

Теорема 3. Нехай задані системи обробки даних $S_1 \subset (X_1^* \times Z_{X_1}) \times Y_1$, $S_2 \subset (X_2^* \times Z_{X_2}) \times Y_2$ і визначено паралельне з'єднання (синтезу) цих систем з допомогою паралельного з'єднання операції

$$S_1 + S_2 = S = (X_1^* \times X_2^* \times Z) \times (Y_1 \times Y_2).$$

Тоді система S допускає розпаралелювання обробки даних на рівні S_1 і S_2 .

Доведення. Якщо за умови побудови синтезу S

$$X_1^* \cap Y_2 = \wedge X_2^*, \quad Z_{X_2} \cap Y_1 =$$

Тобто системи S_1 і S_2 згідно з означенням є інформаційно незалежні, тоді система S допускає розпаралелювання обробки даних на рівні S_1 і S_2 . На рис. 3 представлена система S обробки даних.

Висновок

Описано методи та алгоритми складних систем паралельної обробки даних, що можуть бути ефективно застосовані у задачах комп'ютерного зору. Показано також можливості синтезу паралельних реалізацій.

1. Форсайт Д., Понс Ж. Компьютерное зрение. Современный подход. – М., СПб., К.: Изд. дом "Вильямс", 2004. – 908 с. 2. Джордж Ф. Люггер. Искусственный интеллект. Стратегии и методы решения сложных проблем. – М., СПб., К.: Изд. дом "Вильямс", 2002. – 848 с. 3. Грицик В.В., Грицик В.В. Розпаралелювання обробки даних для реалізації інформаційно-аналітичних систем. – Львів: ДНДІІ, 2009. – 120 с.

УДК 004.414.3:004.432.45:681.5

М. Лобур, О. Маркелов, С. Бобало

Національний університет "Львівська політехніка",
кафедра систем автоматизованого проектування

АНАЛІЗ МОВ ОПИСУ КОРИСТУВАЦЬКИХ ІНТЕРФЕЙСІВ ДЛЯ ЗАСТОСУВАННЯ У ПРОГРАМНОМУ ЗАБЕЗПЕЧЕННІ САПР

© Лобур М., Маркелов О., Бобало С., 2010

Аналітичний огляд перспективних концепцій декларування користувацьких інтерфейсів для розроблення програмного забезпечення (ПЗ) систем автоматизованого проектування (САПР) та їхній ваговий критеріальний експертний аналіз.

Perspective concepts declaration of user interfaces for software development computer-aided design (CAD): review and objective appraisal.

Постановка проблеми та актуальність

Користувацький інтерфейс є однією з найоб'ємніших і найбільш трудомістких частин розроблення будь-якої системи автоматизованого проектування. Саме завдяки зручності і продуманості інтерфейсів значно підвищується ефективність та швидкість виконання проектувальником поставленої задачі. Практично усі сучасні САПР створені з використанням

програмування користувацьких інтерфейсів на базі стандарту інтерфейсних елементів операційної системи. Але ця ситуація є зручною з огляду на те, що якщо користувач навчився користуватися однією з програм, то він швидко засвоїть іншу. Водночас тривале використання ПЗ САПР спонукає досвідченого користувача-проектувальника з великим досвідом використання ПЗ САПР і досвідом у проектуванні адаптувати інтерфейси ПЗ до власних потреб з метою оптимізації 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 застосування програмних бібліотек для якогось 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 задання візуального оформлення інтерфейсів, програмних реакцій та станів для кожного з елементів користувацького інтерфейсу. Під час розроблення ПЗ великих САПР цей «несистемний» підхід означає «початок кінця». Щоб зменшити негативні наслідки такого стану справ, під час проектування ПЗ САПР варто звернутися до наявних та перспективних стандартів у галузі опису користувацьких інтерфейсів, які є об'єктами численних досліджень [1–6].

Мета дослідження

Метою дослідження 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 експертного оцінювання 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 експертного оцінювання.

