

М. В. Ткачук¹, Р. О. Гамзаєв¹, О. Т. Товстокоренко², І. О. Мартінкус¹, К. А. Нагорний¹

¹Харківський національний університет імені В. Н. Каразіна, Харків, Україна

²НТУ «Харківський політехнічний інститут», Харків, Україна

АНАЛІЗ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ CASE - ЗАСОБІВ ДЛЯ МОДЕЛЮВАННЯ ВАРІАБЕЛЬНОСТІ В ПРОЦЕСАХ РОЗРОБКИ ЛІНІЙОК ПРОГРАМНИХ ПРОДУКТІВ

Анотація. Предметом вивчення в статті є інструментальні засоби (CASE – засоби) та метрики для моделювання та дослідження характеристик варіабельності архітектур та окремих компонентів лінійок програмних продуктів (ЛПП). **Метою** є аналітичний огляд функціональності та визначення критеріїв вибору відповідних CASE – засобів для моделювання архітектур ЛПП з урахуванням конкретних показників їх якості, які обчислюються на основі оцінки рівня їх варіабельності із застосуванням кількісних метрик, що має забезпечити можливість робити висновки щодо ефективності побудови та подальшого застосування ЛПП. **Завдання:** проаналізувати сучасні інформаційні джерела за темою дослідження, провести порівняльний аналіз декількох CASE – засобів та обрати тих, що уможливають моделювання характеристик варіабельності у різних типах ЛПП, запропонувати метрики для їх кількісного визначення та провести відповідні експериментальні дослідження. Використовуваними **методами** є: структурний аналіз особливостей побудови ЛПП; кількісні метрики визначення рівня варіабельності архітектури та окремих компонентів ЛПП. Отримані такі **результати:** проаналізовано функціональні можливості сучасних CASE – засобів для моделювання характеристик варіабельності в ЛПП, мотивовано обрано метрики для кількісного визначення рівня варіабельності та експериментально досліджено можливість їх застосування для аналізу ефективності процесів розробки та подальшого використання ЛПП. **Висновки:** проведені експериментальні дослідження для тестових моделей ЛПП показали, що за умов розширення функціональності наявних CASE – засобів шляхом застосування кількісних метрик для визначення рівня варіабельності можливо надавати практичні рекомендації щодо підвищення ефективності побудови та подальшого використання відповідних ЛПП. Напрямок **подальших досліджень** є розробка програмних рішень для вдосконалення функціональності існуючих CASE – засобів та проведення з їх використанням більшої кількості експериментальних досліджень ефективності запропонованого підходу.

Ключові слова: лінійка програмних продуктів, варіабельність, архітектура, компонент, CASE - засіб, метрика.

Вступ

Актуальність проблеми автоматизації застосування методів доменного моделювання при розробці лінійок програмних продуктів. Сучасні методи розробки програмного забезпечення (ПЗ) повинні уможлилювати зменшення витрат на реалізацію відповідного проекту з урахуванням функціональних вимог та атрибутів якості до ПЗ, а також скорочення часу, потрібного для виходу на ринок нових програмних продуктів [1]. Для цього в сучасній інженерії ПЗ широко застосовується концепція предметно-орієнтованого проектування (domain-driven design), в якій центральне місце займає поняття доменної моделі (domain model) як засобу для концептуалізації знань щодо предметної області (ПрО) розробки ПЗ [2].

Такий підхід дозволяє перейти від розробки окремих програмних систем (ПС) до створення лінійок програмних продуктів (software product lines - SPL) [3].

Створення лінійок програмних продуктів (ЛПП) передбачає побудову доменної моделі (ДМ) для заданої ПрО, на основі якої, із використанням відповідних інструментальних засобів (CASE-tools), можлива генерація каркасу програмного коду (source code framework), який потім має бути основою для створення програмних компонентів повторного використання (reusable software component), які застосовуються в ЛПП. При цьому існує можливість комплексного застосування як різних методів побудови ДМ для заданої ПрО, так і відповідних

технологічних середовищ для створення, накопичення та контролю різних версій ЛПП, і ці питання досліджуються в [4-5]. Зокрема, в цих роботах показано, що на ефективність цих процесів впливають такі чинники як: (1) ступінь повторного використання програмного коду, який може бути згенерований на основі певної ДМ; (2) структурно-функціональна складність самої ДМ.

