/> <menuitem id="menuid" label="Properties.." oncommand="getStudy();"/> <menuseparator/> <menuitem id="menuid" label="Preferences.." oncommand="{action}"/> <menuseparator/> <menu id="Import-menu" label="Most Recently Used"> <menupopup id="Import-popup"> <menuitem label="Empty" /> <menuitem label="Clear" /> </menupopup> </menu> <menuseparator/> <menuitem id="menuid" label="Exit" oncommand="{action}"/> </menupopup> </menu> <menu id="menuid" label="View" accesskey=""> <menupopup> <menu id="Toolbar-menu" label="Toolbars"> <menupopup id="Toolbar-popup"> <checkbox label="Standart" checked="{true *false}"/> <checkbox label="Modules" checked="{true *false}"/> </menupopup> </menu> <menu id="Windows-menu" label="Windows"> <menupopup id="Windows-popup"> <checkbox label="Python Console" checked="{true *false}"/> <checkbox label="Object Browser" checked="{true *false}"/> </menupopup> </menu> <menuseparator/> <checkbox label="Status Bar" checked="{true *false}"/> <menuseparator/> <menuitem id="menuid" label="Theme" oncommand="{action}"/> <menuitem id="menuid" label="Full Screen" oncommand="{action}"/> </menupopup> </menu> </pre>	<pre> <Window xmlns="http://schemas.microsoft.com/winfx /2006/xaml/presentation xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" xmlns:d="http://schemas.microsoft.com/expression/blend/2008" xmlns:mc="http://schemas.openxmlformats.org/markup- compatibility/2006" mc:Ignorable="d" x:Class="project.MainWindow" Title="MainWindow" Height="677" Width="1089"> <Grid> <Grid.ColumnDefinitions> <ColumnDefinition Width="145*"/> <ColumnDefinition Width="936*"/> </Grid.ColumnDefinitions> <Menu Height="25" VerticalAlignment="Top" Grid.ColumnSpan="2"> <MenuItem Header="File"> <MenuItem Header="New" HorizontalAlignment="Left" Width="145"/> <MenuItem Header="Open.." HorizontalAlignment="Left" Width="150" Margin="0,0,-5,0" Click="Open"/> <MenuItem Header="Reopen" HorizontalAlignment="Left" Width="145"/> <MenuItem Header="Save" HorizontalAlignment="Left" Width="145"/> <MenuItem Header="Save as.." HorizontalAlignment="Left" Width="145"/> <MenuItem Header="Close" HorizontalAlignment="Left" Width="145"/> <Separator HorizontalAlignment="Left" Width="145"/> <MenuItem Header="Connect" HorizontalAlignment="Left" Width="165" Margin="0,0,-20,0"/> <Separator HorizontalAlignment="Left" Width="145"/> <MenuItem Header="Dump Study.." HorizontalAlignment="Left" Width="175" Margin="0,0,-30,0"/> <MenuItem Header="Load Script.." HorizontalAlignment="Left" Width="160" Margin="0,0,-15,0"/> <Separator HorizontalAlignment="Left" Width="145"/> <MenuItem Header="Properties.." HorizontalAlignment="Left" Width="165" Margin="0,0,-20,0" Click="StudyProp"/> <Separator HorizontalAlignment="Left" Width="145"/> <MenuItem Header="Preferences.." Click="Preferences"/> <Separator HorizontalAlignment="Left" Width="145"/> <MenuItem Header="Most Recently Used" HorizontalAlignment="Stretch" Width="Auto" Margin="0,0,-8,0"> <MenuItem Header="Empty" HorizontalAlignment="Left" Width="145"/> <MenuItem Header="Clear" HorizontalAlignment="Left" Width="145"/> <MenuItem> <Separator HorizontalAlignment="Left" Width="145"/> <MenuItem Header="Exit" HorizontalAlignment="Left" Width="185" Click="Disconnect\$"/> </MenuItem> </pre>
--	--

Рис. 4.9. Приклади фрагментів програмного коду мовами XUL та XAML

XUL описує інтерфейси і дозволяє задавати зовнішній вигляд програми через CSS та визначати поведінку за допомогою JavaScript.

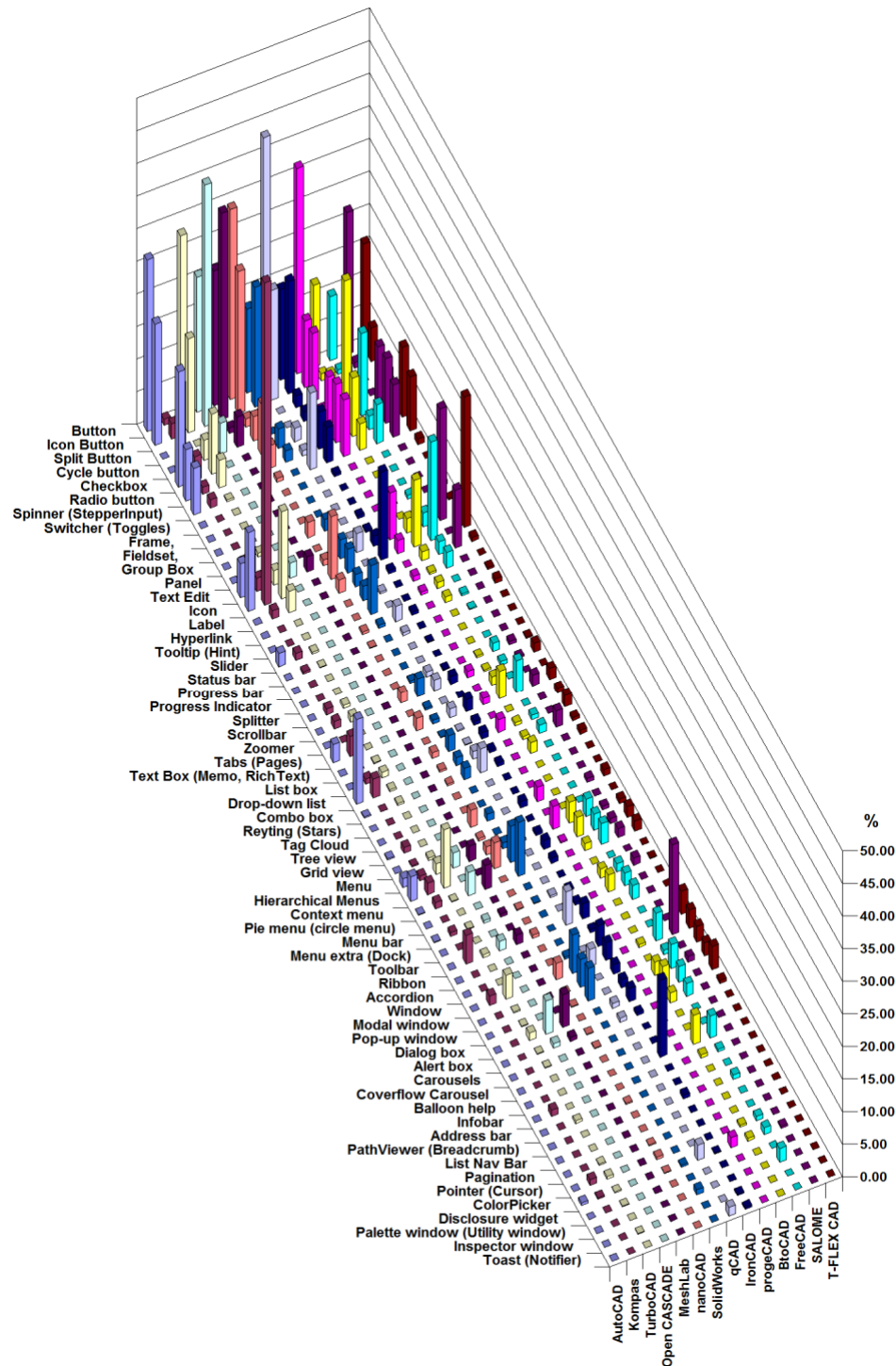


Рис. 4.10. Просторова гістограма спектру компонентів ГІК у САПР машинобудування

Гістограма на рис. 4.10 показує процентне відношення застосування різноманітних типів (табл. 2.1) конструктивних елементів графічного інтерфейсу користувача у досліджуваних САПР і вказує на слабкий рівень

впровадження інноваційних конструктивів у інтерфейси, що вказує на передумову реінженірінгу рівня інтерфейсів із застосуванням новітніх спеціалізованих мов (табл. 2.3) програмування й декларування інтерфейсів користувача програмного забезпечення.

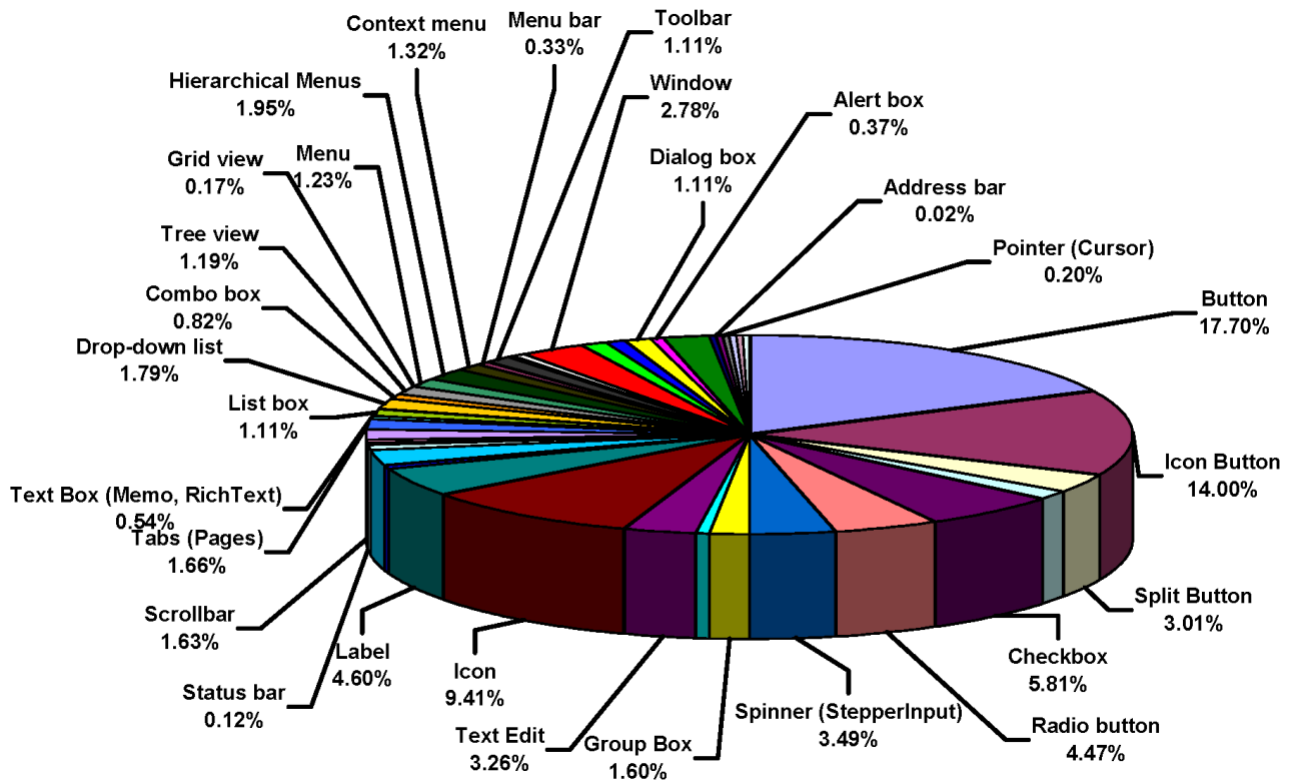


Рис. 4.11. Колова діаграма співвідношень розподілу компонентів ГІК у САПР машинобудування

При аналітичному дослідженні описаних САПР було виявлено, що переважна більшість реалізованого в їхньому ПЗ патернів користувацького інтерфейсу не містить тенденцій інноваційних інтерфейсів описаних в [20, 45].