Аналіз і основна частина

Під час дослідження слід зосередити увагу на визначенні списку критеріїв для експертного оцінювання мов декларування. Кожний з критеріїв матиме назву, яка дублюється скороченим поняттям та літерою англійської мови. Критерії групують за змістом у категорії. Для категорій та критеріїв вказують вагові коефіцієнти важливості та розподілу впливу на прийняття рішення. Підбираючи список спеціалізованих мов опису інтерфейсу користувача, є сенс 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 експертного оцінювання:

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 (eEffects, F)* – здатність до розширення базового відображення залежно від визначених станів.

3. *Візуальна деталізація (Visual detailing, V)* – рівень графічної повноти та відповідності психологічній асоціативності при спостереженні користувачем.

4. *Векторна графіка (vector Graphics, G)* – підтримка векторного формування графічного відображення елементів користувацьких інтерфейсів.

5. *Самотестованість (self-Test, T)* – визначається програмною методологією написання коду з можливістю автоматичного тестування поведінки ПЗ користувацьких інтерфейсів.

6. *Автоналаштуваність (aUtotune, U)* – внутрішня алгоритмічно закладена модель визначення параметрів відображення компонентів користувацького інтерфейсу залежно від характеристик роздільної здатності, вирівнювань на формі, «притягування» до згруповано-взаємодіючих елементів користувацького інтерфейсу.

7. *Ліцензія (Licence, L)* – накладає умови щодо застосування, використання та розповсюдження технології опису користувацьких інтерфейсів.

8. *Доступність платформи (availability, B)* – визначається рівнем розвитку і часом виходу технології в світ.

9. *Серверна частина (Server, S)* – вказує на необхідність окремого стороннього ядра процесора обробки.

10. *Об'єктно-орієнтована мова програмування (OOP, 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 (eNvironment, N)* – визначає спеціалізовану прив'язку до інтегрованого середовища програмування та компіляції (за необхідності). Фактично визначає свободу 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 керівними діями користувача 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.

За оптимального підходу до розв'язання задач вибору інструментарію для проектування ПЗ САПР на рівні описів користувацьких інтерфейсів є сенс враховувати такі критерії:

- 1) якість користувацьких інтерфейсів проекту ПЗ САПР (Quality, Q) → **max**
- 2) витрати на реалізацію користувацьких інтерфейсів ПЗ САПР (exPenses, P) → **min**
- 3) час на реалізацію/модифікацію інтерфейсів ПЗ САПР (pRoductive time, R) → **min**
- 4) складність реалізації /модифікації інтерфейсів ПЗ САПР (complexitY, Y) → **min**

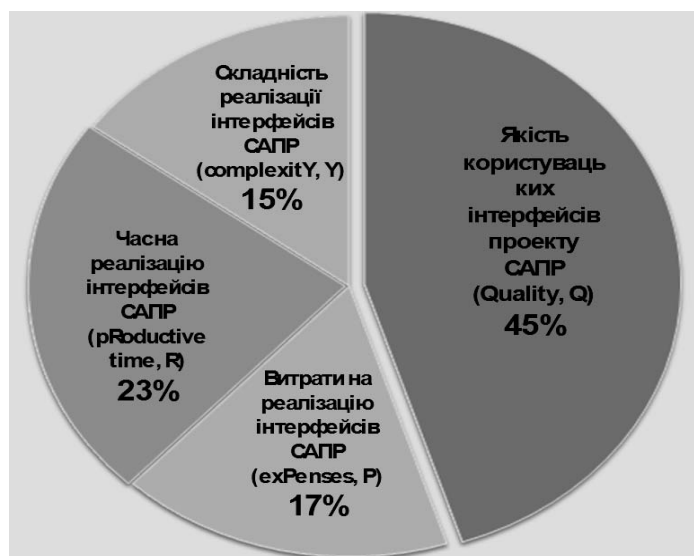


Рис. 1. Вагові бали загальної оцінки у часткових категоріях

Таблиця 1

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

Показник, Умовне позначення		Категорії				Ваговий коефі- цієнт 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 у категорії:		100 %	100 %	100 %	100 %	1