Не менш важливою науково-прикладною проблемою розробки сучасних ЛПП є забезпечення в них відповідного рівня варіабельності (variability). Це поняття в сучасній програмній інженерії (software engineering) узагальнює такі підходи до розробки та супроводу ПЗ як: розробка на основі моделей (model-driven development), адаптивне проектування (adaptive design), побудова ПС, які здатні пристосовуватися до змін в середовищі їх функціонування (self-adaptable software), що можуть бути достатньо легко сконфігуровані та налаштовані на нові вимоги користувачів (configurable software), а також таких ПС, які пристосовані до їх подальшої еволюції (software evolution) та реінжинірингу (software reengineering) [1-3,7]. Для автоматизації цих процесів на різних стадіях життєвого циклу створення та супроводу ЛПП на теперішній час існує спеціалізований клас інструментальних засобів (CASE-засобів), які дозволяють проводити моделювання варіабельності компонентів ЛПП, з метою визначення певних показників якості їх побудови та подальшого застосування. Саме тому тема дослідження, представленою у цій статті, є актуальною і практично значущою.

Метою роботи є аналітичний огляд функціональності та визначення критеріїв вибору відповідних CASE – засобів для моделювання архітектур ЛПП з урахуванням конкретних показників їх якості, які обчислюються на основі оцінки рівня їх варіабельності із застосуванням кількісних метрик, що має забезпечити можливість робити висновки щодо ефективності побудови та подальшого застосування ЛПП. Для цього необхідно вирішити наступні завдання: проаналізувати сучасні інформаційні джерела за темою дослідження, провести порівняльний аналіз декількох CASE – засоби та обрати таких, що уможливають ефективне моделювання характеристик варіабельності у різних типах ЛПП, запропонувати метрики для їх кількісного визначення та провести відповідні експериментальні дослідження.

Стислий огляд останніх публікацій за темою дослідження

На теперішній час, поряд з великою кількістю публікацій щодо теоретичних підходів до побудови ЛПП з урахуванням характеристик їх варіабельності (напр., в [1, 3, 6]), значна увага дослідників привернута до питань розробки та застосування CASE-засобів, які дозволяють автоматизувати ці процеси на різних стадіях життєвого циклу створення та супроводу таких ЛПП. Зокрема, в останніх роботах за цією тематикою слід зазначити такі основні отримані в них результати:

– в [7] надано стислий огляд концептуальних засад розробки ЛПП із урахуванням властивостей варіабельності їх компонентів, включаючи опис формальних мов для моделювання цих процесів, таких як TVL (Text-based Variability Language), CVL (Common Variability Language), SXFM (Simple XML Feature Model), а також зроблено огляд відповідних інструментальних засобів;

– в [8] зроблено систематизований аналітичний огляд більш ніж 50 інструментальних програмних засобів для підтримки функцій управління варіабельністю в процесах розробки ЛПП, при цьому найбільшу увагу приділено тим, які використовують FODA (Feature Oriented Domain Analysis) – моделювання, при цьому наведені цікаві статистичні дані щодо оцінки якості окремих засобів їх користувачами за такими критеріями як продуктивність (Performance), масштабованість (Scalability), здатність до інтеграції з іншими системами (Integration) та зручність використання (Usability);

– в [9] подано найновітній (станом на 2019р) огляд проблем ЛПП на основі модельно-кеованого підходу (), при цьому значну увагу приділено засобам та технологіям для моделювання варіабельності в ЛПП, а саме моделям властивостей (Feature Model), моделям варіабельності архітектури (Product Line Architecture Variability), орто-

гональним моделям варіабельності (Orthogonal Variability Model).

Але слід зазначити, що в цих та у більшості інших наявних публікацій, практично відсутні дані щодо можливості та доцільності застосування у таких інструментальних засобах кількісних метрик для визначення рівня варіабельності, та дослідження можливості надавати практичні рекомендації щодо підвищення ефективності побудови та подальшого використання відповідних ЛПП.

Основна частина

1. Аналіз основних властивостей деяких CASE – засобів для моделювання варіабельності в ЛПП. Один з найбільш змістовних оглядів джерел за тематикою розробки та застосування сучасних інструментальних CASE-засобів, який наведено в [8], визначив близько 40 таких систем для розробки та моделювання характеристик ЛПП. На основі обробки статистичних даних щодо відгуків користувачів, один з прикладів яких наведено на рис. 1, можливо сформулювати певні критерії вибору відповідних CASE-засобів, а також зробити певні висновки щодо їх переваг та недоліків.

