

## СТРУКТУРНІ РІШЕННЯ ПРОЕКТУВАННЯ СИСТЕМИ АВТОМАТИЗОВАНОГО РЕІНЖІНІРИНГУ ІНТЕРФЕЙСУ КОРИСТУВАЧА

Маркелов О.Е., Пастух С.В.

Національний університет «Львівська політехніка»

### I. Постановка проблеми

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 чи їхньою 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 реінжиніринг (перепроєктування) вже існуючого, для розширення сфер використання кінцевими користувачами. Інтерфейс користувача можна створювати різними спеціалізованими мовами програмування [1]. Але зараз 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 такому випадку необхідне автоматичне виявлення усіх інтерактивних взаємодій з елементами інтерфейсу користувача з подальшим автоматизованим перепроєктуванням у нову програмістську технологію чи бібліотеки (frameworks). Об'ємні дослідження користувацького інтерфейсу та дизайну привели до більш глибокого розуміння принципів проектування [1-6]. Інтеграція нових змін у інтерфейсні технології можуть значно підвищити інноваційність існуючих підходів до реінжинірингу систем [7].

### II. Мета та завдання роботи

Метою даного дослідження є пошук і пропозиція методи модернізації існуючих інтерфейсів та структурних рішень для розробки інформаційної архітектури системи чи підсистеми автоматичного чи напівавтоматичного реінжинірингу користувацьких інтерфейсів, таким чином щоб продовжити термін життя програмне забезпечення. Проектування нового інтерфейсу для користувача або перепроєктування вже існуючого - сьогодні, мабуть, найнеобхідна на ринку розробки ПЗ послуга. У роботі розробляється семантичний аналізатор на основі XSLT та база даних еквівалентного співставлення локальних та веб-рішень графічних елементів користувацького інтерфейсу.

### III. Структурні підходів до перепроєктування інтерфейсу користувача

На рис.1 представляється схема роботи системи автоматичного перепроєктування інтерфейсу користувача:

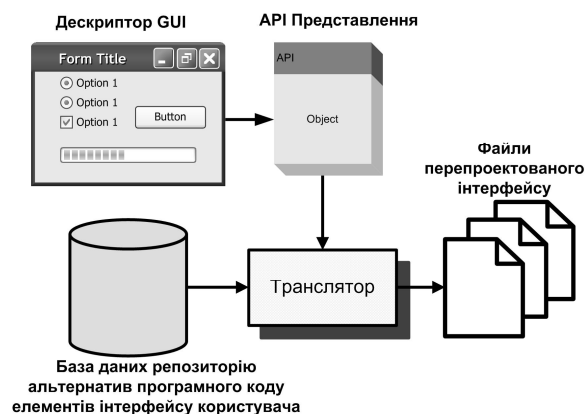


Рисунок 1 - Загальна схема роботи системи

Попередньо система читає усі запущені процеси операційної системи через дескриптори форм GUI, але працює лише з тим, який вказує користувач. За допомогою функцій WinAPI досліджується графічне представлення процесу в ОС: його меню, кнопки, допоміжні вікна, піктограми тощо. Як показано на рис. 1, представлення такого виду, у вигляді функцій, переходить у блок «Транслятор» для подальшого опрацювання. Як бачимо, система використовує базу даних, про вміст якої ми поговоримо дещо пізніше, до якої звертається транслятор при генерації нового інтерфейсу.

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

На рис. 2 детальніше показано схему роботу транслятора:

