

## ПІДХІД ДО ПОБУДОВИ ОБ'ЄКТНО-КОМПОНЕНТНОЇ МОДЕЛІ СІМЕЙСТВА ПРОГРАМНИХ ПРОДУКТІВ

Модель властивостей Сімейства програмних продуктів (СПП) розвинуто об'єктно-компонентним методом проектування ПС (К.М. Лавріщева, В.М. Грищенко). Для опрацювання виявлених обмежень моделі її подано графом складних об'єктів, що відображають функції програмних продуктів (ПП) з СПП разом з їх даними та відношеннями варіантного підпорядкування й залежності. Його конкретизовано графами: інтерфейсних об'єктів (для формалізації взаємодії функцій), інтерфейсів та компонентів повторного використання (КПВ), що реалізують інтерфейси. Показано, що відображення між графами – ізоморфізми алгебраїчних моделей, які зіставляють властивостям ПП його реалізацію збіркою КПВ – образів властивостей. Графи й відображення поєднано в інтегрованій об'єктно-компонентній моделі СПП. Надано її переваги для виведення варіантів ПП, відстеження варіабельності й підтримки еволюції СПП.

### Вступ

Стрімкий розвиток індустрії програмного забезпечення в світі й Україні наразі вимагає створення методологій індустріального виробництва ПП з гарантовано високою якістю, низькою вартістю і стислими термінами випуску, адаптованих до змінних умов використання. Серед численних концепцій, що опрацьовують цей виклик, важливе місце займає інженерія Сімейств ПП (Software Product Lines). Її сутність – заміна автономних ПП (зазвичай для конкретних замовників) СПП ринкового призначення для підтримки ділових процесів певного цільового домену.

Сімейство – набір ПП з керованою множиною спільних властивостей, відповідних потребам кінцевих споживачів у домені, які розробляються попередньо визначеним способом із спільної множини ресурсів повторного використання (РПВ) [1]. Формальне подання складу і взаємозв'язків спільних і розбіжних властивостей ПП традиційно зветься моделлю СПП [2]. Саме вона визначає спосіб розроблення ПП і підтримує узгодженість властивостей, РПВ і артефактів під час еволюції СПП.

Серед сучасних моделей СПП однією з найперспективніших визнано модель властивостей (feature model) FM [1–3]. Численні дослідження фокусуються на аспектах її застосування в СПП:

– відтворенні властивостей з коду та специфікацій наявних ПП [4, 5];

– моделюванні властивостей на підставі потреб споживачів і розробників СПП з усіх функціональних сегментів домену [6];

– оцінюванні рівня відповідності цим потребам спектра спільних і розбіжних властивостей ПП з СПП [7];

– керуванні варіантами ПП [8].

Узагальнення їх результатів наразі висвітлює два обмеження ефективного застосування FM в СПП: труднощі повторного використання фрагментів коду пов'язаних із властивостями та слабку відстежуваність зв'язків між властивостями, РПВ і артефактами. Вони стають особливо критичними за сучасних умов індустріального виробництва ПП: поповнення традиційних характеристик якості ПП їх ситуаційною адаптованістю до неочікуваних змін потреб споживачів і умов використання та зростання інтенсивності таких змін. Отже, нині набуває актуальності проблема доопрацювання цих обмежень.

Для її вирішення є перспективним розвиток FM об'єктно-компонентним методом (ОКМ), створеним і апробованим К.М. Лавріщевою, В.М. Грищенком для проектування програмних систем у парадигмі збіркового програмування [9]. Водночас, такий розвиток складає актуальну проблему парадигми. Його реалізація й дослідження механізмів застосування отриманої моделі в СПП є метою роботи.

Стаття підсумовує результати досліджень за проектом «Розробка теоретич-

ного фундаменту генеруючого програмування та інструментальних засобів його підтримки», виконаного в ІПС НАН України під керівництвом доктора фізикоматематичних наук, професора К.М. Лаврішевої [10].

### Сутність об'єктно-компонентного розвитку моделі властивостей

Сутність пропонованого підходу полягає у фіксації цільових вимог до моделі СПП, що сприяють доопрацюванню обмежень ФМ із збереженням її відомих переваг [1–3, 6, 7], та реалізації вимог у формованій моделі.

На підставі аналізу результатів за основними напрямками досліджень ФМ [1–8] пропонується низка вимог:

а) відтворення ФМ з наявних ПП (їх первинного коду або специфікацій);

б) узгодження відтвореної ФМ з поточними потребами в ПП усіх учасників розроблення СПП в усіх функціональних сегментах цільового домену;

в) підтримка автоматизованого рефакторингу наявних ПП для підвищення їх якості та ефективності повторного використання й реверсної інженерії в СПП;

г) підтримка автоматизованого конструювання ПП з СПП з оптимальним повторним використанням ресурсів (РПВ);

д) забезпечення адаптовності нових ПП до передбачених і непередбачених змін потреб споживачів та умов використання споживачами і розробниками СПП;

е) придатність до узгодженої автоматизованої актуалізації разом з усіма РПВ;

ж) відстежуваність взаємних впливів на всі елементи моделі змін у властивостях, РПВ та артефактах розроблення ПП;

з) підтримка трасовності робочих продуктів процесу розроблення нових ПП: від вимог до коду компонентів повторного використання (КПВ), взаємодіючих у ПП.

Побудова й апробація моделі передбачає дев'ять послідовних кроків.

1. Зіставлення припустимим функціям ПП оброблюваних даних та формування об'єктної моделі: графа “складених” властивостей з явно поданими точками варіабельності, поєднаного зв'язками пере-

творюваності з графом інтерфейсних об'єктів, що описують взаємодію перетворюваних функцій за допомогою зіставлених їм даних.

2. Формування інтерфейсної моделі – виокремлення інтерфейсів взаємодії функцій за допомогою даних та впорядкування їх згідно з відношеннями між “складеними” властивостями в об'єктній моделі, які зіставляють точкам варіабельності варіабельні інтерфейси.

3. Перетворення інтерфейсної моделі в компонентну – зіставлення кожному інтерфейсу програмного компонента повторного використання (КПВ), де він є вхідним.

4. Запровадження узгоджених базових операцій композування й декомпозиції на об'єктах – складених властивостях – та КПВ.

5. Поєднання об'єктної, інтерфейсної, компонентної моделей в інтегрованій об'єктно-орієнтованій моделі СПП.

6. Подання основних артефактів процесу розроблення ПП вкладеними підграфами інтегрованої моделі.

7. Запровадження узгоджених композицій базових операцій над об'єктами та КПВ на підтримку формування нових ПП як різномірних варіантів ПП, що вже наявні.

8. Розвиток механізмів застосування інтегрованої моделі СПП у процесі його розроблення.

9. Створення технології розроблення СПП, керованого інтегрованою об'єктно-компонентною моделлю, та її апробація в середовищі інструментально-технологічного комплексу ІПС НАНУ [10].