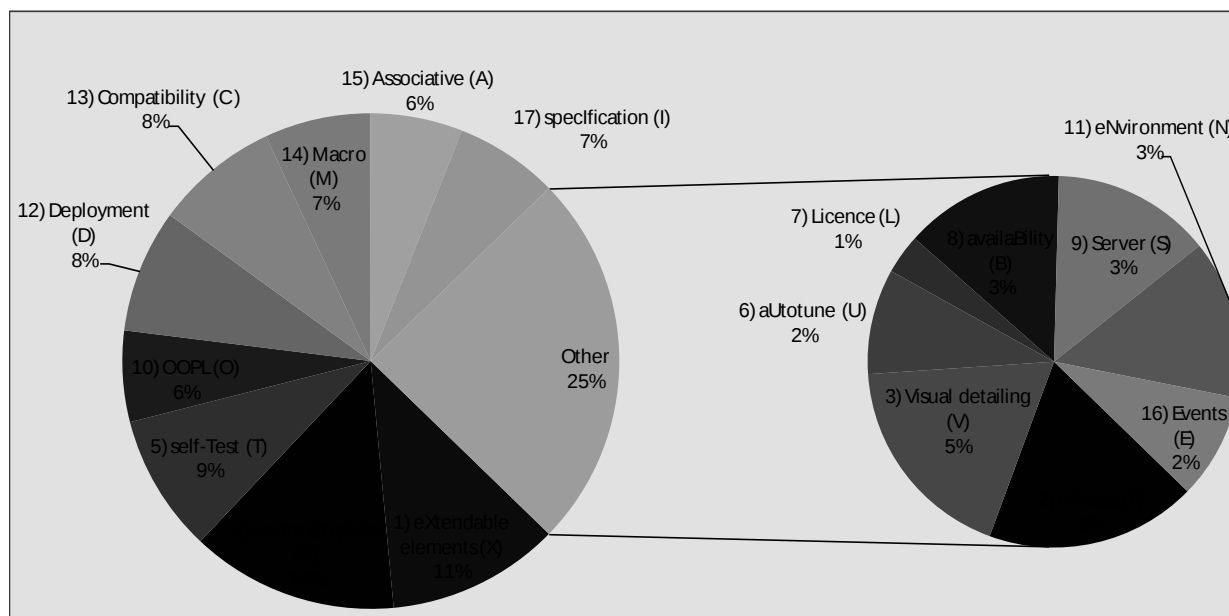


Рис. 2. Діаграми співвідношення показників із врахуванням ваг категорій оцінки мов декларування користувачьких інтерфейсів