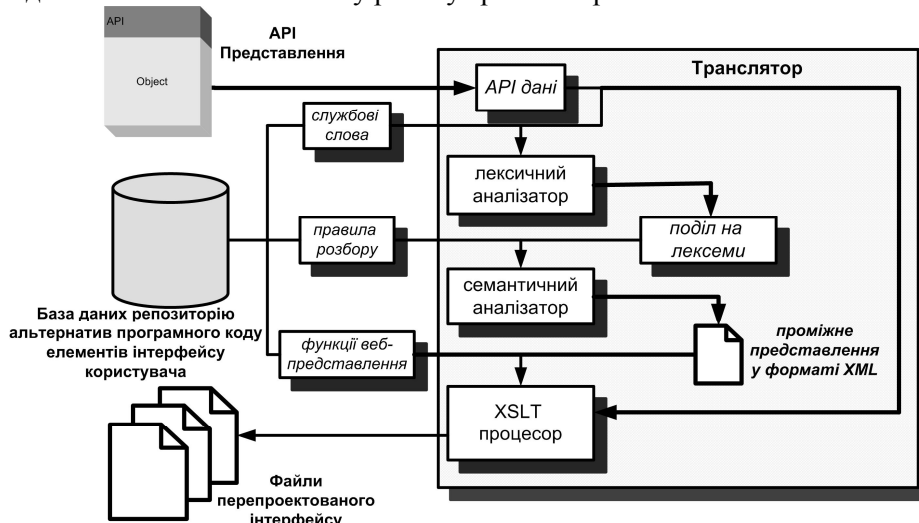


Рисунок 2 - Схема роботи транслятора системи

На вхід транслятора поступає програмний код у вигляді залежності функцій, з яких можна виділити такі характеристики як: тип графічного компонента, назва, висота, ширина, піктограма (фонова картинка), активність, тощо. Уся ця інформація поступає у лексичний блок транслятора. Аналізатор, в свою чергу, розбиває вхідний код на лексичні складові – лексеми (службові слова, ідентифікатори, розділювачі). Варто зрозуміти, що на цьому етапі роботи системи ми оперуємо 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 виступає 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 на проміжній мові, якою обрано XML, для представлення вхідного інтерфейсу на мові опису абстракцій (МОА). XML дозволяє віокремити дані для повноцінного представлення ієрархії взаємодій компонентів інтерфейсу.

Останнім кроком роботи транслятора є генерація коду, яка в системі реалізується за допомогою XSLT процесора. З бази даних співставляються функції веб-рішень локального представлення компонентів інтерфейсу, які поступають на вхід процесора разом із раніше відокремленими даними, та граматику переводу XML на інші мови, такі як JavaScript, HTML,

### Список використаних джерел

1. Лобур М., Маркелов О., Бобало С. Аналіз мов опису користувацьких інтерфейсів для застосування у програмному забезпеченні САПР / Вісник Національного університету "Львівська політехніка" Комп'ютерні науки та інформаційні технології № 663.- Львів: Видавництво НУ"Львівська політехніка". 2010 - С. 30-38
2. Маркелов О.Е., Лобур М.В. Моделювання досвіду користувача програмного забезпечення САПР накопиченням діалогової інформаційної діяльності. Вісник Національного університету "Львівська політехніка": Комп'ютерні системи проектування. Теорія і практика № 685. - Львів: Видавництво НУ"Львівська політехніка". - 2010. - С. 181-186.
3. Oleksandr Markelov, Mykhaylo Lobur, Sofia Bobalo. Collecting Data of User's Activities in Interface Layout of MEMS CAD Software, 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.- Pp. 44-45.
4. Oleksandr Markelov, Mykhaylo Lobur. Kinetic User Interface for Human-Computer Interactions: Systematized Review. Proceeding of the XVIII Ukrainian - Polish Conference on "CAD in Machinery Implementation and Educational Problems" 14 - 16 October 2010.- Lviv, Ukraine: Publishing House Vezha&Co. 2010.- Pp. 69-73.
5. Oleksandr Markelov, Mykhaylo Lobur, Sofiya Bobalo. Tracking User Interface Events: Information Structure of Tool. Proceeding of the V International Scientific and Technical Conference "Computer Science & Information Technologies" 14 - 16 October 2010.- Lviv, Ukraine. 2010.- Pp. 121-122
6. Маркелов О. Систематизований огляд користувацьких інтерфейсних взаємодій з програмним забезпеченням / Маркелов Олександр // Комп'ютерні науки та інженерія : матеріали VI Міжнародної конференції молодих вчених CSE-2010, 25-27 листопада, 2010, Україна, - Львів, С.26-27
7. Ettore Merlo, Pierre-Yves Gagné, Jean-Francois Girard, Kostas Kontogiannis, Laurie Hendren, Prakash Panangaden, Renato De Mori, "Reengineering User Interfaces," IEEE Software, vol. 12, no. 1, pp. 64-73, Jan. 1995