Шляхом аналізу та узагальнення таких агрегованих даних з [8] було визначено, що для подальшого дослідження в цій роботі були обрані такі конкретні CASE-засоби як SPLOT, FeatureIDE та VariaMos, результат порівняльного аналізу функціональних можливостей яких наведено в табл. 1.

Умовні позначення в табл.1 є наступними: «+» - відповідна функціональність присутня, «-» - відсутня, «+/-» - присутня частково. Нижче наведено більш детальний опис цих систем [7, 8].

SPLOT (Software Product Line Online Tool).

Цей інструментальний засіб є некомерційним програмним продуктом з відкритим вихідним кодом, який розроблено на платформі Java [7]. Для опису побудованих моделей ЛПП використовується мова SXFM (Simple XML Feature Model), для чого створено спеціальний графічний Web-редактор. Для SPLOT слід відмітити такі три найбільш сильні сторони: наявність редактора моделей ЛПП, автоматичний аналіз властивостей вже побудованих моделей та наявність онлайн-доступу для всіх основних ре-

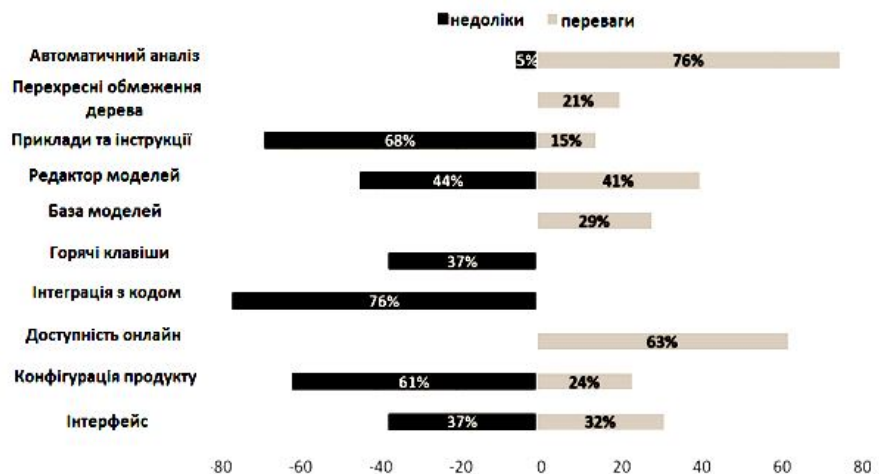


Рис. 1. Приклад статистичних даних щодо переваг та недоліків засобу SPLOT [8]

жимів роботи. До слабких сторін системи можна віднести: недоліки інтерфейсу користувача у редакторі моделей, недостатню кількість прикладів застосування цього продукту у його наявній документації та відсутність можливості генерації вихідним коду на основі вже наявних моделей ЛПП.

Таблиця 1 – Результати порівняльного аналізу обраних CASE-засобів

Засіб \ Властивість	SPLOT	Feature-IDE	VariaMos
Налаштування конфігурацій	+	+	
Збереження моделей	+	+	+
Підтримка та оновлення	+	+	+/-
Автоматичний аналіз моделей	+	+	+
Симуляція процесів часу виконання	-	-	+
Засоби інтеграції з іншими програмними системами	-	+	+
Можливість генерації коду	-	+	-

FeatureIDE. Інструмент FeatureIDE (Integrated Development Environment) є також некомерційним продуктом, який створено на платформі Eclipse і який передбачає можливість використання мов програмування C / C++, Java, Haskell, а також стандарт XML для опису та зберігання специфікацій моделей ЛПП [8]. Можливо стверджувати, що його перевагами додатково до характеристик SPLOT є такі як: наявність відповідних модулів (plug-in) для середовища Eclipse та можливість генерації вихідного коду на різних мовах програмування на основі побудованих моделей властивостей ЛПП.

Як основний недолік більшість користувачів FeatureIDE (прибл. 64%) зазначили проблеми графічного інтерфейсу, зокрема, складність навігації в системі меню для автоматичного аналізу моделі та конфігурації продукту. У цьому контексті досить важливими є інтерактивні механізми (наприклад, анімації, кольоровий відтінок та підсвічування), що використовуються для підтримки роботи користувачів, які переміщуються у дереві проекту Eclipse, вибирають функції та мають встановлювати взаємозалежність між ними.

VariaMos (Variability Models). Засіб моделювання варіабельності VariaMos також розроблено з використанням мови Java і тому його можна застосовувати в різних операційних середовищах (MS Windows, Mac OS and Linux), а для створення конфігураційних файлів та експорту / імпорту даних в ньому використовується формат JSON [10].