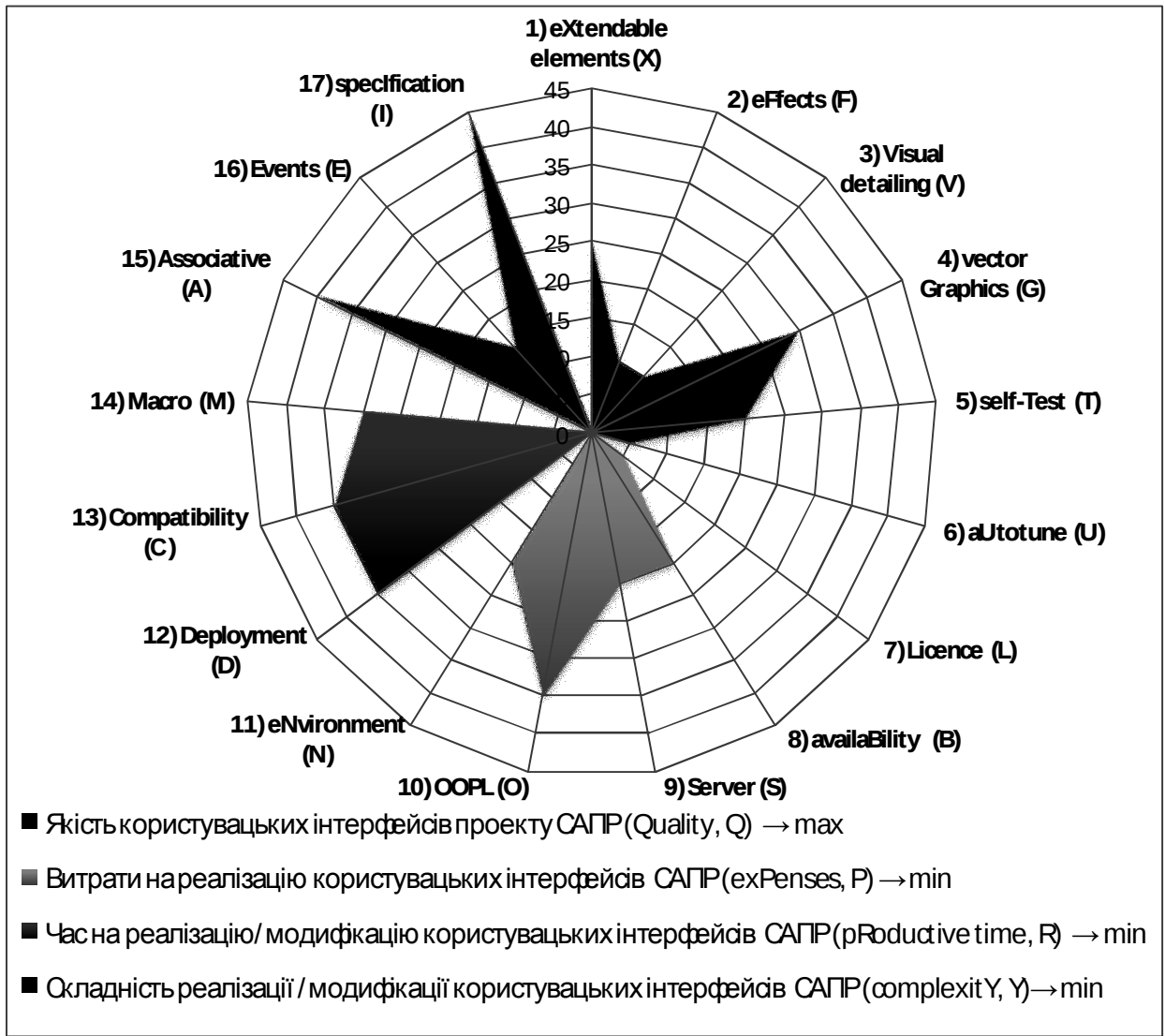


Рис. 3. Радіальна діаграма співвідношень показників у категоріях оцінювання мов користувацьких інтерфейсів

Для визначення підсумкових показників варто скористатись середньгеометричним зваженим:

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

Тоді для категорій з параметрами на основі даних табл.1:

$$i = 1 \text{ K } 17; n = 17;$$

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

Тоді значення у категорії (Quality, Q), з врахуванням процентних співвідношень:

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

Відповідно у категорії (exPenses, P) буде:

$$\bar{P} = \sqrt[20]{L + B^4 + S^4 + O^7 + N^4}; \quad (3)$$

у категорії (productive time, R) буде:

$$\bar{R} = \sqrt[20]{D^7 + C^7 + M^6}; \quad (4)$$

у категорії (complexity, Y):

$$\bar{Y} = \sqrt[20]{A^8 + E^3 + I^9}; \quad (5)$$

Отже, середньоарифметичним зваженим для комплексного показника готовності мови програмування опису користувацького інтерфейсу буде:

$$\bar{x} = \frac{\sum_{i=1}^n w_i \cdot x_i}{\sum_{i=1}^n w_i}, \quad (6)$$

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

$$\bar{Z} = \bar{Q} \cdot 0.45 + \bar{P} \cdot 0.17 + \bar{R} \cdot 0.23 + \bar{Y} \cdot 0.15. \quad (7)$$

Огляд мов опису і програмного декларування користувацьких інтерфейсів

Серед найпоширеніших мов декларування інтерфейсу користувача слід зазначити такі:

1) *UIML (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 (Model View Controller) підходу багаторівневої розробки ПЗ. Визначає: складені елементи користувацького інтерфейсу, способів відображення, тип вмісту, реакції на дії користувача, тип контролю за діями, зв'язок із зовнішніми API [1, 7–10].