### Формальний опис об'єктно-компонентної моделі СПП

**Об'єктна модель СПП.** Для побудови об'єктної моделі застосовується об'єктний аналіз – перша фаза ОКМ [9]. Його мета – подання цільового домену СПП множиною взаємодіючих об'єктів з властивостями, достатніми для другої фази ОКМ – апріорної ідентифікації необхідних КПВ та способу їх композування в ПП.

Аналіз полягає у послідовному формуванні чотирьох рівнів подання домену:

- 1) узагальнюючого – як множини об’єктів із цілком визначеною поведінкою;
- 2) структурного – графом об’єктів, утвореним їх зв’язками “частина-ціле”;
- 3) характеристичного – графом об’єктів, доповнених предикатами їх властивостей, істотних для проектування ПП;
- 4) поведінкового – індукованим графом взаємозв’язків між станами об’єктів.

Склад і взаємозв’язки рівнів 1)–4) для СПП показано на рис. 1.

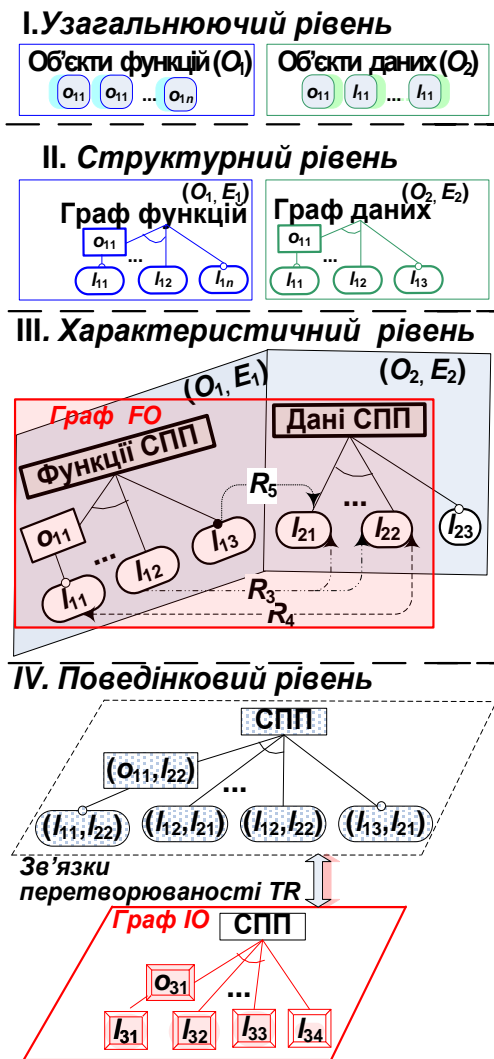


Рисунок 1. Об’єктний аналіз цільового домену для СПП

Згідно з рисунком, на узагальнюючому рівні виокремлено класи об’єктів, що подають два типи варіабельних властивостей ПП з СПП – їх функції ( $O_1$ ) та оброблювані дані ( $O_2$ ). На наступному рівні на множинах  $O_1, O_2$  встановлено властиві FM синтаксичні відношення [5] безумовного ( $R_1$ ) і варіантного ( $R_2$ ) підпорядкування та

обмеження наслідку ( $R_3$ ), зіставлення ( $R_4$ ) й виключення ( $R_5$ ):

$$E_t = \cup_{i=1, \dots, 5} R_i \subseteq \otimes^2 O_t, t = 1, 2. \quad (1)$$

Далі, на характеристичному рівні, на підтримку вимог конструктивності а)–е) функцію  $o_1 \in O_1$  подано кортежем сигнатур методів її реалізації  $M(o_1)$ , а об’єкт даних  $o_2 \in O_2$  – кортежем  $D(o_2)$  ознак певних типів [2], оброблюваних  $M(o_1)$ . Для кожної функції  $o_1 \in O_1$  задані обмеження щодо даних

$$E_{12} = \cup_{i=3,4,5} R_i \subseteq O_1 \otimes O_2, \quad (2)$$

розглядувані як додаткові властивості  $o_1$  і показані на рис. 1 штрих-пунктирною, штриховою і точковою лініями.

Нарешті, на поведінковому рівні сформовано двобічні зв’язки перетворюваності складених властивостей  $(o_1, o_2) | o_1 R_3 \cup R_4 o_2$  в інтерфейсні об’єкти  $o_3 \in O_3$ , які підтримують (варіантні) взаємодії між функціями ПП з СПП за допомогою даних. Стан  $o_3$  подається парою  $\langle M(o_1), D(o_2) \rangle$ . Відношення (1), (2) індукують відношення  $E_3$  на множині  $O_3$ , перетворюючи її у граф

$$IO = (O_3, E_3), E_3 = \cup_{i=1, \dots, 5} R_{3i}. \quad (3)$$

Умовами коректності аналізу є:

- 1) відсутність в елементарних функцій (листіків графа  $(O_1, E_1)$ ) спільних методів, а в елементарних об’єктів даних (листіків  $(O_2, E_2)$ ) – відповідно, спільних ознак;

$$2) \forall o_1 \in O_1 \exists o_2 \in O_1 | o_1 R_3 o_2 \vee o_1 R_4 o_2;$$

$$3) o_1 R_3 \cup R_4 o_2 \Rightarrow \neg(o_1 R_5 o_2);$$

$$o_1 R_3 o_2 \Leftrightarrow \neg \exists (o_1^*, o_2^*) | o_1^* R_i o_2^* \vee o_1 R_i o_2^*, i=3, 4.$$

Для відношень  $R_i$  (1), (2) пропонується об’єктно-орієнтована конкретизація.

**Означення 1.** Будемо говорити, що  $o_{1i} \in O_t, o_{2j} \in SG \subseteq O_t \setminus \{o_{1i}\}, t = 1, 2$  пов’язані:

- 1) безумовним підпорядкуванням  $(o_{1i} R_{t1} o_{2j})$ , якщо реалізація в ПП об’єкта  $o_{1i}$  впливає з реалізації  $o_{2j}$ :

$$M(o_{1i}) = \cup \{M(o_{2j}), o_{1i} R_{11} o_{2j}\}; \quad (4)$$

$$D(o_{2i}) = \cup \{D(o_{2j}), o_{2i} R_{21} o_{2j}\};$$

$$M(o_{1i}) = \cup Ch_{m,n}(\{M(o_{2j}), o_{1i} R_{11} o_{2j}\}); \quad (5)$$

$$D(o_{2i}) = \cup Ch_{m,n}(\{D(o_{2j}), o_{2i} R_{21} o_{2j}\});$$

- 2)  $(m, n)$ -варіантним підпорядкуванням  $(o_{1i} R_{12}(m, n) o_{2j}, 0 \leq m \leq n < |O_t|)$ , якщо

реалізація в ПП об'єкта  $o_{ii}$  впливає з одночасної реалізації від  $m$  до  $n$  елементів  $SG$ , де  $Ch_{m,n}(X)$  – операція вибору не менше  $m$  і не більше  $n$  елементів множини  $X$ .