Усього було опрацьовано понад 22 тис. компонентів графічного інтерфейсу користувача для обраних в експерименті САПР, розміщених на різних віконних формах інтерфейсу користувача та у різних станах активності конструктивів на віконних формах.

4.2. Результати якісної оцінки інструментів моніторингу подій ГІК ПЗ

У порівняльній табл. 4.2 наведено можливості деяких інструментів для моніторингу подій в ПЗ для визначення викликів бібліотечних функції та подій ОС. У більшості випадків ці інструменти застосовуються для аналізування подій від керування користувача, виявлення точних даних процесів.

Таблиця 4.2.

Порівняльна таблиця оцінки інструментів для моніторингу за функціональністю

Показник Функціональність	Інструменти					
	MS Spy++ v11	Blade API Monitor v3.8	API Monitor v2 Alpha	WinApi Override v6.4	Process Hacker 2	API Logger 1.9
1	2	3	4	5	6	7
Монітор подій	Так					
Монітор API ОС	Частково*	Так	Так	Так		Так
Монітор кастомних .dll		Так				
Монітор MFC		Так	Так	Так		Так
Монітор .NET		Ні	Так	Так		Ні
Монітор ActiveX, COM		Так	Так	Так		Так
Ін'єкція .dll			Так*	Так	Так	
Моніторинг кількох процесів	Ні	Ні	Так	Ні	Ні	Ні
Можливість зупинки процесу	Так	Так	Так	Так	Так	Ні
Підтримка сервісів	Ні	Ні	Так	Так	Так	Ні
Підтримка Unicode	Так	Ні	Так	Ні	Так	Так
Вбудована мова сценаріїв	Ні	Так				Так
Підтримка плагінів	Ні	Ні	Ні	Так	Так	Так
Стабільність	Висока	Низька	Висока	Висока		Висока
Швидкість	Висока	Низька	Дуже Висока	Середня		Висока
Зручність користування	Середня	Середня	Висока	Середня+	Висока	Низька

Таблиця 4.3.

Порівняльна таблиця оцінки інструментів для моніторингу за критеріями доступу до інтерфейсу користувача

Показник Функціональність	Інструменти					
	MS Spy++ v11	Blade API Monitor v3.8	API Monitor v2 Alpha	WinApiOverride v6.4	Process Hacker 2	API Logger 1.9
1	2	3	4	5	6	7
Перегляд структури графічних об'єктів ОС	Так					
Перегляд компонентів в кастомних .dll	Частково*	Так	Так	Так		Так
Перегляд потоків процесу		Так				
Перегляд підключених dll		Так	Так	Так		Так
Перегляд дескрипторів процесу		Ні	Так	Так		Ні
Перегляд статичної інформації		Так	Так	Так		Так
Перегляд стеку			Так*	Так	Так	
Перегляд реєстрів процесора	Ні	Ні	Так	Ні	Ні	Ні
Перегляд пам'яті процесу	Так	Так	Так	Так	Так	Ні
Запис результатів в файл	Ні	Ні	Так	Так	Так	Ні
Представлення результатів моніторингу в інтерфейсі програми	Так	Ні	Так	Ні	Так	Так
Пошук процесу за вікном	Ні	Так				Так
Зручність користування	Ні	Ні	Ні	Так	Так	Так
Фільтр по типу	Висока	Низька	Висока	Висока		Висока
Фільтр по бібліотеці	Висока	Низька	Дуже Висока	Середня		Висока

У якості критеріїв була зроблена увага на: можливості моніторингу подій від типових ІК операційної системи (кнопки, меню та інше) через переривання API функцій; моніторинг об'єктів із бібліотек MFC, .NET, об'єктів ActiveX та COM; можливість моніторингу кількох процесів; підтримка опрацювання подій сервісів; наявність мова сценаріїв для часткової автоматизації процесом

моніторингу; можливість написання власних доповнень (плагінів); швидкодія перехоплення подій користувача інтерфейсної складової.

Застосувавши за аналогією формулу (2.31) для критеріїв із табл. 2.4 та із табл. 4.2, 4.3 отримаємо наступну загальну оцінку вибраних інструментів (рис. 4.12).

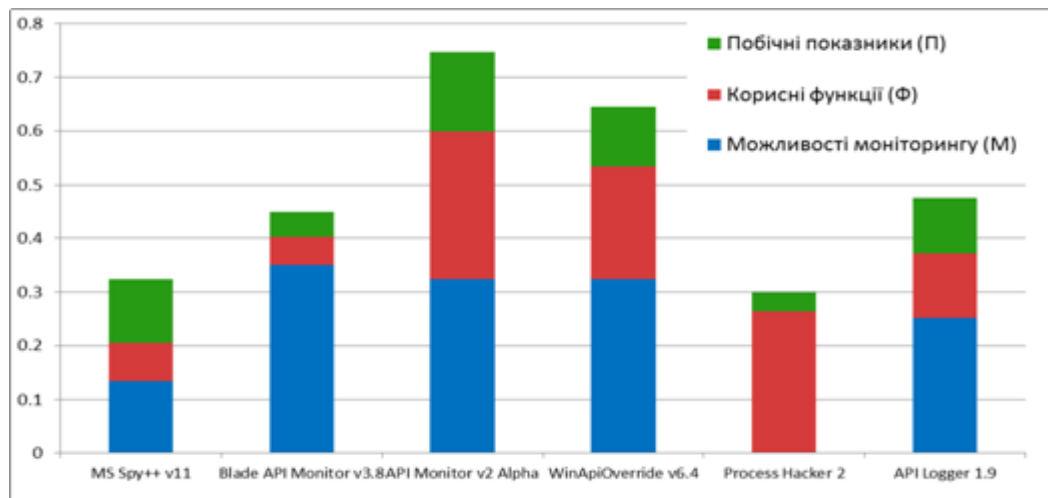


Рис. 4.12. Загальна оцінка досліджених інструментів моніторингу ІК

Отже, було проведено порівняльний аналіз існуючих інструментів, що використовуються для моніторингу та перехоплення викликів бібліотечних функцій та подій операційної системи. Було визначено критерії для оцінки, коефіцієнти їх важливості та, як наслідок, розраховано загальну оцінку для кожного інструменту. Результати експертної оцінки інструментів є цінними для використання у процесі синтезування інтерфейсної складової ПЗ, оскільки вони показують, якими інструментами ефективніше користуватись. З розглянутих інструментів, доцільно застосовувати API Monitor v2 Alpha від компанії Rohitab Batra, USA; незначно відстає від нього WinApiOverride v6.4 від Jacquelin POTIER, FR. Інші інструменти показали недостатній результат.

Цими інструментами доцільно досліджувати інтерактивні взаємодії користувачів із САПР чи іншим ПЗ, що дасть розуміння, в який момент відображати контекстну допомогу; коли користувач робить «перерви» в роботі

й треба підказати спосіб інтерактиву з ІК. Отримані дані моніторингу надалі передаються в систему накопичення даних взаємодій і слугують коефіцієнтами для патернів синтезування програмного коду оновлених інтерфейсів користувача.

4.3. Діалогове прототипування сценарії взаємодій користувача

Розробка ефективних діалогових підсистем [17] є комплексною проблемою, яка містить у собі аналіз і типізацію інформаційних вимог користувачів, синтез типової моделі діалогу для заданої множини користувачів, інформаційні запити яких належать одній предметній області, синтез інформаційного та прикладного програмного забезпечення.

Перед розробкою діалогової підсистеми необхідно виконати наступні попередні етапи: 1) ідентифікувати поняття, об'єкти та величини, визначити логічні зв'язки між поняттями з тим, щоб сформуванню інформаційну базу діалогових задач; 2) розробити сценарії діалогових процесів заданої задачі; 3) розробити алгоритми, що реалізують сформований діалоговий процес.

Побудова моделі діалогової системи дозволяє: 1) адекватно та одноманітно описати концепцію організації та функціонування діалогової системи, тобто пов'язати та представити з єдиних позицій структурні, динамічні та інформаційні аспекти діалогової взаємодії; 2) використати модель як основу для прототипування та реалізації програмного забезпечення діалогової системи; 3) провести аналіз та оцінку діалогової системи. Найбільш трудомістким за методикою етапом є розробка сценарію діалогового процесу.

4.4. Побудова сценарію діалогів для графічного редактора схем ІК

Для прикладу, побудову інтерактивної графічної діалогової підсистеми у вигляді графічного редактору принципів схем інтерфейсів користувача Schematic [10, 11] можна розглядати як процес поступового розбиття узагальнених структур на елементарні складові.

Відправною точкою для виділення основних конкретних об'єктів проектування є розробка загального сценарію діалогової підсистеми. Сценарій діалогу – детальний опис структури та змісту діалогу, достатній для його реалізації. Сценарій [12, 13] створеної підсистеми можна представити діаграмою переходів діалогу графічного редактора, яка зображена на рис. 4.13.