Як перевагу VariaMos слід назвати можливість побудови двох типів попередньо визначених метамоделей ЛПП: так званих адаптивних та функціональних моделей. Відповідно до обраного типу метамоделі, потім в цій системі можливо побудувати локальні моделі відповідної ЛПП: модель варіабельності (variability model), модель контексту (context model) та модель активів (assets model).

VariaMos також забезпечує імітаційні операції, такі як повторення всіх рішень часткової конфігурації, візуалізацію можливих адаптацій системи, оцінку рішення конфігурації та пропозицію альтернатив [10].

На основі результатів проведеного порівняльного аналізу було виявлено, що FeatureIDE має потужніші можливості для створення Feature моделей, але не має можливостей для моделювання та симуляції створення ДЛПП, які має VariaMos. Обидва CASE засоби мають можливість інтеграції компонентів моделей з вихідним кодом для управління розробкою ЛПП, де можна побачити які модулі все розроблені, а які ні, якої бракує SPLOT. Додатково, FeatureIDE має можливість генерації коду продукту з вихідного коду окремих модулів відповідно до вибраної конфігурації продукту. Отож, було прийнято рішення використати та протестувати FeatureIDE та VariaMos та провести практичне порівняння результатів застосування цих CASE-засобів для моделювання показників якості ЛПП.

2. Метрики якості ЛПП з урахуванням показників її варіабельності. Слід зазначити, що більшість загальновідомих метрик якості ПЗ (див., на пр., в [11]), які базуються на обчисленні певних характеристик вихідного коду, є недостатніми для визначення показників якості ЛПП. Вони не дозволяють визначити рівень варіабельності, як однієї з основних показників якості ЛПП. Слід зазначити, що сама концепція створення ЛПП передбачає два рівня опису її побудови, а саме [12]:

(i) рівень опису певної еталонної архітектури тих ПС, які є окремими програмними продуктами, або членами у відповідній ЛПП (SPL members), це так звана архітектура продуктової лінії (Product Line Architecture - PLA);

(ii) рівень опису окремих програмних компонентів, що входять до складу окремих ПС, які є членами ЛПП.

Саме тому відповідні метрики варіабельності повинні враховувати характерні особливості та кількісні параметри опису ЛПП на цих рівнях (i) – (ii), і деякі з них представлені нижче [12].

Коефіцієнт варіабельності структури (Structure Variability Coefficient - SVC) для певної ЛПП може бути обчислений за формулою:

$$SVC = |C_V| / (|C_C| + |C_V|), \quad (1)$$

де $|C_C|$ – кількість спільних, а $|C_V|$ – кількість змінних (тобто варіабельних) програмних компонентів в усіх продуктах, які належать до цієї ЛПП. Відповідно до цієї формули, значення цього показника належить до інтервалу [0,1] і чим воно є вищим, тим більш ефективним є повторне використання такої ЛПП.

Частота повторного використання компонентів (Component Reusability Rate - CRR) для певної ЛПП визначається наступною формулою

$$CRR = \sum_i Ex(M_i) / |M|, \quad (2)$$

де M – це загальна кількість програмних продуктів у

цій ЛПП, а $Ex(M_i)$ є булева функція, як визначається за наступними правилами: $Ex(M_i) = 1$, якщо компонент є внутрішнім компонентом для M_i продукту ЛПП, інакше $Ex(M_i) = 0$. Якщо всі компоненти програмні компоненти є спільними у всіх продуктах цієї ЛПП, то $CRR = 1$.

Коефіцієнта корисності повторного використання (Reuse Benefit Rate-RBR) компонентів ЛПП, який визначається за наступною формулою:

$$RBR = \sum_i Size(M_i) / \sum_j Size(C_j), \quad (3)$$

де $Size(C_j)$ – кількість рядків (line of code) вихідного коду компонента C_j у певній ЛПП, $\sum_j Size(C_j)$ – це сума рядків коду всіх компонентів у ЛПП, а $Size(M_i)$ – це кількість рядків коду у продукту M_i цієї ЛПП. У свою чергу, значення $Size(M_i)$ також може бути розраховано як

$$Size(M_i) = \sum_k Size(C_k), \quad (4)$$

де $\sum_k Size(C_k)$ – це рядки коду всіх компонентів у коду у продукту M_i цієї ЛПП. Слід зазначити, що значення метрики RBR не є нормованим, тобто може бути більше одиниці.