Об'єкт  $o_{ii}$  назвемо точкою варіабельності, а  $o_{ij}$  – варіантом її реалізації в ПП;

3) обмеженням зіставлення  $R_3 \subset \subset \otimes^2 O_1 + \otimes^2 O_2 + O_1 \otimes O_2$ , якщо пов'язані ним об'єкти мають реалізуватися в ПП одночасно. Зокрема,  $o_1 R_3 o_2$  означає, що задане відношення  $RL(o_1, o_2)$  зіставлення кожній унікальній змінній з сигнатур методів  $M(o_1)$  ознаки з  $D(o_2)$ , і тільки такі пари можуть реалізуватися в ПП;

4) обмеженням потреби  $R_4 \subset \otimes^2 O_1 + \otimes^2 O_2 + O_1 \otimes O_2$ , якщо реалізація першого об'єкта потребує реалізації другого, тобто  $RL(o_1, o_{2j})$ , задані для всіх  $o_{2j}$ , зіставляють змінним з  $M(o_1)$  всі припустимі ознаки з  $D(o_2)$ ;

5) обмеженням виключення  $R_5 \subset \subset \otimes^2 O_1 + \otimes^2 O_2 + O_1 \otimes O_2$ , якщо реалізація першого об'єкта зумовлює неможливість реалізації другого. Зокрема,  $o_1 R_5 o_2$  означає, що задане відношення зіставлення унікальним змінним з сигнатур  $M(o_1)$  ознак з  $D(o_2)$ , заборонених до використання в ПП.

Зауваження 1. Базовими формами варіантного підпорядкування є: опціональне  $(0,1)$ ; альтернативне  $(1,1)$ ; опціонально-варіантне  $(0;n)$ ; варіантне  $(1,n)$ .

Пропонований спосіб формування інтерфейсних об'єктів та введення індукованих відношень між ними  $R_{3i}$  (5) фіксує

*Означення 2.* Нехай  $O^* = \{(o_1, o_2) \parallel R_{3i} o_2 \vee o_1 R_{4i} o_2\}$  – множина складених властивостей ПП з СПП. Зв'язки перетворюваності задаються відображенням

$$tr: O^* \leftrightarrow O_3, tr(o^*) = \langle M(o_1), D(o_2) \rangle. \quad (6)$$

Мають місце відношення:

$o_{3i} R_{33} o_{3j}$ , якщо методи  $M(o_{1j}), M(o_{1i})$  реалізуються в ПП одночасно;

$o_{3i} R_{34} o_{3j}$ , якщо реалізація методів  $M(o_{1i})$  потребує реалізації  $M(o_{1j})$ ;

$o_{3i} R_{35k} o_{3j}$ , якщо реалізація методів  $M(o_{1i})$  унеможливує реалізацію  $M(o_{1j})$ .

$$o_{3i} R_{31} o_{3j} \Leftrightarrow M(o_{1j}) \subset M(o_{1i}) \vee (M(o_{1j}) = M(o_{1i}) \wedge D(o_{2j}) \subset D(o_{2i})); \quad (7)$$

$$o_{3i} R_{32}(m,n,p,q) o_{3j} \Leftrightarrow Ch_{m,n}(M(o_{1j})) \subset M(o_{1i}) \vee (M(o_{1j}) = M(o_{1i}) \wedge Ch_{p,q}(D(o_{2j})) \subset D(o_{2i})),$$

$$o_{3i} = \langle M(o_{1i}), D(o_{2i}) \rangle, o_{3j} = \langle M(o_{1j}), D(o_{2j}) \rangle \in O_3.$$

Зауваження 2. Послідовна декомпозиція виразу (6) згідно з виразами (4) і (5) дозволяє подати його в термінах елементарних функцій і даних – листків  $l_t \in L_t \subset O_t$  графів  $(O_t, E_t)$ ,  $t=1,2$ :

$$tr(o_1, o_2) = \langle \langle M(l_1), l_1 ID_1 o_1 \rangle;$$

$$\langle Ch_{0,1}(M(l_1)), l_1 IV_1 o_1 \rangle;$$

$$\langle D(l_2), l_2 ID_2 o_1 \rangle; \langle Ch_{0,1}(D(l_2)), l_2 IV_2 o_2 \rangle; RLL \rangle = \\ = \langle \langle l_3, l_3 ID_3 o_3 \rangle; \langle Ch_{0,1}(l_3), l_3 IV_3 o_3 \rangle \rangle,$$

де  $ID_t, IV_t \subset O_t \otimes L_t$ ,  $t=1,2,3$  – відношення опосередкованого безумовного й варіантного підпорядкування, що означають наявність в  $(O_t, E_t)$  гілки між  $r_t$  і  $l_t$ , утвореної  $R_{t1}$  і, відповідно, хоча б однією дугою  $R_{t2}(m,n)$ ;

$RLL$  – відношення, індуковане на  $L^* = \{(l_1, l_2) \parallel l_1 R_3 l_2 \vee l_1 R_4 l_2\}$  відношенням  $RL(o_1, o_2)$ .

Вигляд отриманої об'єктної підмоделі встановлює

*Означення 3.* Об'єктна підмодель – орієнтований зв'язний граф, у якому коренева вершина ( $r$ ) задає ім'я СПП, а підграфи об'єктів-властивостей  $FO$  (3), (4) та інтерфейсних об'єктів  $IO$  (5), (7) поєднані зв'язками перетворюваності  $TR$  (6):

$$OM = \langle r; FO; TR; IO \rangle;$$

$$FO = \langle O_1 \cup O_2, E_1 \cup E_2 \cup E_{12} \rangle; \quad (8)$$

$$TR = \{(o^*, tr(o^*)), o^* = (o_1, o_2) \in O^*\}.$$

У виразі (5) підграф  $FO$  відіграє роль традиційної ФМ [5], надаючи об'єктне подання меж СПП (score), тобто складу і взаємозв'язків всіх припустимих функцій і даних у ПП з СПП. Однак  $FO$  декомпозує ці функції до методів, змінних і елементів даних, необхідних для їх реалізації у ПП методом компонентного програмування, у вигляді графа складених властивостей  $(O^*, E^*)$ . Підграф  $IO$  конструктивізує це подання на підтримку вимог а)–е) для подальшого відображення в інтерфейсній моделі. Спосіб встановлення взаємно-однозначної відповідності між  $O^*$  та  $O_3$  висвітлює