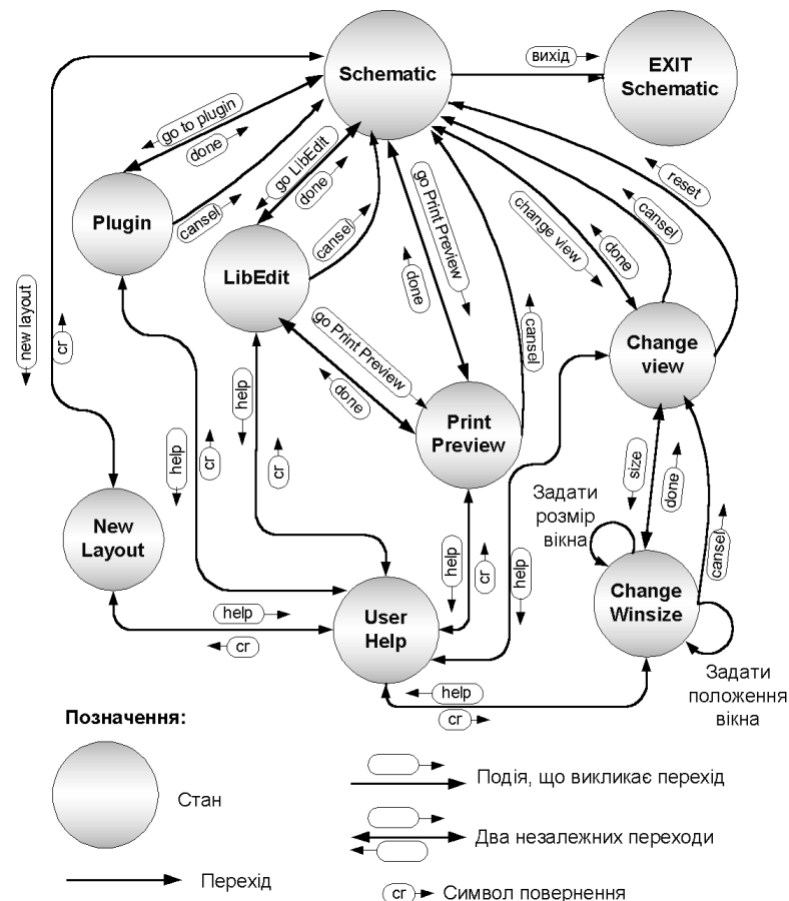


Рис. 4.13. Діаграма переходів сценарію діалогу графічного редактора схем ІК

Загальний сценарій (рис. 4.13) є інтегрованим сценарієм локальних підсценаріїв (рис. 4.14, 4.15, 4.16) діалогу, що будуються на основі схем розв'язання задачі в діалозі, і які доповнюються формами представлення вхідних даних та результатів. Структура сценарію діалогів САПР розроблялась з використанням об'єктно-орієнтованого підходу. Робота користувача за комп'ютером полягає в цілеспрямованій зміні деякого об'єкта, який має внутрішню структуру, певний зміст та зовнішнє символічне чи графічне

представлення. Об'єкт при цьому розуміється в широкому змісті слова. Користувач має можливість створювати світ об'єктів, використовуючи його, впливати безпосередньо на окремі об'єкти, змінюючи їхні атрибути та зв'язки з іншими об'єктами.

Структура сценарію діалогу (рис. 4.14) відповідає закладеній метафорі в створювану підсистему.

Сценарій діалогу для інтерактивної роботи користувачів з графічною схемою є складовою частиною сценарію організації діалогу всього графічного редактору схем інтерфейсів користувача.

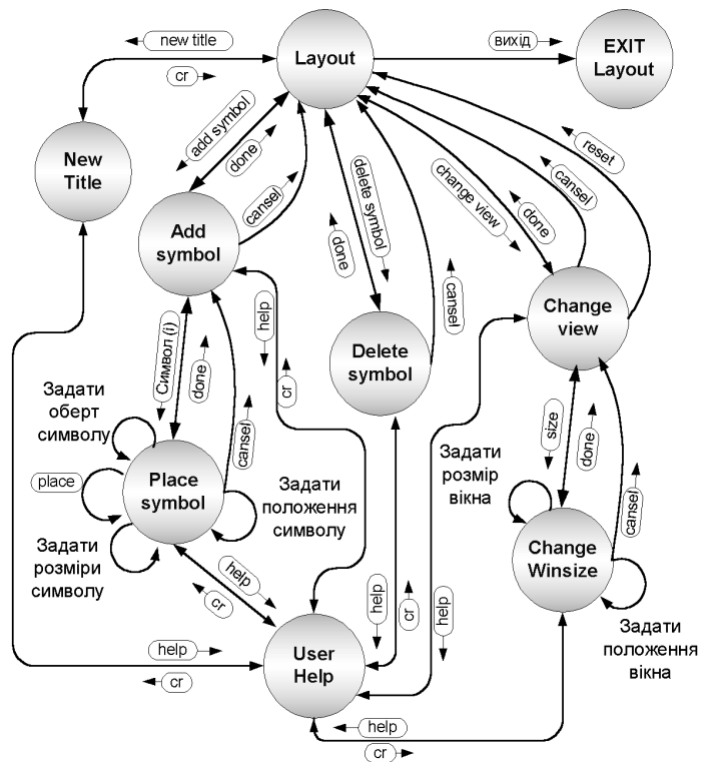


Рис. 4.14. Діаграма переходів сценарію діалогу для роботи з графічною схемою

Сценарій діалогу редактора бібліотек базових елементів схеми описує події та операції, які використані при розробці програмного модуля для організації роботи з базою даних, що містить інформацію про умовні графічні позначання елементів на принципових схемах графічного інтерфейсу користувача.

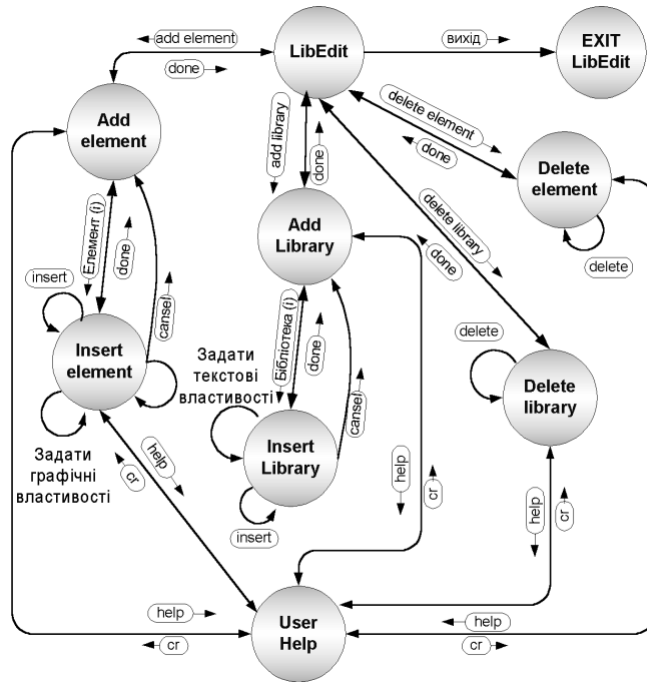


Рис. 4.15. Діаграма переходів сценарію діалогу для редактора бібліотек

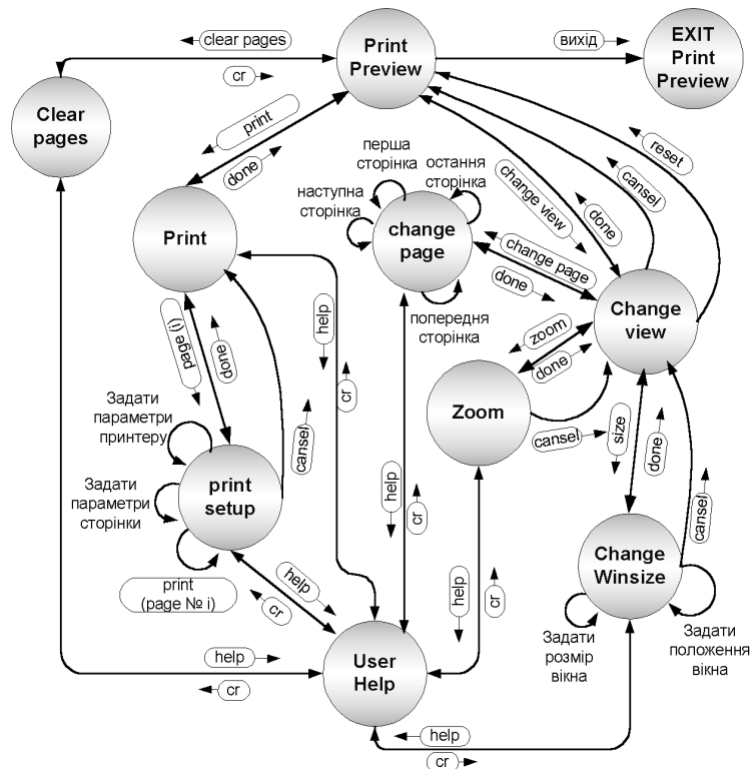


Рис. 4.16. Діаграма переходів сценарію діалогу для попереднього перегляду друку

Формування серії діаграм (рис. 4.13, 4.14, 4.15, 4.16) переходів сценаріїв діалогу дозволяє задати конструктиви інтерфейсів користувача як вікна, форми введення даних, спливаючі вінка, які у спою чергу містять наповнення кнопками, списками, написами та іншими типовими конструктивами ІК (). Це, так би мовити, верхній рівень абстракції для генерування класів програмного коду для контейнерів вмісту інтерфейсів.

Підсистема синтезу діалогових взаємодій конструктивів ІК використовує метод проектування «Матриця зв'язності» [15], який дозволяє забезпечити систематичний пошук взаємозв'язків між елементами в межах даної проблеми.

«Взаємозв'язок» визначається як необхідність передачі інформації. Результати [12, 13] використання методу наведено у табл. 4.4, де основні елементи позначаються відповідними номерами, а потреба у зв'язку оцінюється за двобальною системою: 1 – необхідний зв'язок; 0 – зайвий зв'язок. Матриця взаємодії є несиметричною, наявність або відсутність зв'язку визначається напрямком впливу одного елемента на інший.

Таблиця 4.4.

Матриця взаємодій об'єктів сценарію діалогів користувача

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0
3	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
4	1	0	0	0	0	1	1	1	1	0	0	1	0	0	0	0	0	0	0	0
5	0	0	1	0	0	1	1	1	1	1	1	0	1	1	1	1	0	1	0	
6	0	0	1	1	1	0	1	1	1	0	0	0	0	0	1	0	0	0	1	
7	0	0	1	1	0	1	0	0	0	0	0	0	0	0	1	0	0	1	1	
8	0	0	1	1	1	1	0	0	0	0	0	0	0	0	1	0	0	1	1	
9	0	0	1	1	0	1	0	0	0	0	1	1	0	0	1	0	0	1	1	
10	0	0	1	0	1	0	0	0	0	0	1	1	0	0	1	0	0	0	0	
11	0	0	1	0	1	0	0	0	1	1	0	1	0	0	1	0	0	1	1	
12	0	0	1	1	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	
13	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	
14	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	
15	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	
16	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	
17	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0	1	
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
20	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1	1	0	1	1	

Це матриця (для прикладу) заповнена зв'язками діалогів користувача з інтерфейсами ПЗ графічного редактора принципів схем інтерфейсів для системи моделювання. При цьому можна виділити такі основні елементи програмного модуля побудови схеми: 1) текстовий файл опису принципової схеми на проблемно орієнтованій мові моделювання схеми; 2) база даних; 3) файл графічної принципової схеми; 4) компілятор з графічного представлення схеми у текстове; 5) робоче поле графічного креслення – “макетниця”; 6) візуальний елемент ІК; 7) номінальні параметри візуального елемента схеми ІК; 8) назва візуального елемента схеми; 9) контактні точки візуального елемента схеми; 10) елемент з'єднання провідників Point; 11) провідники Line; 12) вузли схеми; 13) текстовий напис; 14) графічний рисунок; 15) аркуш документу; 16) поля документу; 17) вікно модуля побудови схеми; 18) панелі інструментів, меню, стрічка стану, навігатор бібліотек; 19) курсор; 20) растр та інше.