3. Експериментальне дослідження CASE – засобів та аналіз отриманих результатів. Для експериментального дослідження обраних CASE – засобів (п. 1) із використанням вищенаведених метрик (1) – (4), у середовищі FeatureIDE були побудовані тестові моделі 2-х умовних ЛПП, а саме:

- 1) ЛПП, членами якої є програмні редактори фотографій;
- 2) ЛПП, членами якої є програмні редактори зображень (рис. 2 та 3 відповідно).

На цих діаграмах представлені їх загальні компоненти (вони мають зв'язки, що позначені чорними кульками) та змінні компоненти (позначені прозорими кульками). Функціональність окремих компонентів цих ЛПП є цілком зрозумілою з їх найменувань, напр., на рис. 2 це: формат зображення (Format), зміна розміру (Change_size) та ін.

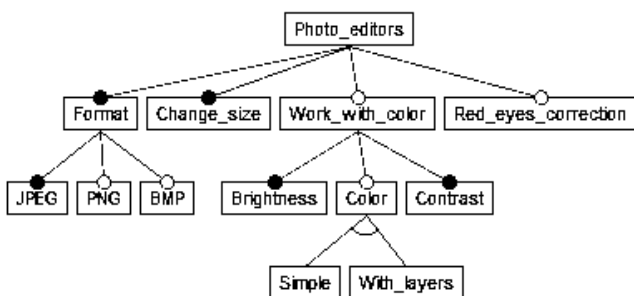


Рис. 2. Модель ЛПП редакторів фотографій

Використовуючи формулу (1) ми можемо розрахувати коефіцієнт варіабельності структури (SVC) для цих моделей ЛПП, які мають 11 спільних функцій та 3 відмінних відповідно, тобто:

$$SVC = 3 / (11+3) = 3 / 14 = 0,214.$$

Цей результат є цілком зрозумілим з точки зору Про застосування цих ЛПП, бізнес-логіка яких є

подібною, і таким чином, їх рівень варіабельності є доволі низьким. У реальному проекті такий результат є підставою зробити висновок щодо доцільності об'єднання цих ЛПП в одну цільову систему.

Для дослідження метрик ЛПП, які визначаються за формулами (2) – (4), у середовищі CASE – засобу VariMos, в режимі симуляції моделей варіабельності, були побудовані FODA – моделі для 3-х різних продуктів (SPL - members) у складі тестової ЛПП, яка була раніше представлена на рис. 3 (ці моделі наведені на рис. 4 – 6 відповідно).

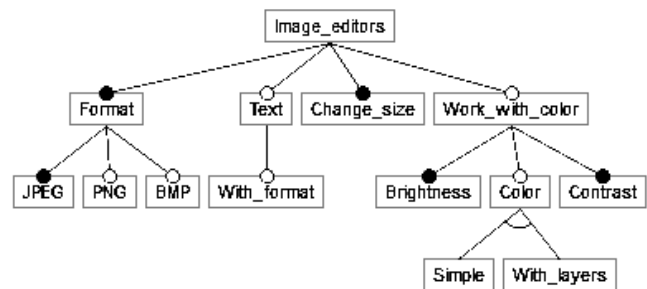


Рис. 3. Модель ЛПП редакторів зображень

Вони відрізняються варіабельними компонентами, які мають спеціальні візуальні позначки - орієнтовані праворуч трикутники у своїх верхній частині. Для цих 3-х моделей значення коефіцієнту повторного використання компонентів (CRR) обчислюється за формулою (2), і нижче наведено його значення для наступних компонентів ЛПП: «Текст», «Формат - PNG» та «Колір - простий», а саме:

$$CRR(\text{«Текст»}) = (0+1+0)/3 = 1/3 = 0.33 ;$$

$$CRR(\text{«Формат - PNG»}) = (1+0+1)/3 = 2/3 = 0.67;$$

$$CRR(\text{«Колір - простий»}) = (0+0+0)/3 = 0.$$

З точки зору програмної реалізації відповідних ЛПП це показує те саме компонент «Формат - PNG» має найвищий рівень варіабельності, тобто його використання є доцільним у всіх наявних та перспективних продуктах цієї ЛПП.