**Лема 1.** Нехай  $R^*_k \subseteq \otimes^2 O^*$ ,  $k = 1, \dots, 5$ ,  $o_i^* = (o_{1i}, o_{2i})$ ,  $o_j^* = (o_{1j}, o_{2j}) \in O^*$  та  $o_i^* R^*_1 o_j^* \Leftrightarrow o_{1i} R_{11} o_{1j} \vee (o_{1i} = o_{1j} \wedge o_{2i} R_{21} o_{2j})$ ;

$$o_i^* R^*_{2(m,n,p,q)} o_j^* \Leftrightarrow (o_{1i} R_{12(m,n)} o_{1j}) \vee \vee (o_{1i} = o_{1u} \wedge o_{2i} R_{22(p,q)} o_{2j}); \quad (9)$$

$$o_i^* R^*_{k} o_j^* \Leftrightarrow o_{1i} R_{1k} o_{1j}, k = 3, 4, 5.$$

Відображення  $tr$  (6) є ізоморфізмом алгебраїчних моделей  $\langle g^*, R^*_i(g^*), i=1, \dots, 5 \rangle$  (9) і  $\langle g_3, R_{3i}(g_3), i=1, \dots, 5 \rangle$  (7) типів  $\langle 2, 2, 2, 2, 2 \rangle$ ,

де  $g \subseteq O^*, g_3 = \{tr(o^*), o^* \in O^*\}$ ;

$R^*_i(g^*), R_{3i}(g_3)$  – обмеження  $R^*_i$  і  $R_{3i}$  на  $g^*$  і  $g_3$  відповідно.

Істинність леми випливає з зіставлення виразів (7), (4), (5) та (9).

*Наслідок.* Для  $g \subseteq O^*$  відображення  $tr$  (6) – ізоморфізм моделей  $\langle g^*, E^*(g^*) \rangle$  та  $\langle g_3, E_3(g_3) \rangle$  типу  $\langle 2 \rangle$ , де  $E_3(g_3) = \cup_{i=1, \dots, 5} R_{3i}(g_3)$ ;  $E^*(g^*) = \cup_{i=1, \dots, 5} R^*_i(g^*)$ .

Лема 1 уможлиблює ізоморфне відображення графа складених властивостей  $(O^*, E^*)$  (9) в інтерфейсну підмодель – другий складник формованої об'єктно-компонентної моделі СПП (2).

**Інтерфейсна модель.** Виходячи з визначення інтерфейсу в об'єктно-компонентному методі [2], для побудови інтерфейсної моделі пропонується відображення  $\varepsilon$ , що відокремлює від інтерфейсних об'єктів  $o_3 \in O_3$  (3), (6) (див. рис. 1) підтримувані ними інтерфейси  $int(o_3) \in IN$  і впорядковує їх згідно з відношеннями між “батьківськими” об'єктами  $o_3$ .

Опис інтерфейсної моделі надає

*Означення 4.* Інтерфейсна модель СПП, об'єктною моделлю якого є  $OM$  (8), – орієнтований зв'язний граф інтерфейсів з інтерфейсних об'єктів  $o_3 \in O_3$ :

$$IM = (IN, EI), EI = \cup_{i=1, \dots, 5} RI_k; \quad (10)$$

$$\varepsilon: (O_3, E_3) \leftrightarrow IM, o_3 = tr(o_1, o_2);$$

$$\varepsilon(o_3) = in = \langle id.in, M(o_1), D(o_2), RL.in \rangle; \quad (11)$$

$$in_i RI_k in_j \Leftrightarrow \varepsilon^{-1}(in_i) R_{3k} \varepsilon^{-1}(in_j),$$

де  $id.in$  – унікальне ім'я інтерфейса;

$RL.in$  – відношення зіставлення змінним з сигнатур  $M(o_1)$  ознак з  $D(o_2)$ .

Зауваження 3. Вираз (11) свідчить, що композиція відображень  $tr$  (8) та  $\varepsilon$

$$\iota = \varepsilon \circ tr: \langle g^*, E^*(g^*) \rangle \leftrightarrow \langle gi, EI \rangle, gi \subseteq IN. \quad (12)$$

є ізоморфізмом алгебраїчних моделей.

Таким чином, інтерфейсна підмодель надає еквівалентне конструктивне перетворення об'єктного подання МХ у систему взаємопов'язаних інтерфейсів  $in \in IN$  (10), що забезпечують взаємодію функцій ПП. Для кінцевої реалізації методів з  $in$  у ПП з СПП пропонується зіставлення  $in$  єдиного програмного компонента повторного використання (КПВ), призначеного для його підтримки в середовищі експлуатації ПП. У цього КПВ єдиним вхідним інтерфейсом має бути  $in$ , а вихідними – інтерфейси, підпорядковані  $in$  у  $IM$ .

Зважаючи на посередницьку роль  $IM$  у формованій моделі СПП, операції над її елементами не запроваджуються.

**Компонентна модель.** Згідно з формальним визначенням КПВ [2], вигляд формованої моделі надає

*Означення 5.* Компонентна підмодель СПП, інтерфейсною підмоделлю якого є  $IM$  (10), – орієнтований зв'язний граф КПВ, кожний з яких реалізує один, і тільки один інтерфейс  $in \in IN$  за допомогою відображення

$$k: (IN, EI) \leftrightarrow CM = (C, EC), EC = \cup_{i=1, \dots, 5} RC_k; \quad (13)$$

$$k(in) = c = \langle id.in, II.c, (S.c, Ch.c), IO.c \rangle;$$

$$II.c = \langle in \setminus id \rangle \cup \langle mch \rangle = \langle M(o_1), D(o_2), mch, RL.in \rangle;$$

$$IO.c = \cup \{ \langle in_j \setminus id \rangle, in RI_1 in_j \} \cup$$

$$\cup \{ \langle in_j \setminus id \rangle, in RI_2(m, n, p, q) in_j \};$$

$$c_i RC_k c_j \Leftrightarrow k^{-1}(in_i) R_{3k} \varepsilon^{-1}(in_j), \quad (14)$$

де  $id.in$  – унікальне ім'я КПВ, успадковане від підтримуваного інтерфейса  $in$  (11);

$II.c$  і  $IO.c$  – вхідний інтерфейс і множина вихідних інтерфейсів  $c$ ;

$mch$  – допоміжний метод, що забезпечує вибір одного з припустимих варіантів  $in_j$  реалізації варіабельного інтерфейсу  $in$  ( $in RI_2(m, n, p, q) in_j$ );

$S.c$  – множина програмних реалізацій вхідного інтерфейсу  $II.c$ .

Зазначимо, що у виразі (14) і далі вилучено “технічні” інтерфейси керування екземплярами КПВ та визначення системних сервісів його взаємодії з компонентним середовищем [2], оскільки їх дослідження істотно виходить за межі роботи.