4.5. Перевірка автоматизації ідентифікації компонентів інтерфейсів користувача із графічних ескізів

Графічний ескіз користувацького інтерфейсу, створеного в растровому графічному редакторі або сфотографованого з паперу необхідно зберегти у вигляді файлу з розширенням PNG, який надалі завантажується у систему як джерело вхідних даних. Надалі опрацювання файлу здійснюється за описаною в підрозділі 3.3 процедурою [119]. Після завантаження файлів графічні ескізи попередньо обробляються, щоб уникнути спотворення зображення, які можливо з'явилися у процесі фотографування або недосконалості графічних редакторів. Попередньо вводиться балансування контрастності й яскравості по всіх регіонах зображення і здійснюються геометричні перетворення: зміщення, вирівнювання, корекція трапеції, поворот на кут. На рис. 4.17 показаний тестовий приклад растрового зображення з контурами конструктивів ІК.

Рис. 4.17. Растровий приклад тестового графічного прототипу віконної форми інтерфейсу

Система використовує програмні об'єктні класи із бібліотеки OpenCV [121] для розпізнавання контурів шаблонних конструктивів інтерфейсу, а програмна бібліотека класів Tesseract [122] використана для розпізнавання тексту.

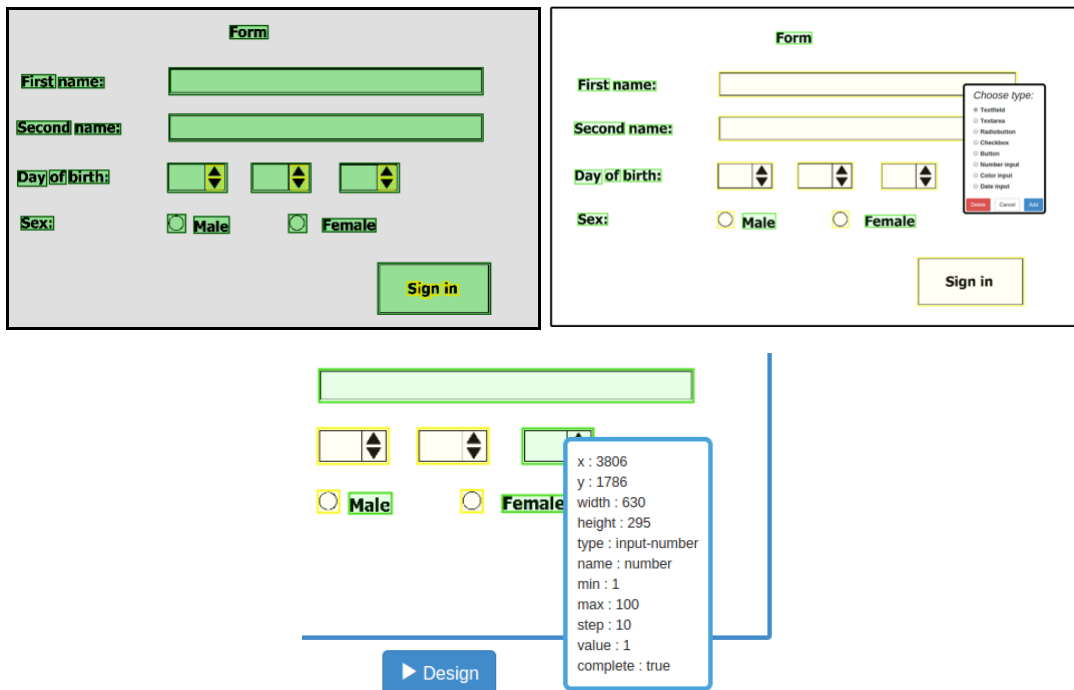


Рис. 4.18. Виділення автоматично сегментованих позиціонування тексту та контурів ГІК

Регіони з розпізнаного тексту і контури конструктивних елементів користувацького інтерфейсу (значення позицій, розмірів) відзначені різнокольоровими прямокутниками після стадії визначення (приклад на рис. 4.18). Натискання курсором миші на ці підсвічені області активує спливаюче меню, в якому користувач-проектант інтерфейсів може вибрати тип елемента (наприклад: текстове поле, радіокнопки «RatioButton», прапорець «CheckBox», кнопки «Button» тощо), якщо автоматизовано вони не точно визначились, і може встановити додаткові атрибути для цього елемента у ручному режимі. На етапі результуючого відображення автоматизовано ідентифікованих компонентів доступна функція перетягування елементів і додавання візуальних стилів оформлення Bootstrap CSS [123] для програмного прототипу.

На виході [120] система генерує HTML код (приклад рис. 4.19) програмного прототипу інтерфейсної віконної форми ІК, який можна скопіювати в буфер обміну або завантажити на жорсткий диск у вигляді файлу з розширенням HTML. Користувач може зверстати отриманий код за шаблоном стандарту HTML5 з використанням патернів кодування і надалі завантажити прототип у вигляді веб-сторінки сайту.

```

<!DOCTYPE html>
<html lang='en'>
  <head>
    <meta charset='UTF-8'>
    <title>Generated</title>
  </head>
  <body>
    <span style="position: absolute; top: 26px; left: 342px; ">Form</span>
    <span style="position: absolute; top: 103px; left: 36px; ">First name:</span>
    <input type="text" style="width: 451px; height: 36px; ">
    <span style="position: absolute; top: 171px; left: 32px; ">Second name:</span>
    <input type="text" style="width: 451px; height: 36px; ">
    <span style="position: absolute; top: 243px; left: 33px; ">Day of birth:</span>
    <input type="number" style="width: 83px; height: 39px; ">
    <span style="position: absolute; top: 307px; left: 33px; ">Sex:</span>
    <input type="radio" style="width: 23px; height: 23px; "> Male
    <input type="radio" style="width: 23px; height: 23px; "> Female
    <button style="width: 160px; height: 71px; ">Sign in</button>
  </body>
</html>

```

Рис. 4.19. Приклад автоматизовано згенерованого програмного коду прототипу інтерфейсної форми.

4.6. Висновки до розділу

1. Результати статистичного аналізу спектру конструктивів ГІК в ПЗ САПР інтегровані в репозиторій критеріїв оцінки інтерфейсів користувачів для формування коефіцієнтів компонування макетів поля ГІК.
2. Коректність виконаних експериментів автоматизації ідентифікування візуальних прототипів графічних інтерфейсів користувача з растрових зображень підтверджується малими розбіжностями до 5% (для конструктивів з прямокутними контурами і текстових написів) між візуальним відображенням інтерфейсу у графічному файлі на вході системи та візуальним відображенням інтерфейсу при відображенні вихідних файлів з розміткою інтерфейсу у інтерпретаційних програмах, що отримуються на виході системи.

Висновки

У роботі вирішено науково-технічне завдання вдосконалення підходів, методів, процесів та розроблення інструментальних інформаційних засобів для автоматизації синтезу прототипів інтерфейсів користувача комп'ютерних систем із використанням патернів програмних кодів компонентів графічного інтерфейсу користувача. Це дає можливість пришвидшити виробничі процеси програмістів, тестерів програмного забезпечення та інформаційних архітекторів на рівні інтерфейсів користувача.

При цьому отримані такі основні результати:

1. Проаналізовано існуючий стан концепцій, підходів, типів та видів способів у галузі проектування інтерфейсів користувачів на різних етапах життєвого циклу проектування. Здійснено критичний порівняльний аналіз існуючих інформаційних технологій та спеціалізованих мов розмітки для програмування інтерфейсів користувача.
2. Сформульовано ієрархічну систематизацію користувацьких взаємодій та кінематичних форм організації інтерфейсів користувача з апаратно-програмними засобами. Така систематизація у вигляді ієрархічного дерева дозволяє загалом відобразити рівні сутностей інтерфейсів користувача та їхньої діяльності в інформаційних системах, і як наслідок, дозволяє зменшити тривалість обрання інформаційних архітектурних рішень проектувальником на етапах планування програмних продуктів.
3. Сформульовано систематизацію конструктивних елементів графічного інтерфейсу користувача, що дозволяє спростити кількість кроків визначення необхідного компонента розробником програмного забезпечення на етапах ескізного прототипування інтерфейсів

користувача, спираючись на опис параметрів і функціонування певних компонентів. Проведено експериментальні емпіричні та статистичні дослідження для перевірки повноти типів наявних компонентів графічного інтерфейсу користувача в існуючих САПР. Статистичні дані вказують на переважаюче застосування проектувальниками САПР лише 10-15 основних типів компонентів із всього систематизованого спектру, що вказує на необхідність впровадження новітніх конструктивів інтерфейсу у програмні продукти протягом реінженірінгу.

4. Формалізовано метод моделювання досліду користувача на профілі площинного компоунування та формалізовано процес поділу поля макетів каркасів компоунування графічного інтерфейсу користувача.
5. Сформульовано критерії та отримано: метод оцінювання засобів прототипування графічного інтерфейсу користувача, метод оцінювання інформаційних інструментів моніторингу подій графічного інтерфейсу користувача програмного забезпечення, метод оцінювання мов декларування програмних кодів інтерфейсів користувача, що сприяє зменшенню виробничої тривалості для прийняття рішення з обрання відповідного засобу чи програмної технології на рівні концептуального проектування інтерфейсів користувача.
6. Запропоновано та обґрунтовано інформаційну архітектуру системи, яка дала можливість інтегрувати дані репозиторія метрик інтерактивного використання графічного інтерфейсу користувача із даними репозиторія програмних кодів, автоматизації фази визначення габаритно-візуальних параметрів компоунування елементів графічного інтерфейсу користувача під час створення чи модифікування програмного продукту.
7. Запропоновано інформаційну архітектуру підсистеми реінженірінгу програмних кодів декларування інтерфейсів користувача програмного

забезпечення, що дозволяє покращити продуктивність програмістів на етапі оновлення засобів інтерфейсного доступу до функціоналу існуючих програмних засобів.

8. Розроблено інформаційне, лінгвістичне та програмне забезпечення у вигляді програмно-методичного комплексу інструментальних інформаційних засобів для накопичення даних взаємодій, даних статистичних оцінок, даних патернів програмних кодів, даних патернів макетування площинного компонування, засобу синтезування сценаріїв діалогів інтерфейсів, засобу візуального ідентифікування компонентів та подальшого синтезування програмних кодів.
9. Результати впроваджено в навчальний процес кафедри «Системи автоматизованого проектування» Національного університету «Львівська політехніка» у курс «Методи побудови інтелектуального інтерфейсу користувача в автоматизованому проектуванні» для студентів спеціальності 7.05010103 / 8.05010103 «Системне проектування» (спеціалісти та магістри) та використовуються у діяльності програмістів й тестерів програмного забезпечення в ТзОВ «Едвантіс» (м.Львів, Україна).