Для обчислення коефіцієнта корисності повторного використання варіабельних компонентів (RBR) у цих FODA - моделях потрібно мати дані щодо кількості рядків коду компонентів ЛПП, умовний приклад яких при реалізації на мові програмування Java представлено у табл. 2:

Таблиця 2 – Тестові дані про кількість рядків Java коду у кожному з компонентів ЛПП

Назва компонента	Кількість рядків коду
«Формат»	343
«Формат – PNG»	258
«Формат – BMP»	177
«Формат – JPEG»	202
«Текст»	44
«Форматування тексту»	663
«Зміна розміру»	50
«Робота з кольором»	36
«Яскравість»	48
«Колір»	89
«Колір – простий»	101
«Колір – за шарами»	209
«Контраст»	73

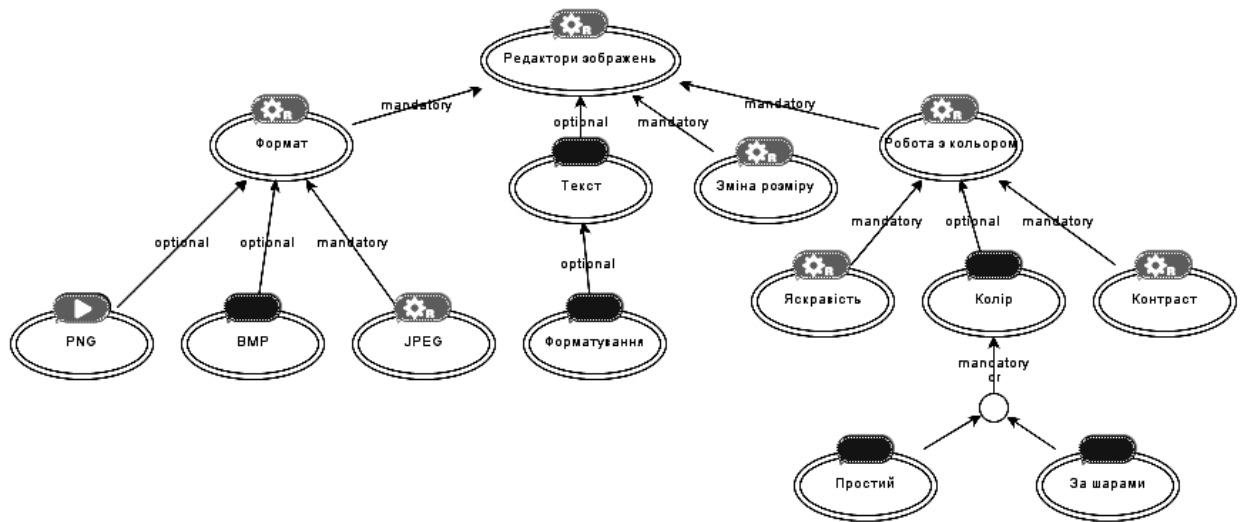


Рис. 4. Перший продукт у ЛПП

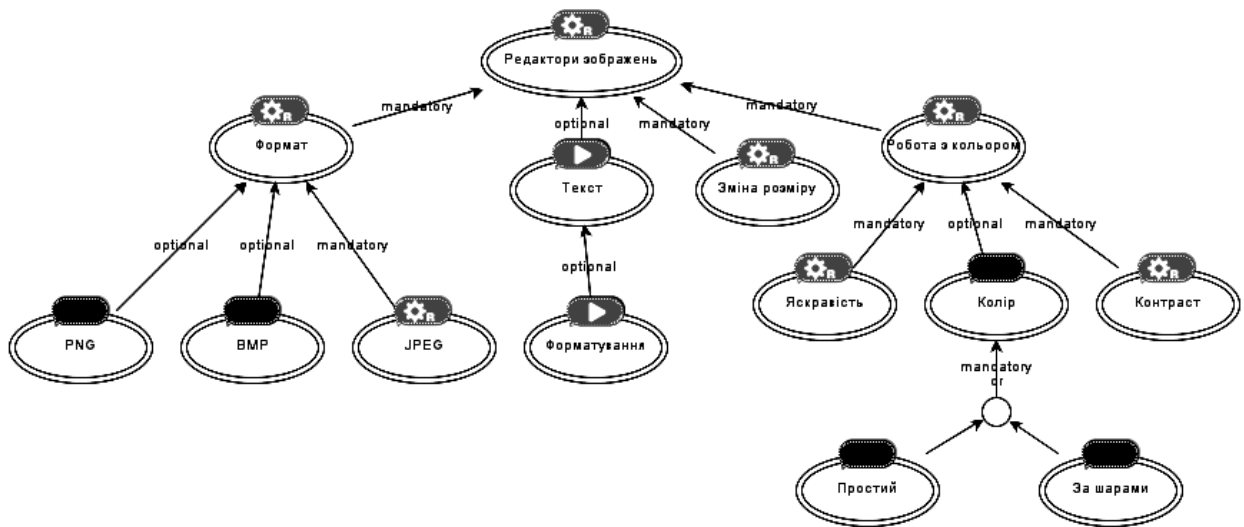


Рис. 5. Другий продукт у ЛПП

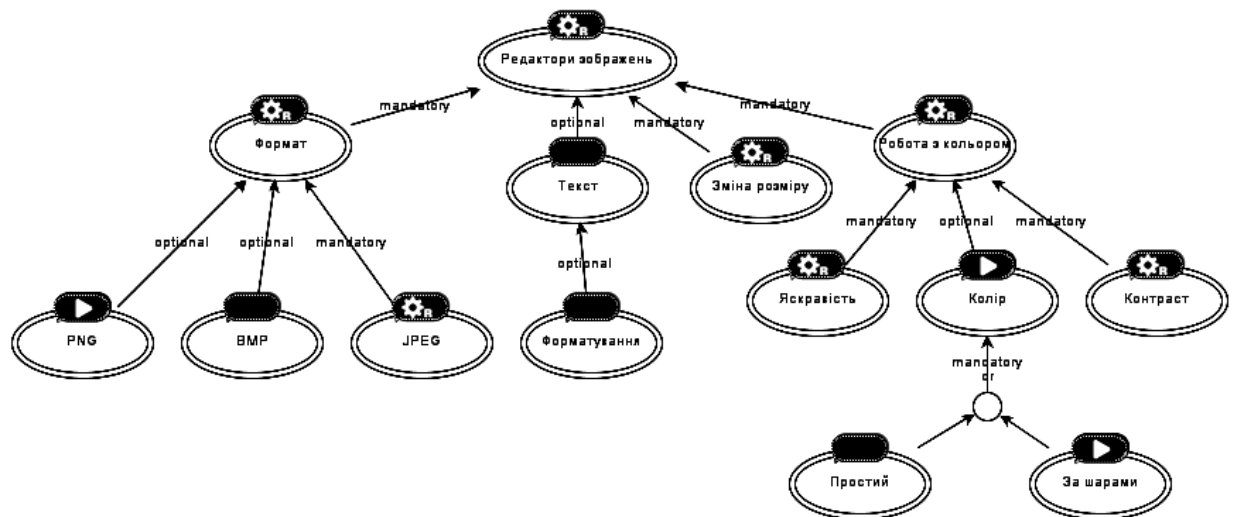


Рис. 6. Третій продукт у ЛПП

Після отримання цих даних, використовуючи формули (3) і (4), з урахуванням характеристик варіабельності кожного з вищезазначених компонентів «Текст», «Формат - PNG» та «Колір - простий» (див. їх FODA – моделі на рис. 4-6), отримуємо таке зна-

чення цього коефіцієнту:

$$RBR = (1010+1459+1308) / 2293 = 3777 / 2293 = 1.65,$$

що підтверджує доцільність їх сумісного використання в окремих продуктах цієї ЛПП.

Висновки і напрямки подальших досліджень

В роботі представлені результати порівняльного огляду функціональних можливостей сучасних CASE – засобів для моделювання характеристик варіабельності в ЛПП, мотивовано обрано метрики для кількісного визначення рівня варіабельності та експериментально досліджено можливість їх застосування для аналізу ефективності процесів розробки та подальшого використання ЛПП. Побудовані тестові моделі умовних ЛПП, проведені їх експериментальні дослідження, які показали, що за умов роз-

ширення функціональності наявних CASE – засобів шляхом застосування кількісних метрик для визначення рівня варіабельності можливо надавати практичні рекомендації щодо підвищення ефективності побудови та подальшого використання відповідних ЛПП.

Напрямок подальших досліджень є розробка програмних рішень для вдосконалення функціональності існуючих CASE – засобів з урахуванням кількісних метрик варіабельності компонентів ЛПП та проведення з їх застосуванням більшої кількості експериментальних досліджень ефективності запропонованого підходу.