Зауваження 4. Згідно з виразом (14), відображення  $k$  також є ізоморфізмом ал-

гебраїчних моделей. Елементарним інтерфейсам (листкам  $IM$ )  $il$  відповідають базові КПВ  $cl = \kappa(il) \in LC$ :

$$II.cl = IO.cl = \langle M(l_1), D(l_2), RL.il \rangle, S.lc_1 \cap S.lc_2 = \emptyset,$$

де  $S.lc$  – множина фрагментів предметно-орієнтованого виконуваного коду певною мовою програмування, що разом реалізують методи  $M(l_1)$  на даних  $D(l_2)$  згідно з відношенням  $RL.il$ . На відміну від  $S.lc$ , реалізації складних КПВ з  $S.c$ ,  $c \notin LC$  не є предметно-орієнтованими, а тільки забезпечують (віддалений) виклик методів з  $S.lc$  для  $lc$ , опосередковано підпорядкованих КПВ  $c$  у підмоделі  $CM$  (13), і надаються компонентним середовищем розроблення (CORBA, MS.Net, J2EE тощо).

Оскільки КПВ  $c \in C$  мають єдиний вхідний інтерфейс  $II.c$  (14), виконуються умови їх цілісності та взаємодії підпорядковуючого  $c_i$  й підпорядкованого  $c_j$  [2]:

$$c_i RC_1 c_j \vee c_i RC_2(m, n, p, q) c_j \Leftrightarrow II.c_j \in IO.c_i,$$

і всі реалізації з  $S.c_j$  (14) підтримують  $II.c$ .

Зіставлення виразів для  $II.c$  і  $IO.c$  у (14), підстановка виразу (10) для інтерфейсу  $in$  та врахування зауваження 2 дозволяє позбавитись від елементів інтерфейсної моделі у виразі (14).

**Лема 2.** Вхідний інтерфейс КПВ  $c \in C$  являє собою кортеж інтерфейсів, безумовно та варіантно підпорядкованих  $c$  на довільному рівні в  $CM$ :

$$II.c = \langle \langle II.c_j, cRC_1 c_j \rangle; \langle II.c_j, cRC_2(m, n, p, q) c_j \rangle \rangle = \dots = \langle \langle II.cl, cRC_1 cl \rangle; \langle II.cl, cRC_2(0, 1, 0, 1) cl \rangle; \langle II.cl, cRC_2(0, 1, 0, 0) cl \rangle; \langle II.cl, cRC_2(0, 0, 0, 1) cl \rangle \rangle.$$

Отже, граф вхідних інтерфейсів  $II.c$ , упорядкованих згідно з відношеннями між самими  $c \in C$ , збігається з інтерфейсною моделлю  $IM$  (10).

**Інтеграція моделей.** Здійснення п'ятого й шостого кроків запропонованого підходу на підтримку вимог а)–е) надає

**Означення 6.** Інтегрована об'єктно-компонентна модель СПП – це пара

$$PL = \langle AS; MOD; \rangle, \quad (15)$$

$$MOD = \langle (OM, \iota); (IM, \kappa); CM \rangle,$$

де  $AS$  – умови коректності об'єктного аналізу 1)-4);

$OM, IM, CM$  – відповідно, об'єктна, інтерфейсна та компонентна моделі СПП;

$\iota$  (12) і  $\kappa$  (14) – спеціальні відображення між елементами  $OM$  і  $IM$  та, відповідно,  $IM$  і  $CM$ .

Запропонована інтегрована модель  $MOD$  (15) відповідає сформульованим вимогам конструктивності, надаючи спосіб уніфікованого розроблення всіх ПП з СПП у руслі парадигми компонентного програмування. Саме, вона уможливорює узгоджене об'єктно-компонентне модельне подання основних артефактів  $ar_t$  цього процесу як відповідних підграфів підмоделей  $MOD$ , являючи собою розвиток моделі варіабельності артефактів [11]. Такими артефактами є: потреби споживачів СПП у функціях ПП ( $t=1$ ); вимоги до ПП ( $t=2$ ); проект його компонентної архітектури ( $t=3$ ); специфікація ( $t=4$ ) й програмна реалізація ( $t=5$ ) окремого КПВ у ній; власне ПП ( $t=6$ ). Послідовні перетворення артефактів  $ar_t$  впродовж розроблення ПП відображаються в їх об'єктно-компонентних моделях  $MA_t$  і відстежуються за допомогою ізоморфізмів  $\iota$  (12) і  $\kappa$  (14) з  $MOD$ .

Позицію  $ar_t$  у  $MOD$  встановлює

**Означення 7.** Нехай  $PL$  (15) – модель СПП. Об'єктно-компонентна модель артефакта СПП  $ar_t$  має вигляд

$$MA_t = \left\{ \begin{array}{l} \langle id_1; g_1; g_2 \rangle \subseteq O_1 \otimes O_2, t = 1; \\ \langle id_2; (g_1, g_2); g^*(g_1, g_2) \rangle, \\ g^* \subseteq (O^*, E^*), t = 2; \\ \langle id_2; (g_1, g_2); g^*(g_1, g_2); \iota(g^*) \rangle, \\ \iota(g^*) \subseteq IM, t = 3; \\ \langle id_4; (o_1, o_2); o^*(o_1, o_2); \iota(o^*) \rangle, \\ \iota(o^*) \in \iota(g^*), t = 4; \\ \langle id_5; (o_1, o_2); o^*(o_1, o_2); \iota(o^*); \kappa(\iota(o^*)) \rangle, \\ \kappa(\iota(o^*)) \in \kappa(\iota(g^*)), t = 5; \\ \langle id_2; (g_1, g_2); g^*(g_1, g_2); \\ \iota(g^*); \kappa(\iota(g^*)) \rangle, \kappa(\iota(g^*)) \subseteq CM, t = 6, \end{array} \right. \quad (16)$$

де  $id_t$  – унікальний ідентифікатор моделі  $ar_t$  (збіг ідентифікаторів для  $t=2, 3, 6$  пояснюється тим, що модель вимог  $MA_2$  повні-

стю визначає моделі проекту архітектури  $MA_3$  і самого ПП  $MA_6$ , тоді як модель потреб  $MA_1$  може конкретизуватися різними  $MA_2$ , а моделі специфікації та реалізації КПВ призначені для багаторазового використання у різних ПП).

Модель (16) уніфіковано подає всі різнотипні артефакти СПС: ресурси повторного використання (РПВ); наразі створювані; потенційно необхідні для реалізації ПП з СПП та різноаспектні зв'язки між ними. Разом із “материнською” моделлю СПП *MOD* (15) вона забезпечує також траєкторність довільного артефакта як “догори”, до підтримуваних ним вимог і потреб, так і “донизу”, до КПВ, що нададуть його програмну реалізацію. При цьому точці варіабельності в артефактах довільного типу та її варіанту явно зіставляються їх прообрази “згори” та образи “знизу”. Додатково виникає можливість повноаспектного аналізу “що, якщо” стосовно впливу будь-яких змін – у репозиторії КПВ, усередині підмоделей, у потребах споживачів – як на модель *MOD*, так і на портфелі створюваних та запланованих ПП.