Список використаних джерел

1. Дженифер Тидвелл, Разработка пользовательских интерфейсов. 2-е издание / Дженифер Тидвелл // Питер, 2011, 480 с., ISBN:978-5-459-00434-2
2. Билл Скотт, Проектирование веб-интерфейсов / Билл Скотт, Тереза Нейл // Символ-Плюс, 2010, 352 с., ISBN 978-5-93286-172-1, 978-0-596-51625-3
3. Клименко Станислав, Графические интерфейсы и средства их разработки / Клименко Станислав, Уразметов Василь // Индустрия программирования, 1-6 октября 1996 г., Москва
4. Моделі, методи та засоби підвищення ефективності інтерфейсу "користувач - ЕОМ" у системах організаційного управління: Автореф. дис... канд. техн. наук: 05.13.06 / Л.М. Радванська; Херсон. держ. техн. ун-т. — Херсон, 1999. — 17 с. — укр.
5. Моделі, методи та засоби адаптивності користувальницького інтерфейсу: Автореф. дис... канд. техн. наук: 05.13.06 / Д.В. Ходаков; Херсон. держ. техн. ун-т. — Херсон, 2003. — 19 с. — укр.
6. Jeffrey Nichols, Andrew Faulring, Automatic Interface Generation and Future User Interface Tools // Language Learning & Technology Refereed Journal, volum 2, num 1, 2006.
7. Engineering the User Interface: From Research to Practice, VII International Conference on Interacción Persona-Ordenador (IPO), Human-Computer Interaction in Spanish, which took place in Puertollano (Ciudad Real) on November 13th-17th, 2006, Springer; 1st ed. 2009. Corr. 2nd printing edition (May 7, 2009), -277 p. ISBN-13: 978-1848001350

8. Ажеронок В. А., Разработка управляемого интерфейса / В. А. Ажеронок, А. В. Островерх, М. Г. Радченко, Е. Ю. Хрусталева // 1С-Пабблишинг, 2010, 728 с., ISBN: 978-5-9677-1148-0
9. Алан Купер, Об интерфейсе. Основы проектирования взаимодействия / Алан Купер, Роберт Рейман, Дэвид Кронин // Символ-Плюс, Серия: Профессионально, 2009, 688 с., ISBN 978-5-93286-132-5, 978- 0-470-08411-3
10. Близнюк М. Аналіз підходів до розробки діалогових систем та їх реалізація в інтерактивному графічному редакторі схем системи “Micro-PC” / Близнюк М. , Маркелов О. // Вісн. Нац. ун-ту “Львів. політехніка”. - 2001. № 415: Комп’ютер. системи проектув. Теорія і практика. - С. 75-87. - Бібліогр.: 12 назв.
11. Blyzniuk M. Analysis of the approaches to dialogue system development and their realisation in interactive graphical schematic editor of “Micro-PC” circuit simulator / Blyzniuk M. , Markelov O. // The experience of designing and application of CAD systems in microelectronics: Proc. of the VI-th Intern. conf. CADSM-2001, 12-17 Febr. 2001, Lviv, Slavsko, Ukraine, - Lviv, 2001. - P.94-95. - Bibliogr.: 5 titles. - Парал. тит. арк. укр.
12. Маркелов О.Е. Розробка сценаріїв діалогів графічного редактора електричних схем системи схемотехнічного проектування “Micro-PC” / О.Е. Маркелов // Вісн. Нац. ун-ту “Львів. політехніка”. - 2002. - № 443: Радіоелектрон. та телекомунікації. - С. 193-199. - Бібліогр.: 7 назв.
13. Markelov O. The development of the dialofue scripts of the graphical schematic editor of “Micro-PC” circuit simulator / Oleksandr Markelov // Сучасні проблеми радіоелектроніки, телекомунікацій, комп’ютерної інженерії: Матеріали Міжнар. конф. TCSET’2002, 18-23 лют. 2002, Львів-Славскоб Україна. - Л.: Вид-во Нац.ун-ту “Львів. політехніка”, 2002. - С.307-309. - Bibliogr.: 6 titles. - Парал. тит. арк. англ.

14. Разработка САПР. В 10 кн. Кн. 5. Организация диалога в САПР: Практ. пособие / В.И.Артемьев, В.Ю.Строганов.; Под ред. А.В.Петрова. — М., 1990.
15. Джонс Дж.К. Методы проектирования: Пер. с англ. 2-е изд., доп. — М.: Мир,1986.— 326 с.
16. Диалоговые системы и представление знаний: Спр. пособие /Л.В.Кокорева, О.Л.Перевозчикова, Е.Л.Ющенко. — К., 1993.
17. Мамикопов А.Г. Анализ диалоговых систем (модели и методы) /А.Г.Мамикопов, В.В.Кульба, С.А.Косяченко и др. — М., 1986.
18. Диалоговые системы схемотехнического проектирования / В.И.Анисимов, Г.Д.Дмитревич, К.Б.Скобельцын и др.; Под ред. В.И.Анисимова. — М., 1988.
19. Oleksandr Markelov, The Software User Interface Interactions: Systematized Review / Oleksandr Markelov // Computer Science and Engineering: Proceedings of the IV-th International Conference of Young Scientists CSE-2010, November 25-27: 2-nd International Youth Science Festival, - Lviv, Ukraine, Publishing House of Lviv Polytechnic National University, - 408 с., 26-27 с. ISBN: 978-966-553-999-5.
20. Маркелов О.Е. Класифікація інтерактивних взаємодій користувача з програмним забезпеченням / Маркелов О.Е. // Вісн. Нац. ун-ту “Львів. політехніка” - № 711: Комп’ютер. системи проектув. Теорія і практика:[зб. наук. пр.] / відп. ред. М. В. Лобур. — Л., 2011. -176 с. - С. 15-21. - Бібліогр.: 19 назв.; - ISBN: 0321-0499
21. Методи побудови інтелектуального інтерфейсу користувача в автоматизованому проектуванні : конспект лекцій для студентів спец. 7.05010103/8.05010103 "Систем. проектування" / Маркелов Олександр Едуардович // М-во освіти і науки України, Нац. ун-т "Львів. політехніка", Ч. 1 . -Львів : Вид-во Львів. політехніки, 2014 . -92 с.-Бібліогр.: с. 85-91 (106 назв) . -14,00 Шифр: 004.5(075.8) Авторський знак: M266

22. Anand Agarawala, Ravin Balakrishnan, Keepin' It Real: Pushing the Desktop Metaphor with Physics, Piles and the Pen in BumpTop [Electronic resource] // Dynamic Graphics Project, Mode of access: <http://www.dgp.toronto.edu>, <http://www.youtube.com/watch?v=B6z2DWXmRkg> / – Last access: 2011-Sep-02
23. Light Touch™ instantly turns any flat surface into a touch screen [Electronic resource] Mode of access: <http://lightblueoptics.com/products/light-touch/>
24. Pranav Mistry, SixthSense – A Wearable Gestural Interface [Electronic resource] // Mode of access: <http://www.pranavmistry.com/projects/sixthsense/>– Last access: 2010-Sep-01. – Title from the screen.
25. Hiroshi Ishii, The tangible user interface and its evolution, Research on Organic User Interfaces (OUI) [Electronic resource] / Hiroshi Ishii, — NY, USA, Association for Computing Machinery, Volume 51 , Issue 6 (June 2008), PP. 32-36, ISSN:0001-0782, Mode of access: <http://www.organicui.org/>
26. Reactable Systems [Electronic resource] / Pompeu Fabra University, Mode of access: <http://www.reactable.com>
27. Microsoft Surface [Electronic resource] // Microsoft Mode of access: <http://www.microsoft.com/surface/> – Last access: 2010-Sep-25. – Title from the screen.
28. High Sign - Increase Productivity Using Mouse Gestures [Electronic resource] Mode of access: <http://highsign.codeplex.com/>
29. Ubuntu Operating System [Electronic resource] // Ubuntu, Canonical Group Limited, Mode of access: <http://www.ubuntu.com/> – Last access: 2010-Sep-01. – Title from the screen.
30. Nintendo Wii [Electronic resource] Mode of access: <http://www.nintendo.com/wii>
31. Google Gmail Motion [Electronic resource] // Google Inc., Mode of access: <http://mail.google.com/mail/help/motion.html> – Last access: 2010-Sep-01.

32. Microsoft Xbox Kinect [Electronic resource] / Microsoft Corporation Mode of access: <http://www.xbox.com/kinect> – Last access: 2011-Sep-01. – Title from the screen.
33. Evan A. Suma, Flexible Action and Articulated Skeleton (FAAST) [Electronic resource] / Evan A. Suma, Belinda Lange, Skip Rizzo, David Krum, and Mark Bolas // University of Southern California, USC Institute for Creative Technologies, USC Interactive Media. Mode of access: <http://projects.ict.usc.edu/mxr/faast/> – Last access: 2010-Sep-01.
34. Samsung Galaxy Note 3 Review / Tim Schiesser // TechSpot, Inc., December 27, 2013 [Electronic resource] Mode of access: <http://www.techspot.com/review/759-galaxy-note-3/page3.html>
35. Microsoft Reveals 12-Inch Surface Pro 3 BY / Chloe Albanesius // PCMag Digital Group, May 20, 2014, [Electronic resource] Mode of access: http://www.pcmag.com/image_popup/0,1740,iid=418908,00.asp
36. Roberto Mateu, Introducing Opera Face Gestures [Electronic resource] / Roberto Mateu, Opera Software ASA, 01 Apr 2009, Mode of access: <http://labs.opera.com/news/2009/04/01/>
37. Zooming user interface [Electronic resource] // Wikipedia – the free encyclopedia, Mode of access: http://www.wikiwak.com/wak/Zooming_user_interface – Last access: 2011-Sep-02. – Title from the screen.
38. Deborah L. Stone, User interface design and evaluation / Deborah L. Stone, Debbie Stone // Morgan Kaufmann, 2005, 669 p., ISBN 0120884364
39. Google Maps [Electronic resource] // Google, Mode of access: <http://maps.google.com> – Last access: 2011-Sep-02. – Title from the screen.
40. Nanoscience - NRC Cambridge (video concept) // Nokia Conversations [Electronic resource] Mode of access: <http://youtu.be/PKihhDC7-bI> – Title from the screen.
41. Мандел Т. Разработка пользовательского интерфейса / Мандел Т., Пер. с англ. – М.: ДМК Пресс, 2001. – 416 с., ил.