СПИСОК ЛІТЕРАТУРИ

1. Лавріщева К. М. Нові теоретичні засади технології виробництва сімейств програмних систем у контексті генерувального програмування: монографія. К.: Ін-т програм. систем. НАН України, 2011. 277 с.
2. Эванс Э. Предметно-ориентированное проектирование (Domain-Driven Development - DDD): Структуризация сложных программных систем. 2011. 448 с.
3. Reinhartz-Berger I. Domain Engineering: Product Lines, Languages, and Conceptual Models. Heidelberg, Springer, 2013.
4. Мартінкус І. О., Ткачук М. В., Гамзаєв Р. О. Конструювання лінійок програмних продуктів із застосуванням доменного моделювання та метрик повторного використання коду. *Системи управління, навігації та зв'язку*: зб. наук. пр. Полтава: ПНТУ, 2017. – Вип. 3 (43). – С. 93-97.
5. Tkachuk, M., Martinkus, I., Gamzayev, R. An Integrated Approach to Evaluation of Domain Modeling Methods and Tools for Improvement of Code Reusability in Software Development. Heinrich C. Mayr, Martin Pinzger (Eds.): *INFORMATIK 2016, Lecture Notes in Informatics*. Vol. P-259: Kollen Druck+Verlag GmbH, Bonn, 2016. P. 143-156.
6. Tkachuk Mykola, Gamzaev Rustam, Martinkus Iryna. Towards Effectiveness Assessment of Domain Modelling Methods and Tools in Software Product Lines Development. *Enterprise Modelling and Information Systems Architectures – International Journal of Conceptual Modeling*. Vol. 13 (2018), Germany. P. 190-206.
7. Koznov D., Novitsky I., Smirnov M. Variability Management Tools: Readiness for Industrial Use. *SPIIRAS Proceedings*. 2013. Issue 3(26). –P. 297-331.
8. Bashroush R., Garba M., Rabiser E. CASE Tool Support for Variability Management in Software Product Lines. *ACM Computing Surveys*. 50 (1). 2017. P. 1-45.
9. Yousaf N., Akram M., Bhatti A. Investigation of Tools, Techniques and Languages for Model Driven Software Product Lines (SPL): A Systematic Literature Review. *Journal of Software Engineering and Applications*, 2019, 12. P. 293-306.
10. Mazo, R., Muñoz-Fernández J., Rincón-Perez L. VariaMos: An extensible Tool for Engineering Dynamic Product Lines. *Proceedings of SPLC'2015 Confegence*, July 2015, Nashville, USA. P. 374-379.
11. Андон Ф.И., Коваль Г.И., Коротун Т.М. Основы инженерии качества программных систем. 2-е изд. К.: Академперіодика, 2007 – 672 с.
12. Zhang, T., Deng, L., Wu, J. Some Metrics for Accessing Quality of Product Line Architecture. *International Conference on Computer Science and Software Engineering*, 2008.

Received (Надійшла) 23.12.2019

Accepted for publication (Прийнята до друку) 06.02.2020

Analysis and experimental research of CASE – tools for variability modeling in software product lines development

M. Tkachuk, R. Gamzayev, O. Tovstokorenko, I. Martinkus, K. Nagorni

Abstract. The subject of study in the article are CASE - tools and metrics for modeling and research the variability characteristics of architectures and particular components in software product lines (SPL). **The goal** is to provide an analytical review of functionality and determination of selection criteria of the appropriate CASE - tools for modeling SPL architectures taking into account specific indicators of their quality, which are calculated on the basis of assessing their level of variability using quantitative metrics, which should provide the opportunity to draw conclusions about the effectiveness of the construction and further use of LPP. The **tasks** are: to analyze modern information sources on the research topic, to conduct a comparative analysis of several CASE - tools and to choose those to provide effective modeling of the variability characteristics in various types of SPLs, to propose metrics for their quantitative determination and conduct relevant experimental studies. The **methods** used are: structural analysis of the design features of SPL; quantitative metrics for determining the level of variability of the architecture and particular components in SPL. The following **results** were obtained: the functional capabilities of modern CASE - tools for modeling the variability characteristics in SPL were analyzed, metrics for quantitatively determining the level of variability were selected, and the possibility of their usage to analyze the effectiveness of the development processes and further utilize of SPL was experimentally investigated. The **conclusions** are: the experimental studies were conducted for testing SPL models, and they showed that under the conditions of expanding the functionality of the existing CASE-tool, it is possible to formulate practical recommendations for improving the construction efficiency and further use of the corresponding SPL by using quantitative metrics to determine their variability level. The direction of **further research** is the development of software solutions to improve the functionality of existing CASE – tools, and performance with their usage more experimental studies on effectiveness of the proposed approach.

Keywords: software product line, variability, architecture, component, CASE - tool, metric.