2) *XUL (XML User Interface Language)* – мова опису розмітки для динамічного створення користувацьких інтерфейсів на основі описів мовою XML (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 [1, 11–14].

3) *XAML (eXtensible Application Markup Language)* – розширювана мова опису розмітки декларативного програмування, розроблена Microsoft. Визначає набір властивостей, методів та подій, які дають змогу об'єднати веб-документи у прикладну програму. Програмні об'єкти контролюють виконання програми і генерують події для коду користувацького інтерфейсу. У XAML-документ також включені програмна логіка та стилі. Функціонально його можна розглядати як комбінацію XUL, SVG (Scalable Vector Graphics), CSS (Cascading Style Sheets) та JavaScript (JS) в одній схемі XML [1, 15–17].

4) *MXML (Macromedia eXtensible Markup Language)* – мова опису розмітки користувацького інтерфейсу, оснований на XML. Використовується переважно для декларативного опису інтерфейсу програми, а також може бути використана для реалізації бізнес-логіки поведінки і інтернет-додатків. Мова дає змогу наочно описати структуру користувацького інтерфейсу. Розробляється специфікація фірмою Adobe у межах технологій Flex та AIR [1, 18–21].

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

1) *AUIML (Abstract User Interface Markup Language)* є діалектом 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 дає змогу розробникам створювати додатки і запускати їх в Swing або на WWW без будь-яких змін. Проект ініційований компанією IBM, але зараз розвиток мови припинений [22].

**Порівняльна таблиця мов розмітки користувацьких інтерфейсів
у застосуванні розробки САІР**

Критерій \ Мова	UIML	XUL	XAML	MXML (Flex)
Якість користувацьких інтерфейсів проекту САІР (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 САІР (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. XUL Explorer	MSExpression, MSVisualStudio, XAMLPad	Adobe Flex Builder
Час на реалізацію/модифікацію користувацьких інтерфейсів САІР (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 САІР (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

2) *XAL (eXtensible Application Language)* – технологія, створена Nexaweb. Для розроблення 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) [23].

3) *LZX (OpenLaszlo XML)* – декларативна мова опису користувацького інтерфейсу, яка визначає елементи інтерфейсу, компоновку програмного додатку та скриптові виклики подій. Є вільною відкритою технологією. OpenLaszlo підтримує об'єктно-орієнтований підхід та прив'язку даних до елементів інтерфейсу відображення, дає змогу створювати програмні додатки в одному файлі зі всіма вкладеними даними [24].

4) *HMVCUL (Hierarchical Model View Controller User Interface Language)* – мова користувацького інтерфейсу, застосована на XML, яка підтримує створення та зв'язування елементарних компонент у MVC тріади. Зв'язування досягається відслідковуванням деревоподібної структури, яка описана в файлі. Крім того, оскільки користувацький інтерфейс побудований з функціональних атомарних блоків, то блоки можуть бути повністю багаторазові і взаємозамінні між додатками [25].

5) *UIDL (User Interface Description Language)* – об'єктно-орієнтована мова, заснована мовою JavaScript. Коди виконуються у спеціалізованому клієнтському середовищі з асинхронним доступом до серверних даних. Користувацький інтерфейс може бути одночасно описаний вставками мовою текстової розмітки HTML [26].

Підсумки

Загальне враження про використання тих чи інших мов опису користувацького інтерфейсу з погляду програмного впровадження у розробку ПЗ САПР є таким. Застосування, що декларуються такими провідними корпораціями, як Microsoft, Adobe, є найбільш розвинутими і за рівнями функціональності, і за рівнями якості отримуваних інтерфейсів для впровадження як локальних, так і веб-програмних продуктів. Привабливість для розробників ПЗ САПР є високою. Але слід зазначити значну вартість серверних рішень для технологій та середовищ розроблення. Крім цього, запропоновані технології є трудомісткими при програмуванні. Інструмент XUL від групи Mozilla надає 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 найперспективнішою 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.