42. Speaktoit // Palo Alto, California, USA [Electronic resource] Mode of access: <http://www.speaktoit.com/> – Last access: 2014-Sep-02. – Title from the screen.
43. Майкл Янг, Microsoft Office 97. Разработка Web-документов , 1998, Мир, 544 с. , ISBN: 0-7821-2075-X544
44. Pie menu. Wikipedia - the free encyclopedia [Электронный ресурс] Режим доступа: http://en.wikipedia.org/wiki/Pie_menu
45. Markelov Oleksandr. Kinetic User Interface for Human-Computer Interactions: Systematized Review / Oleksandr Markelov, Mykhaylo Lobur // CAD in Machinery Design. Implementation and Education Problems: Proceeding of the XVIII Ukrainian-Polish Conference CADMD'2010 (Oct 14-16, 2010). - Lviv, Ukraine: Publishing House Vezha&Co, 2010. -130 с., - PP.69-73, ISBN 978-966-2191-15-8
46. Usability Engineering [Electronic resource] Mode of access: <http://www.sherazstudio.com/images/spa.jpg> – Title from the screen.
47. What is User Experience (UX) //User Experience Professionals Association [Electronic resource] Mode of access: <https://uxpa.org/resources/about-ux> – Title from the screen.
48. Ahmed Seffah, Human-Centered Software Engineering - Integrating Usability in the Software Development Lifecycle: Volume 8 / Ahmed Seffah, Jan Gulliksen, Michel C. Desmarais // Published by Springer, 2005, Pages: 406, ISBN-101-4020-4027-X
49. Перевалов Я. М. Процесс эргономического проектирования программных систем / Перевалов Я. М. // Третий семинаре по Usability. Москва, 7 апреля 2001 г. [Электронный ресурс] Режим доступа: <http://www.usability.ru/Articles/ErgoProc.htm>
50. Embarcadero Delphi and C++Builder Quick Start Tutorial, Product Documentation, XE Release, PDF Last Published October 18, 2010, Mode of access: <http://www.embarcadero.com/products/rad-studio> – Last access: 2010-Dec-10. – Title from the screen.

51. Lazarus Documentation, Lazarus is a free cross-platform IDE, Mode of access: <http://www.lazarus.freepascal.org/> – Last access: 2010-Dec-12. – Title from the screen.
52. NetBeans IDE - free, open-source Integrated Development Environment for software developers, Swing GUI Builder Mode of access: <http://netbeans.org>. – Last access: 2010-Dec-20. – Title from the screen.
53. Microsoft Visual Studio 2010. Mode of access: <http://www.microsoft.com/visualstudio>. – Last access: 2010-Dec-24. – Title from the screen.
54. Oleksandr Markelov. Usage of MS Visual Studio WinForms Designer as Schematic Prototyping Tool for SolidWorks 3D: Experience of both API Integration / Oleksandr Markelov, Andrii Kovalchuk // CAD in Machinery Design. Implementation and Education Problems: Proceeding of the XXII Ukrainian-Polish Conference (CADMD'2014, October 10-11, Lviv, Ukraine), Publishing House Vezha&Co, — 192 p., 2014. — Lviv., PP.46-49, - 6 refer. ISBN: 978-617-607-644-5
55. Qt – cross-platform application and UI framework. Online Reference Documentation, Nokia Corporation. Mode of access: <http://qt.nokia.com/>. – Last access: 2010-Dec-27. – Title from the screen.
56. Microsoft Visio Professional. Mode of access: <http://www.microsoft.com> – Last access: 2012-Sep-10. – Title from the screen.
57. Evolus Pencil. Mode of access: <http://pencil.evolus.vn> – Last access: 2012-Sep-10. – Title from the screen.
58. Balsamiq Mockups. Mode of access: <http://www.balsamiq.com/products/mockups> – Last access: 2012-Sep-10. – Title from the screen.
59. Pidoco. Mode of access: <https://pidoco.com> – Last access: 2012-Sep-10. – Title from the screen.
60. Produle Systems Mockflow. Mode of access: <http://www.mockflow.com/> – Last access: 2012-Sep-10. – Title from the screen.

61. Rudrasoft MockupTiger. Mode of access: <http://www.mockuptiger.com/> – Last access: 2012-Sep-10. – Title from the screen.
62. Cyberopsis Webdata Proto.IO. Mode of access: <https://proto.io> – Last access: 2012-Sep-10. – Title from the screen.
63. Site9 Inc. ProtoShare. Mode of access: <http://www.protoshare.com> – Last access: 2012-Sep-10. – Title from the screen.
64. Axure Software Solutions. Mode of access: <http://www.axure.com> – Last access: 2012-Sep-10. – Title from the screen.
65. Tool Software UI Mockup. Mode of access: <http://www.toolsoft.net/ui-mockup.html> – Last access: 2012-Sep-10. – Title from the screen
66. ERGOSIGN Technologies Antetype. Mode of access: <http://www.antetype.com> – Last access: 2012-Sep-10. – Title from the screen.
67. Endloop Systems iMockups. Mode of access: <http://www.endloop.ca/imockups> – Last access: 2012-Sep-10. – Title from the screen.
68. CATCH Screen Architect. Mode of access: <http://www.catchlimited.com/index.php?page=screen-architect> – Last access: 2012-Sep-10. – Title from the screen
69. iPhone MockUp / Lukas Mathis Mode of access: <http://iphonemockup.lkmc.ch> – Last access: 2012-Sep-10. – Title from the screen
70. UICloud. A project by Double-J Design. Mode of access: <http://ui-cloud.com> – Last access: 2012-Sep-10. – Title from the screen
71. Protosketch [Электронный ресурс] URL: <https://protosketch.uistencils.com/> Last access: 2014-Oct-10. – Title from the screen
72. Napkee - make your mockups come alive [Электронный ресурс] URL: <http://www.napkee.com/> Last access: 2012-Sep-10. – Title from the screen
73. wire2app – Beta [Электронный ресурс] URL: <http://www.wire2app.com/> Last access: 2012-Sep-10. – Title from the screen

74. Display resolution // Wikipedia – the free encyclopedia [Електронний ресурс]: – Режим доступу: http://en.wikipedia.org/wiki/Display_resolution Назва з екрану. – Дата звернення: 22.04.2014.
75. List of displays by pixel density // Wikipedia – the free encyclopedia [Електронний ресурс]: – Режим доступу: http://en.wikipedia.org/wiki/List_of_displays_by_pixel_density Назва з екрану. – Дата звернення: 22.04.2014.
76. Маркелов О.Е. Систематизація площинних шаблонів компонування груп елементів користувацького інтерфейсу / Маркелов О.Е., Ковальчук А.М. // Сучасні комп'ютерні інформаційні технології: матеріали IV Всеукр. школи-семінару молодих вчен. і студ. (АСІТ' 2014), 16-17 трав. 2014, Тернопіль, Україна/М-во освіти і науки України, Терноп. нац. екон. ун-т, Ф-т комп'ютер. інформ. технологій, Асоц. фахівців комп'ютер. інформ. технологій, -Тернопіль., 2014 .-С. 131-132 .- Бібліогр.: 3 назв. - ISBN: 978-966-654-343-4
77. Markelov Oleksandr. Analytical Technique for Selection a Tool to User Interface Prototyping / Oleksandr Markelov // VII-th International Scientific and Technical Conference “Computer Sciences and Information Technologies”(CSIT'2012, Lviv, UKRAINE, November 20-24), 2012 — 223 P., — Lviv, PP. 80-81, - 2 refer. ISBN: 978-617-607-352-9
78. Шейко Дмитрий, Языки описания пользовательских интерфейсов. с.15, 08 марта 2005 г. [Електронний ресурс], URL:<http://www.cmsdevelopment.com> , Дата звернення: 15.12.2009
79. User Interface Markup Language: Draft specification for language version 3.1 of March 2004 // Marc Abrams, James Helms, Organization for the Advancement of Structured Information Standards (OASIS) UIML Technical Committee, [Електронний ресурс], URL: <http://www.uiml.org> , Дата звернення: 15.12.2009

80. Proceedings of the Fourth Intl Conference on Computer-Aided Design of User Interfaces, held May 17, 2002 in Valenciennes, France, Springer; 1 edition (April 30, 2002), -400 p., ISBN-13: 978-1402006432
81. Computer-Aided Design of User Interfaces V: Proceedings of the Sixth International Conference on Computer-Aided Design of User Interfaces CADUI '06 (6-8 June 2006, Bucharest, Romania), Springer; 1 edition (December 10, 2007), -298 p. ISBN-13: 978-1402058196
82. Human-Centered Software Engineering: Software Engineering Models, Patterns and Architectures for HCI (Human-Computer Interaction Series), Springer; 1 edition (June 22, 2009), -398 p., ISBN-13: 978-1848009066
83. Vaughn Bullard, Kevin T. Smith, Michael C. Daconta, Essential XUL Programming, John Wiley & Sons, 2001, 448 p. ISBN-13: 978-0471415800
84. Kenneth C. Feldt, Programming Firefox: Building Rich Internet Applications with XUL, O'Reilly Media, 2007, - 494 p. ISBN-13: 978-0596102432
85. Jurgen Schumacher, Markus Stäuble, ZK Developer's Guide, Packt Publishing (March 28, 2008), -187 p. ISBN-13: 978-1847192004
86. Mozilla Developer Center. XUL Documentation. Mozilla Foundation. [Электронный ресурс], URL: <https://developer.mozilla.org/En/XUL> , Дата звернення: 15.12.2009
87. Extensible Application Markup Language. Microsoft Developer Network. [Электронный ресурс], URL: <http://msdn.microsoft.com/en-us/library/ms747122.aspx>, Дата звернення: 15.12.2009
88. Applications = Code + Markup: A Guide to the Microsoft Windows Presentation Foundation (Pro - Developer), Microsoft Press; 1 edition (September 13, 2006), -1020 p. ISBN-13: 978-0735619579
89. Brennon Williams , Microsoft Expression Blend Unleashed, Sams Publishing; 1 edition (June 13, 2008), -600p., ISBN-13: 978-0672329319

90. Flex Interface Guide. Flex Development Center. Adobe Developer Connection. [Електронний ресурс], - Режим доступу: <http://www.adobe.com/devnet/flex/>, Дата звернення: 15.12.2009
91. Adobe Flex v.3.4 Language Reference. Flex Development Center. Adobe Developer Connection. [Електронний ресурс], URL: <http://livedocs.adobe.com/flex/3/langref/overview.html>, Дата звернення: 15.12.2009
92. Michele E. Davis, Jon A. Phillips, Flex 3: A Beginner's Guide, McGraw-Hill Osborne Media; 1 edition (March 21, 2008), -274p., ISBN-13: 978-0071544184
93. Charles E. Brown , The Essential Guide to Flex 3, friends of ED (May 26, 2008), -600 p., ISBN-13: 978-1590599501
94. Abstract User Interface Markup Language Toolkit. Development team at the IBM Rochester lab, February 16, 2004. [Електронний ресурс], URL: <http://www.alphaworks.ibm.com/tech/auiml>, Дата звернення: 15.12.2009
95. Nexaweb Developer Guide, Nexaweb Dev Center, February 2009 Edition, [Електронний ресурс], URL: <http://www.openxal.org/>, Дата звернення: 15.12.2009
96. An XML Framework for Rich Internet Applications, Laszlo Systems Technology White Paper, July, 2005, [Електронний ресурс], URL:<http://www.laszlo.com>, Дата звернення: 15.12.2009
97. Hierarchical Model View Controller User Interface Language (HMVCUL) [Електронний ресурс], URL:<http://www.priorartdatabase.com/IPCOM/000160727/>, Дата звернення: 15.12.2009
98. UIDL: User Interface Description Language, Reference Documentation. [Електронний ресурс], URL:<http://www.uidl.net/>, Дата звернення: 15.12.2009
99. Lobur M. UI Technologies for CAD Software Development: Evaluative Approach of Componentry Applications. / M. Lobur, O. Markelov // Сучасні проблеми радіоелектроніки, телекомунікацій, комп'ютерної інженерії / Матеріали Міжнар. конф. TCSET'2010, 23-27 лют. 2010, Львів-Славсьько,