Для окремого ПП модель (16) формалізує спектр змістовних рішень з конкретизації моделі потреб  $MA_1$  у вигляді моделі вимог  $MA_2$ . Вона сприяє зосередженню архітекторів СПП і розробників окремих ПП на креативному прийнятті рішень, знижуючи ризики внесення дефектів під час рутинного “ручного” перетворення вимог у ПП завдяки його автоматизації (звичайно, за наявності КПВ з необхідними моделями  $MA_5$  у репозиторії СПП). З іншого боку, модель специфікації КПВ  $MA_4$  надає підстави для їх паспортизації.

Нарешті, модель (16) є підставою для формального опису компонентно-орієнтованого ПП.

**Означення 8.** Нехай  $PL$  (15) – модель СПП. ПП з СПП, який реалізує вимоги з моделлю  $MA_2 = \langle id_2; (g_1, g_2); g^*(g_1, g_2) \rangle$  – це множина КПВ, що взаємодіють між собою згідно з підграфом  $k(u(g^*))$  компонентної моделі  $CM$  (13), (14) зі складу  $PL$ . Підграф  $k(u(g^*))$  зветься компонентною конфігурацією ПП.

Якщо модель  $MA_2$  включає точку варіантності, ПП зветься варіантним.

**Корисні властивості моделі СПП.** З означень (15), (16) випливає низка тверджень.

**Лема 3.** Відношення безумовного та варіантного підпорядкування на множинах об'єктів, інтерфейсів і КПВ є лінійними порядками.

**Теорема 1.** Нехай  $PL$  (15) – модель СПП. Композиція відображень (12) і (14)  $\tau = \kappa \circ \iota$  є ізоморфізмом між об'єктною та компонентною моделями СПП як алгебраїчними моделями типу  $\langle 2; 2; 2; 2; 2 \rangle$ .

Отже, елементарний об'єкт – листок графа  $(O^*, R^*)$  (3), (8)  $l \in O^*$  реалізується єдиним базовим КПВ (14) – листком у графі  $CM$ , вхідний і вихідний інтерфейси якого збігаються з елементарним інтерфейсом, властивим  $l$ . Для реалізації вузлів графа  $(O^*, R^*)$  запровадимо необхідні операції над КПВ.

**Означення 9.** Результат операції *конфігурування* КПВ  $x, y \in C$  з компонентної моделі (13), (14) – складний КПВ  $x \oplus y \in C$ , названий їх *конфігурацією*, вхідний інтерфейс якого – покоординатне об'єднання вхідних інтерфейсів КПВ  $x$  і  $y$  як кортежів, а множини вихідних інтерфейсів і реалізацій – об'єднання однойменних множин у складі  $x$  і  $y$ .

З означення 9 випливає

**Лема 4.** Кожний КПВ – вузол у графі компонентної моделі  $CM$  (13), (14) – являє собою конфігурацію всіх КПВ, підпорядкованих йому в  $CM$  на довільному рівні, жодний з яких не підпорядкований іншому.

Наслідок. Нехай  $FC$  – множина конфігурацій довільних підмножин базових КПВ у компонентній моделі  $CM$  без повторюваних елементів,  $VC$  – множина вузлів з  $CM$ . Тоді  $|FC| < \infty$ , і  $VC \subseteq FC$ .

**Означення 10.** Результат операції *перетину* КПВ  $x, y \in FC$  – це КПВ  $x \diamond y \in FC$ , вхідний інтерфейс якого – покоординатний перетин (позначений символом  $\cap$ ) вхідних інтерфейсів  $x$  і  $y$  як кортежів, а множини вихідних інтерфейсів і реалізацій – перетини однойменних множин у складі  $x$  і  $y$ :

$$\begin{aligned} II.x \diamond y &= II.x \cap II.y; \\ IO.x \diamond y &= IO.x \cap IO.y; S.x \diamond y = S.x \cap S.y. \end{aligned} \quad (17)$$

**Лема 5.** Для результатів операцій конфігурування та перетину (17) на  $FC$  виконується умова цілісності згідно з ОКМ-методом.

**Теорема 2.** Множина з операціями  $\langle FC; \oplus, \diamond \rangle$  – бульова решітка, а множина  $\langle FC; \oplus, \diamond, \setminus, z, u \rangle$  – бульова алгебра типу  $\langle 2,2,1 \rangle$ , де нулем є КПВ-шаблон  $z = \langle \emptyset; \emptyset; \emptyset \rangle$ , одиницею – “максимальний” КПВ  $u$ , тобто конфігурація всіх базових КПВ з  $CM$ , а операція  $\setminus$  є доповненням у бульовій решітці:

$$\forall vc \in FC \quad vc \oplus (\setminus vc) = u; \quad vc \diamond (\setminus vc) = z.$$

Зауваження 5. Визначимо на  $FC$  відношення підпорядкування  $\sigma$ :

$$\forall (vc_i, vc_j) \in FC \quad \sigma(vc_i, vc_j) \Leftrightarrow (\exists vc_k \in FC / \sigma(vc_i, vc_k) \wedge vc_i = vc_k \oplus vc_j). \quad (18)$$

Означення (18) і означення 1 збігаються на множині  $VC$ .

**Означення 11.** *Варіантна компонентна* ПС (ВПС) з функціями, поданими довільним підграфом  $F$  з об’єктної моделі СПП (15) – реалізація КПВ  $id_5(F)$ , об’єктною моделлю якого є  $F$ , а в компонентній моделі базові КПВ реалізують елементарні функції зі складу  $F$ .

**Теорема 3.** ВПС з функціями, поданими довільним підграфом  $F$  з об’єктної моделі СПП (15), є реалізацією одного, й тільки одного, КПВ  $vc \in FC$ , що є конфігурацією базових КПВ, прототи́пи яких – елементарні функції зі складу  $F$ .

У термінах генеруючого програмування об’єктна модель є поданням простору проблеми, компонентна – простору рішень, а відображення  $\tau = \tau^{\circ k}$  – генеруюча модель ПП із заданими функціями, що обробляють задані дані.

## Механізми використання інтегрованої моделі СПП

**Автоматизоване конструювання продуктів у СПП.** ПП із запитаними функціями створюються як нові або як варіанти наявних (передбачені й довільні).

Нові ПП, що можуть бути довільними варіантами наявних, конструюються за допомогою композиції запроваджених операцій конфігурування та перетину на

підставі рамкових схем конфігурації: базової, видалення підкомпонентів, довільної припустимої заміни.

Варіанти наявних продуктів створюються за допомогою операцій їх варіантно́ї та передбачувано-варіантно́ї заміни на підставі відповідних схем.

Для вибору способу, оптимального за критерієм  $q \in Q = \{\text{час, витрати, якість}\}$ , розв’язуються задачі добору припустимої множини схем відповідного типу та оптимізації на ній.

Композиція операцій довільної заміни визначає трансформаційну генеруючу модель, решти – конфігураційну модель, а їх композиція – гібридну модель [11].