1. Шейко Д. Язyki описания пользовательских интерфейсов. 08 марта 2005 г. с.15. – [Електронний ресурс], URL:<http://www.cmsdevelopment.com>, Дата звернення: 15.12.2009.
2. Клименко С., Уразметов В. Графические интерфейсы и средства их разработки // Индустрия программирования. – 1996, 1–6 октября.
3. 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.
4. Моделі, методи та засоби адаптивності користувацького інтерфейсу: Автореф. дис... канд. техн. наук: 05.13.06 / Д.В. Ходаков; Херсон. держ. техн. ун-т. – Херсон, 2003. – 19 с. – укр.
5. Jeffrey Nichols, Andrew Faulring, Automatic Interface Generation and Future User Interface Tools //

Language Learning & Technology Refereed Journal, volum 2, num 1, 2006. 6. 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. 7. 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, [Электронный ресурс], URL: <http://www.uiml.org>, Дата звернення: 15.12.2009. 8. 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. 9. 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. 10. 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. 11. Vaughn Bullard, Kevin T. Smith, Michael C. Daconta, Essential XUL Programming, John Wiley & Sons, 2001, 448 p. ISBN-13: 978-0471415800. 12. Kenneth C. Feldt, Programming Firefox: Building Rich Internet Applications with XUL, O'Reilly Media, 2007, - 494 p. ISBN-13: 978-0596102432. 13. Jurgen Schumacher, Markus Stäuble, ZK Developer's Guide, Packt Publishing (March 28, 2008), -187 p. ISBN-13: 978-1847192004. 14. Mozilla Developer Center. XUL Documentation. Mozilla Foundation. [Электронный ресурс], URL: <https://developer.mozilla.org/En/XUL>, Дата звернення: 15.12.2009. 15. Extensible Application Markup Language. Microsoft Developer Network. [Электронный ресурс], URL: <http://msdn.microsoft.com/en-us/library/ms747122.aspx>, Дата звернення: 15.12.2009. 16. 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. 17. Brennon Williams, Microsoft Expression Blend Unleashed, Sams Publishing; 1 edition (June 13, 2008), -600p., ISBN-13: 978-0672329319. 18. Flex Interface Guide. Flex Development Center. Adobe Developer Connection. [Электронный ресурс], - Режим доступа: <http://www.adobe.com/devnet/flex/>, Дата звернення: 15.12.2009. 19. Adobe Flex v.3.4 Language Reference. Flex Development Center. Adobe Developer Connection. [Электронный ресурс], URL: <http://livedocs.adobe.com/flex/3/langref/overview.html>, Дата звернення: 15.12.2009. 20. 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. 21. Charles E. Brown, The Essential Guide to Flex 3, friends of ED (May 26, 2008), -600 p., ISBN-13: 978-1590599501. 22. Abstract User Interface Markup Language Toolkit. Development team at the IBM Rochester lab, February 16, 2004. [Электронный ресурс], URL: <http://www.alphaworks.ibm.com/tech/auiml>, Дата звернення: 15.12.2009. 23. Nexaweb Developer Guide, Nexaweb Dev Center, February 2009 Edition, [Электронный ресурс], URL: <http://www.openxal.org/>, Дата звернення: 15.12.2009. 24. An XML Framework for Rich Internet Applications, Laszlo Systems Technology White Paper, July, 2005, [Электронный ресурс], URL: <http://www.laszlosystems.com>, Дата звернення: 15.12.2009. 25. Hierarchical Model View Controller User Interface Language (HMVCUL) [Электронный ресурс], URL: <http://www.priorartdatabase.com/IPCOM/000160727/>, Дата звернення: 15.12.2009. 26. UIDL: User Interface Description Language, Reference Documentation. [Электронный ресурс], URL: <http://www.uidl.net/>, Дата звернення: 15.12.2009.