- Україна ; Нац. ун-т "Львів. політехніка". - Л. : Вид-во Нац. ун-ту "Львів. політехніка", 2010. - 380 с. : іл. - Парал. тит. арк. англ. - С. 41
100. Лобур М. Аналіз мов опису користувачьких інтерфейсів для застосування у програмному забезпеченні САПР / Лобур М., Маркелов О., Бобало С. // Вісн. Нац. ун-ту "Львів. політехніка"ю - 2010. - № 663: Комп'ютерні науки та інформаційні технології. - С. 30-38 . - Бібліогр.: 25 назв
 101. Лозинський А.Я. Фактори впливу на параметри людино-технічних, людино-інформаційних комплексів / Лозинський А.Я., Маркелов О.Е., Іванців Р.-А.Д. // Сучасні комп'ютерні інформаційні технології : матеріали IV Всеукр. школи-семінару молодих вчен. і студ. (АСІТ' 2014), 16-17 трав. 2014, Тернопіль, Україна/М-во освіти і науки України, Терноп. нац. екон. ун-т, Ф-т комп'ютер. інформ. технологій, Асоц. фахівців комп'ютер. інформ. технологій, -Тернопіль., 2014 .-С. 177-178 .- Бібліогр.: 2 назв. - ISBN: 978-966-654-343-4
 102. Norman K. L. Levels of Automation and User Participation in Usability Testing [Electronic resource] / Norman, K. L. and Panizzi, E. // Interacting with Computers, Vol. 18, No. 2, – 2006, –P.246-264. – Access mode: <http://dl.acm.org/citation.cfm?id=1221014>
 103. Chen M. What Can a Mouse Cursor Tell Us More? Correlation of Eye/mouse Movements on Web Browsing [Electronic resource] / Chen M., Anderson J. R., Sohn M. // CHI '01 Extended Abstracts on Human Factors in Computing Systems. – New York, USA: ACM Press, 2001. – P.281-282.: ISBN: 1-58113-340-5. – Access mode: <http://dl.acm.org/citation.cfm?id=634234>
 104. Mueller F. Cheese: Tracking Mouse Movement Activity on Websites, a Tool for User Modeling [Electronic resource] / Mueller F., Lockerd A. // Ext. Abstracts CHI, – 2001. – Access mode: <http://floydmueller.com/portfolio/cheese.pdf>.
 105. Huang, J. No clicks no problems: Using cursor movements to understand and improve search.: [Electronic resource] / Huang, J., White, R. and Dumais, S.T. // CHI '11 Proceedings of the SIGCHI Conference on Human Factors in

- Computing Systems . – New York, NY, USA: ACM, 2011. – P. 1225-1234.: ISBN: 978-1-4503-0228-9. – Access mode: <http://dl.acm.org/citation.cfm?id=1979125>.
106. LayoutEditor MEMS/IC CAD Tool (Germany) [Електронний ресурс] — Режим доступу: <http://www.layouteditor.net>. – Назва з екрану.
 107. Oleksandr Markelov, Collecting Data of User's Activities in Interface Layout of MEMS CAD Software / Oleksandr Markelov, Mykhaylo Lobur, Sofia Bobalo // Perspective Technologies and Methods in MEMS Design : Materials of the VI-th International Conference MEMSTECH'2010, April 20-23, 2010, —Lviv-Polyana: Publishing House Vezha&Co, 2010, -284 p.,—PP.44-45;
 108. Маркелов О.Е., Моделювання досвіду користувача програмного забезпечення САПР накопиченням діалогової інформаційної діяльності / О.Е. Маркелов, М.В. Лобур. // Вісн. Нац. ун-ту "Львів. політехніка" - №685: Комп'ютерні системи проектування. Теорія і практика. - Львів: Видавництво НУ"Львівська політехніка". - 2010. - С. 181-186.
 109. Blade API Monitor – Home, [Електронний ресурс], URL: <http://www.bladeapimonitor.com/>, Дата звернення: 16.03.2014.
 110. API Monitor: Spy on API Calls and COM Interfaces, [Електронний ресурс], <http://www.rohitab.com/apimonitor/>, Дата звернення: 16.03.2014.
 111. WinAPIOverride: Free Advanced API Monitor, spy or override API or exe internal function, [Електронний ресурс], <http://jacquelin.potier.free.fr/winapioverride32/>, Дата звернення: 16.03.2014.
 112. Overview – Process Hacker, [Електронний ресурс], <http://processhacker.sourceforge.net/>, Дата звернення: 16.03.2014.
 113. API Logger, [Електронний ресурс], http://blackninja2000.narod.ru/rus/api_logger.html, Дата звернення: 16.03.2014.
 114. Home Page: Spy++, [Електронний ресурс], <http://msdn.microsoft.com/en-us/library/aa264396%28v=vs.60%29.aspx>, Дата звернення: 16.03.2014.

115. Маркелов О.Е. Дослідження програмних інструментів моніторингу подій ПЗ для виявлення продуктивності взаємодій з користувачами / Маркелов О.Е., Ковалик Ю. М. //Сучасні комп'ютерні інформаційні технології:матеріали IV Всеукр. школи-семінару молодих вчен. і студ. (АСІТ' 2014), 16-17 трав. 2014, Тернопіль, Україна/М-во освіти і науки України, Терноп. нац. екон. ун-т, Ф-т комп'ютер. інформ. технологій, Асоц. фахівців комп'ютер. інформ. технологій, -Тернопіль., 2014 .-С. 129-130 .- Бібліогр.: 6 назв. - ISBN: 978-966-654-343-4
116. Фисенко В.Т. Компьютерная обработка и распознавание изображений: учеб. пособие. / Фисенко В.Т. , Фисенко Т.Ю.- СПб: СПбГУ ИТМО, 2008. – 192 с.
117. Местецкий Л. М. Математические методы распознавания образов [Электронный ресурс] / Л. М. Местецкий / МГУ, ВМиК, кафедра «Математические методы прогнозирования». – 2002-2004. – Режим доступа: <http://www.ccas.ru/frc/papers/mestetskii04course.pdf> – Назва з екрану.- Дата звернення: 01.12.2013.
118. Pavel Torgashov, Contour Analysis for Image Recognition in C# [Электронный ресурс].- Режим доступа: <http://www.codeproject.com/Articles/196168/Contour-Analysis-for-Image-Recognition-in-C> .- Назва з екрану .- Дата звернення: 01.12.2013.
119. Маркелов О.Е. Процес перетворення графічних прототипів віконних форм інтерфейсу користувачів ПЗ у програмний код / Маркелов О.Е., Казарян А.Г. // Сучасні комп'ютерні інформаційні технології:матеріали IV Всеукр. школи-семінару молодих вчен. і студ. (АСІТ' 2014), 16-17 трав. 2014, Тернопіль, Україна/М-во освіти і науки України, Терноп. нац. екон. ун-т, Ф-т комп'ютер. інформ. технологій, Асоц. фахівців комп'ютер. інформ. технологій, -Тернопіль., 2014 .-С. 127-128 .- Бібліогр.: 3 назв. - ISBN: 978-966-654-343-4

120. Markelov Oleksandr, Computer Aided Programming of Segment User Interface Prototype from Graphic Sketch / Markelov Oleksandr, Kazarian Artem // Computer science and information technologies : proceedings of the IXth International scientific and technical conference CSIT 2014, 18-22 November 2014, Lviv, Ukraine, Lviv Polytechnic National University. — 167 p. — Lviv. ISBN : 9786176076698
121. Open Source Computer Vision Library (OpenCV) [Electronic resource] URL: <http://opencv.org/>
122. Tesseract OCR engine [Electronic resource] URL: <https://code.google.com/p/tesseract-ocr>
123. Bootstrap Framework [Electronic resource] URL: <http://getbootstrap.com/>
124. Маркелов О.Е. Структурні рішення проектування системи автоматизованого реінжинірингу інтерфейсу користувача / Маркелов О.Е., Пастух С.В. // Сучасні комп'ютерні інформаційні технології: Матеріали Всеукраїнської школи-семінару молодих вчених і студентів АСІТ'2011. - Тернопіль : Економічна думка. 2011. с.113-114, ISBN 987-966-654-272-7
125. Oleksandr Markelov. Repository of Reengineering Source Code Patterns in UI Designing: the General Information Structure / Oleksandr Markelov, Sergii Pastukh, Mykhaylo Lobur, Sofia Bobalo // Proc. of XII-th International Conference on the Experience of Designing and Application of CAD Systems in Microelectronics (CADSM'2013), February, 20-24, Lviv-Polyana, Ukraine, 2013 — 480 P., PP. 455-455. - 4 refer. ISBN: 978-617-607-393-2
126. Oleksandr Markelov, Repository of User Interface Controls in area of Electronic Design Automation Software for Usability Improvement / Oleksandr Markelov, Mykhaylo Lobur. // Proc. of the XI Intern. Conf. on The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM'2011), February 23-25.. — Lviv - Polyana: Publishing House Vezha&Co. 2011. - Pp. 314-315. - ISBN: 978-966-2191-16-5, IEEE Catalog Number: CFP11508-PRT; INSPEC Accession Number:11930482

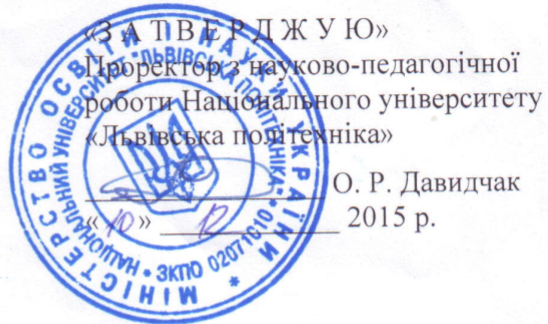
127. Oleksandr Markelov. The Review of CAD Software in Machinery Design: Spectral Characteristics of GUI components / Oleksandr Markelov, Mykhaylo Lobur // CAD in Machinery Design. Implementation and Education Problems: Proceeding of the XX Ukrainian-Polish Conference CADMD'2012 (Oct 11-13, 2012). - Lviv, Ukraine: Publishing House Vezha&Co, 2012. -136 c., - PP. 96-98, ISBN: 978-617-607-332-1
128. AutoCAD [Electronic resource] , Mode of access: <http://www.autodesk.com/products/autocad/overview/>. – Last access: 2012-Sep-20. – Title from the screen.
129. BRL-CAD [Electronic resource] , Mode of access: <http://brlcad.org/>. – Last access: 2012-Sep-20. – Title from the screen.
130. BtoCAD [Electronic resource] , Mode of access: http://www.yfcad.com/product_feature.php?pname=BtoCAD. – Last access: 2012-Sep-20. – Title from the screen.
131. FreeCAD [Electronic resource] , Mode of access: <http://freecad.sourceforge.net/>. – Last access: 2012-Sep-20. – Title from the screen.
132. IronCAD [Electronic resource] , Mode of access: <http://www.ironcad.com/>. – Last access: 2012-Sep-20. – Title from the screen.
133. Компас [Electronic resource] , Mode of access: http://ascon.net/solutions/kompas_3d/. – Last access: 2012-Sep-20. – Title from the screen.
134. MeshLab [Electronic resource] , Mode of access: <http://meshlab.sourceforge.net/>. – Last access: 2012-Sep-20. – Title from the screen.
135. NanoCAD [Electronic resource] , Mode of access: <http://www.nanocad.ru/>. – Last access: 2012-Sep-20. – Title from the screen.
136. Open CASCADE [Electronic resource] , Mode of access: <http://www.opencascade.org/>. – Last access: 2012-Sep-20. – Title from the screen.