**Обґрунтоване контролювання рівня варіабельності в СПП.** Цей напрям використання запропонованої моделі СПП реалізується за допомогою трьох керованих нею механізмів: моделі обґрунтованого оцінювання варіабельності; системи операцій контролювання рівня варіабельності; структури середовища розроблення СПП та механізму вимог до операцій.

Вимоги до операцій включають [4]: *обґрунтованість* – наявність підстав їх виконання, зрозумілих для всіх учасників розроблення СПП; *узгодженість* – інформаційну наступність і цільову непротиричність операцій (щодо всіх артефактів СПП у всіх функціональних сегментах цільового домену); *трасовність* – відстежуваність зв’язків між точками варіантно́ї й варіантами (на всіх рівнях реалізації варіабельності в ПП та у всіх сегментах домену); *масштабовність* – застосовність операцій для СПП довільного призначення й обсягу; *візуалізованість* точок варіантно́ї, варіантів та зв’язків між ними.

Інтегрована об’єктно-компонентна модель СПП (15) визначає чотири наступні складники інформаційного середовища: репозиторій РПВ з обов’язковим включенням КПВ, що підтримують елементарні функції – листки графа  $OM$ ; репозиторій протоколів повторного використання ресурсів (РПВ) ( $AP$ ); множину робочих продуктів наразі створюваних РПВ і ПП з протоколами їх “локального” повторного використання ( $IP$ ); нову модель експертно-аналітичного оцінювання варіабельності



ПП з СПП ( $EV$ ). Остання формує репозиторій профілів варіабельності СПП – структурованих обґрунтованих оцінок адекватності стану СВС потребам всіх учасників його розроблення у вигляді спеціальних звітів ( $VP$ ). Таким чином,

$$ENV = \langle \langle SV, AV \rangle; \langle RP, AP, IP \rangle; EV; VP \rangle. \quad (19)$$

Вираз (19) висвітлює основні типи невідповідності наявної й запитаної варіабельності:

1) “надлишковість” – наявність артефактів  $id_t \in G_t$  в  $SV$  і РПВ у  $RP$ , використання яких під час розроблення ПП не відбувається або ж не окупується;

2) “неповнота” – відсутність артефактів  $id_t \in G_t$ ,  $t = 1, 2, 3$ , відображених у запитах до споживачів і розробників, що подають істотні потреби у функціях і показниках якості ПП;

3) “клони” – наявність артефактів і РПВ, повністю взаємозамінних при створенні ПП;

4) “хибні обмеження” – наявність артефактів  $id_t \in G_t$  в  $SV$ , які потребують зняття/встановлення статусу точки варіантності;

5) “хибні залежності” – наявність підграфів рівня  $t$ , що потребують перепідпорядкування і/або зміни типу залежності;

Призначенням операцій контролювання варіабельності СПП є:

1) надання підстав для виконання операцій еволюції СПП;

2) формування ретроспективи оцінок профілю варіабельності для її подальшого опрацювання відомими методами ідентифікації структури багатовимірних кількісних і не кількісних даних з метою раннього прогнозування та аналізу чинників впливу.

На підтримку цього призначення пропонується:

– динамічна актуалізація протоколів РПВ і артефактів у момент їх використання;

– експрес-оцінювання рівня варіабельності СПП, поданого структурованим кортежем частотних оцінок термінальних критеріїв з моделі  $EV$ ;

– обґрунтоване оцінювання профілю варіабельності СПП за моделлю  $EV$ ;

– обґрунтоване оцінювання відповідності варіабельності СПС пакету запитів на ПП – на заданий момент і впродовж заданого терміну – за моделлю  $EV$ ;

– відстеження динаміки профілю варіабельності впродовж заданого терміну;

– вчасне виявлення проявів невідповідності варіабельності СПП поточним потребам його розробників і споживачів ПП та обґрунтоване оцінювання рівня її критичності;

– вироблення рекомендацій з усунення виявленої невідповідності;

– включення/вилучення/перепідпорядкування графів властивостей і відповідних їм артефактів;

– долучення/вилучення РПВ і базових КПВ;

– виявлення й опрацювання РПВ-клонів.

Вимоги до  $EV$  з боку наведених операцій обумовлюють перспективність її формування за допомогою узгодженого розвитку ортогональної моделі варіабельності А. Мецгера [1, 4] та моделі COSVAM [7]. Для цього з графів (15) і (16) вилучаються вершини, що подають спільні властивостей всіх ПП з СПП, та їх нащадки за відношеннями перетворюваності  $TR$ . Отриманий тривимірний граф поділяється на підграфи: точок варіантності ( $P$ ), який подає варіантні артефакти процесу розроблення ПП з СПП; варіантів для цих точок ( $V$ ), що відображає припустимі способи реалізації артефактів. Вершини графів  $P$  і  $V$  пов'язуються збереженими зв'язками варіантного підпорядкування і  $TR$ . В “ортогональній” моделі, утвореній  $P$  і  $V$ , виділяються двовимірні графи  $(P_t, V_t)$ ,  $t=1, \dots, 5$ , вершини яких – артефакти типу  $t$  з  $P$  і з  $V$ . Графам  $(P_t, V_t)$  зіставляється третій вимір – структурована оцінка обумовленої ними варіабельності у шкалі відношень або інтервалів ( $ES_t$ ).

Результуюча гібридна модель являє собою п'ятирівневий трьохелементний кортеж, значення елементів якого і є де-

кларованими підставами операцій еволюції СПП:

$$EV = \langle P; V; ES \rangle; P = \langle (P_t, TR_t), t = 1, \dots, 5 \rangle;$$

$$V = \langle (V_t, TR_t), t = 1, \dots, 5 \rangle;$$

$$ES = \left\langle \begin{array}{l} \langle il; \langle \langle vl_t; \langle rl_t, pl_t, ml_t \rangle \rangle, t = 1, \dots, 5 \rangle \rangle; \\ \langle ia; \langle \langle va_t; \langle ra_t, pa_t, pu_t \rangle \rangle, t = 1, \dots, 5 \rangle \rangle; \\ \langle NA_{ii}, t, i = 1, \dots, 5 \rangle \end{array} \right\rangle, \quad (20)$$

де  $il$  та  $ia$  – інтегральні рівні автономної варіабельності в СПП та, відповідно, її адекватності потребам всіх учасників процесу розроблення СПП;

$vl_t$  і  $va_t$  – проміжні рівні автономної варіабельності в СПП та її адекватності, обумовлені артефактами типу  $t$  (тобто “верхні грані паралелепіпедів”  $(P_t, V_t)$ );

$rl_t, pl_t$  і  $ml_t$  – вкладені рівні передбаченої в моделі  $SV$  (1) і, відповідно, реалізованої в  $RP$  варіабельності, а також рівня відповідності між ними;

$ra_t, pa_t, pu_t$  – вкладені рівні адекватності заявленої та реалізованої варіабельності потребам споживачів і розробників ПП з СПП, а також ефективності використання РПВ;

$NA_{ii}$  – множини артефактів типу  $t$ , що обумовлюють виявлену неадекватність варіабельності типу  $i$ .