137. progeCAD [Electronic resource] , Mode of access: <http://www.progesoft.com/ru/downloads>. – Last access: 2012-Sep-20. – Title from the screen.
138. QCAD [Electronic resource] , Mode of access: <http://www.qcad.org>. – Last access: 2012-Sep-20. – Title from the screen.
139. SALOME [Electronic resource] , Mode of access: <http://www.salome-platform.org/>. – Last access: 2012-Sep-20. – Title from the screen.
140. SolidWorks [Electronic resource] , Mode of access: <http://www.solidworks.com>. – Last access: 2012-Sep-20. – Title from the screen.
141. T-FLEX CAD [Electronic resource] , Mode of access: <http://www.tflex.ru>. – Last access: 2012-Sep-20. – Title from the screen.
142. TurboCAD [Electronic resource] , Mode of access: <http://www.turbocad.ru/>. – Last access: 2012-Sep-20. – Title from the screen.
143. Markelov O. Database of visual and source code components of user interface: information model design / Markelov O. // Вісн. Нац. ун-ту “Львів. політехніка” - № 747: Комп’ютер. системи проектув. Теорія і практика. - Львів: Видавництво НУ“Львівська політехніка”. - 2012. - С. 109-114. - Бібліогр.: 22 назв.;
144. Markelov Oleksandr. Tracking User Interface Events: Information Structure of Tool / Oleksandr Markelov, Mykhaylo Lobur, Sofiya Bobalo // V-th International Scientific and Technical Conference “Computer Sciences and Information Technologies”(CSIT'2010, Lviv, UKRAINE, October 14-16), 2010. - Lviv, PP. 121-122
145. Oleksandr Markelov, Information integration of the repository of software UI components with the data structure of user interaction / Oleksandr Markelov // Вісн. Нац. ун-ту “Львів. політехніка” - №777: Комп’ютер. системи проектув. Теорія і практика:[зб. наук. пр.]/відп. ред. М.В. Лобур. - Л., - 2013. 5 ст. - Бібліогр.: 4 назв.

146. Comparison of EDA software, Wikipedia, the free encyclopedia, Mode of access: http://en.wikipedia.org/wiki/Comparison_of_Free_EDA_software. – Last access: 2010-Dec-27.
147. Electric VLSI Design System / Steven Rubin // Static Free Software, California. Mode of access: <http://www.staticfreesoft.com/>. – Last access: 2010-Dec-27. – Title from the screen.
148. KiCad Free Open-Source EDA Suite. Mode of access: <http://kicad.sourceforge.net/>. – Last access: 2010-Dec-27. – Title from the screen.
149. FreePCB. Mode of access: <http://www.freepcb.com/>. – Last access: 2010-Dec-27. – Title from the screen.
150. Ганеев Р.М., «Проектирование интерфейса пользователя средствами WIN32 API», – К.:Телеком, 2001. - 336с.
151. Дэвид Хантер, XML Базовый курс = Beginning XML / Дэвид Хантер, Джефф Рафтер и др. — М. : Вильямс, 2009. — 1344 с. — ISBN 978-5-8459-1533-7
152. Сергеев А. П. HTML и XML. Профессиональная работа / Сергеев А. П. , — М. : Диалектика, 2004. — 880 с. — ISBN 5-8459-0676-8
153. Валиков А. Технология XSLT/ Валиков А. // БХВ-Петербург, 2002, ISBN 5-94157-129-1
154. Бибо Б., jQuery. Подробное руководство по продвинутому JavaScript / Бибо Б., Кац И. - Москва : Символ-Плюс, 2008, – 384 с., ISBN 5-93286-135-5
155. Markelov Oleksandr. Information System for Parametrized Comparison of Set of Tools for User Interfaces Prototyping / Oleksandr Markelov // VIII-th International Scientific and Technical Conference “Computer Sciences and Information Technologies”(CSIT'2013, Lviv, UKRAINE, November 11-16), 2013 — 200 P., — Lviv, PP. 132, - 6 refer. ISBN: 978-617-607-520-2

156. Download.com [Electronic resource] : The place where people go to discover free-to-try legal downloads. – Access mode: <http://www.download.com/> – Last access: 2013-Oct-23. – Title from the screen.
157. Apple Online Store [Electronic resource] . – Access mode: <http://store.apple.com>. – Last access: 2013-Oct-23. – Title from the screen.
158. Amazon Service [Electronic resource]. – Access mode: <http://www.amazon.com>. – Last access: 2013-Oct-23. – Title from the screen.
159. Google Play Store [Electronic resource] . – Access mode: <https://play.google.com/store>. – Last access: 2013-Oct-23. – Title from the screen.
160. Oleksandr Markelov. Development of computer-aided tools for UI synthesis based on monitoring of usage and the source code repository / Oleksandr Markelov // International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), Volume 5 Issue 12, December-2016, pp.: 2733-2737. ISSN: 2278-1323 (Online), Shri Pannalal Research Institute of Technology, India. [Electronic resource]. – Access mode: http://ijarcet.org/?page_id=4835

Додаток А. Акт впровадження в навчальний процес



А К Т

впровадження у навчальний процес результатів дисертаційної роботи Маркелова Олександра Едуардовича «Інструментальні засоби автоматизації синтезу інтерфейсів користувача на базі репозиторію інтелектуальних патернів» на здобуття наукового ступеня кандидата технічних наук

Дисертаційна робота аспіранта Маркелова Олександра Едуардовича «Інструментальні засоби автоматизації синтезу інтерфейсів користувача на базі репозиторію інтелектуальних патернів» подана на здобуття наукового ступеня кандидата технічних наук за спеціальністю 05.13.12– «Системи автоматизації проектувальних робіт». Дисертаційна робота спрямована на розв'язання актуальної науково-технічної проблеми автоматизації процесів проектування інтерфейсів користувачів програмного забезпечення з врахуванням накопичених даних тестування, попередньої експлуатації, досвіду використання, метрик та патернів компонування та програмування компонентів інтерфейсів користувача.

Результати досліджень: ієрархічна систематизація користувацьких взаємодій, ієрархічна систематизація кінетично-орієнтованих інтерфейсів користувачів, метод критеріального аналізу мов декларування користувацьких інтерфейсів, систематизація візуальних конструктивів інтерфейсів користувача програмного забезпечення, результати спектрального аналізу компонентів інтерфейсу користувача ПЗ, інформаційна модель бази даних візуальних і програмних компонентів інтерфейсу користувача, аналітичний метод для обрання засобу прототипування інтерфейсів користувача, інформаційна архітектура накопичення діалогової інформаційної діяльності користувачів ПЗ, інформаційна архітектура репозиторію патернів програмного перепроєктування інтерфейсів користувача, інформаційна модель для формування репозиторію конструктивних елементів користувацьких інтерфейсів. Наведені вище результати дисертаційних досліджень використовуються у процесі викладання дисципліни «Методи побудови інтелектуального інтерфейсу користувача в автоматизованому проектуванні» для студентів спеціальності 7.05010103 / 8.05010103 «Системне проектування» (спеціалісти та магістри).

Акт затверджено на засіданні кафедри САП від 27.08.2015 р., протокол № 1.

Директор Інституту комп'ютерних наук та інформаційних технологій
д. т. н., професор

Медиковський М. О.

Зав. кафедри систем автоматизованого проектування д. т. н., професор

Лобур М. В.

Додаток Б. Акт впровадження (використання) в діяльності ТзОВ

«ЗАТВЕРДЖУЮ»
 Директор ТзОВ «Едвантіс»
 _____ О. В. Глазунов
 «15» _____ 2015 р.

А К Т

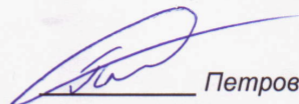
Впровадження (використання) результатів дисертаційної роботи
 Маркелова Олександра Едуардовича
 «Інструментальні засоби автоматизації синтезу інтерфейсів користувача на
 базі репозиторію інтелектуальних патернів»
 на здобуття наукового ступеня кандидата технічних наук

Дисертаційна робота Маркелова Олександра Едуардовича «Інструментальні засоби автоматизації синтезу інтерфейсів користувача на базі репозиторію інтелектуальних патернів» на здобуття наукового ступеня кандидата технічних наук за спеціальністю 05.13.12 – «Системи автоматизації проектувальних робіт» є спрямованою на дослідження та систематизацію підходів, концепцій і компонентів процесів автоматизації проектування інтерфейсів користувачів програмного забезпечення із застосуванням даних метричних тестувань та досвіду експлуатації інтерфейсів користувача.

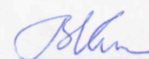
Такі результати досліджень, як ієрархічна систематизація користувацьких взаємодій та візуальних компонентів інтерфейсів користувача, метод критеріального аналізу мов декларування користувацьких інтерфейсів, результати спектрального аналізу компонентів інтерфейсу користувача, аналітичний метод для обрання засобу прототипування інтерфейсів користувача, інформаційна архітектура репозиторію патернів програмного проектування інтерфейсів користувача, мають значну наукову та практичну цінність і здатні підвищити якісні характеристики проєктованих користувацьких інтерфейсів.

Наведені вище результати дисертаційних досліджень Маркелова О. Е. використовуються в ТзОВ «Едвантіс» на етапі створення прототипів програмних продуктів, а також під час проектування графічних візуальних інтерфейсів нових програмних засобів. Використання вказаних підходів дозволило скоротити час і витрати на створення сучасних користувацьких інтерфейсів, а також покращити їх ергономічні показники для кінцевого користувача.

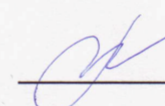
Директор технічний, к.т.н.

 Петров Д. В.

Начальник відділу проектування ПЗ

 Климко В. Р.

Підписи завіряю
 Інспектор з кадрів

 Шевлягіна Т. П.