**Підтримка обґрунтованої автоматизованої еволюції СПП.** Для даного напрямку операції контролювання рівня варіабельності замінено операціями узгодженої актуалізації СПП з метою усунення невідповідностей 1)–5). Еволюція СПП реалізується композицією узгоджених операцій над їх об’єктами з моделі  $SV$  і репозиторію  $RP$ :

– долучення нового графа  $gn_t$  до об’єктної моделі СПП;

– вилучення підграфа  $id_t$  з його нащадками за відношеннями перетворюваності  $TR_t$  і варіантного підпорядкування;

– аналогічне вилучення  $id_t$  з перепідпорядкуванням його окремих нащадків;

– заміна заданого графа  $gn_{it}$  іншим графом  $gn_{ij}$ ;

– підпорядкування графа  $gn_{it}$  іншій вершині  $id_t$  в об’єктній і компонентній моделях СПП;

– актуалізація відношень в об’єктній, інтерфейсній та компонентній моделях СПП (автономна та внаслідок змін у репозиторії і складі підтримуваних функцій і даних ПП).

Еволюція репозиторію  $RP$  включає:

– розроблення й долучення до  $RP$  КПВ, який підтримує нову елементарну функцію;

– доручення до  $RP$  КПВ, реалізація якого є економічно виправданою;

– видалення РПВ та підтримуваних функцій і робочих продуктів;

– видалення РПВ із збереженням окремих функцій і/або робочих продуктів

Перспективними засобами реалізації запропонованої інтегрованої об’єктно-компонентної моделі є спеціальні мови: XVCL [8] та Common Variability Language, стандартизована OMG [13].

## Висновки

1. Запропоновано об’єктно-компонентну конструктивізацію традиційної моделі властивостей Сімейства програмних продуктів на повноаспектну підтримку всіх її функцій у процесі розроблення Сімейства. Складні властивості програмних продуктів з його складу – їх функції та оброблювані ними дані – взаємопов’язані в об’єктній підмоделі, інтерфейси взаємодії функцій подані в інтерфейсній, а відповідні їм компоненти повторного використання – у компонентній підмоделі. Процеси реалізації функцій і даних у продукті подаються ізоморфізмами між моделями.

2. Показано, що об’єктна та компонентна підмоделі із запровадженими базовими операціями об’єднання, перетину та доповнення множин їх елементів є бульовими решітками з одиницею – повним об’єднанням всіх елементів та нулем – порожнім елементом. Доведено, що відображення між ними є ізоморфізмом решіток з унарним відношенням припустимості підмножини складних властивостей (відповідно, КПВ для них) в окремих продуктах.

3. Запропоновано узгоджені об'єктно-компонентні моделі основних артефактів Сімейства – від вимог до коду продукту – як підграфів моделі Сімейства, пов'язаних відображеннями між її підмоделями. Надано композиції базових операцій над моделями для автоматизованого створення продуктів із заданими властивостями та їх рефакторингу для врахування передбачених і непередбачених змін вимог та умов використання.

4. Окреслено механізми застосування запроваджених моделей для обґрунтованого моніторингу рівня варіабельності та еволюції Сімейства.

5. Подальші дослідження включають:

– аналіз перспектив інтеграції запропонованих моделей із численням змінності (Choice Calculus, ) та сервісно-компонентною архітектурою для створення розподілених продуктів, адаптованих до передбачених і непередбачених змін;

– розроблення підходу до оптимізації складу КПВ для заданих функцій та даних;

– створення технології автоматизованого конфігурування таких КПВ у продукт з подальшим розвитком макетного зразка відповідного Конфігуратора;

– апробацію технології в середовищі інструментально-технологічного комплексу ІПС НАН України.

1. *Clements P., Northrop L.* Software Product Lines: Practices and Patterns – Addison Wesley, 2001.
2. *Lee K., Kang K.C., Lee J.* Concepts and Guidelines of Feature Modeling for Product Line Software Engineering // Proc.7th Int. Conf. on Software Reuse: Methods, Techniques, and Tools – London, Springer-Verlag, 2002. – P. 62–77.
3. *Acher M.* et al. Support for Reverse Engineering and Maintaining Feature Models // VaMoS '13, January 23–25, 2013, Pisa, Italy.
4. *She S.* et al. Reverse Engineering Feature Models // ICSE '11, May 21–28, 2011, Waikiki, Honolulu, HI, USA.
5. *Berger T.* et al. Feature-to-Code Mapping in Two Large Product Lines, Software Product

Lines: Going Beyond, vol. 6287: Springer Berlin / Heidelberg – 2010. – P. 498–499.

6. *Czarnecki K.* et al. Cool Features and Tough Decisions: A Comparison of Variability Modeling Approaches", Variability Modelling of Software-intensive Systems (VaMoS), Leipzig, Germany, ACM Press, 01/2012. – P. 173–182.
7. *Deelstra S., Sinnema M., Bosch J.* Variability assessment in software product families. In: Information and Software Technology, vol. 51 Elsevier. – 2009. – P. 195–218.
8. *Zhang H., Jarzabek S.* An XVCL Approach to Handling Variants: A KWIC Product Line Example // APSEC, 2003. – P. 116–125.
9. *Лаврищева Е.М., Гриценко В.Н.* Сборочное программирование. Основы индустрии программных продуктов // 2-изд. доп. и перераб.– Киев: Наук. думка, 2009. – 372 с.
10. *Лаврищева К.М.* Инструментально-технологичний комплекс для розроблення й навчання прийомам виробництва програмних систем // Вісн. НАН України. – 2012. – № 3. – С. 67–79.
11. *Лаврищева К.М.* Теоретичні аспекти керування варіабельністю в сімействах програмних систем // Вісн. КНУ ім. Т. Шевченка. Сер.фіз.-мат. науки. – 2011. – № 1. – С. 45–53.
12. *Слабоспицька О.О.* Підхід до підтримки обґрунтованої еволюції сімейства програмних систем TAAPSD. – 2012. – С. 45–49.
13. *OMG.* ad/09-12-03. Common Variability Language RFP with ab changes, 2009.

Одержано 01.04.2013

### **Про авторів:**

*Лаврищева Катерина Михайлівна,*  
доктор фізико-математичних наук,  
професор, завідувачка відділу  
програмна інженерія,

*Слабоспицька Ольга Олександрівна,*  
кандидат фізико-математичних наук,  
старший науковий співробітник.

### **Місце роботи авторів:**

Інститут програмних систем  
НАН України,  
03187, Київ-187,  
проспект Академіка Глушкова, 40.  
Тел.: (044) 526 4579.  
E-mail: ols.07@mail.ru