

ПІДТРИМКА ПРОЦЕСУ КЕРУВАННЯ ВАРІАБЕЛЬНІСТЮ В СІМЕЙСТВАХ ПРОГРАМНИХ СИСТЕМ

А.Л. Колесник

Інститут програмних систем НАН України,
03187, Київ, проспект Академіка Глушкова, 40, т.: +380 (50) 444 2299

Варіабельність – здатність сімейства ПС, окремої системи або її компоненту до розширення, змінювання, пристосування або конфігурування з метою використання у конкретній предметній області. Варіабельність має забезпечуватися для всіх вимог до ПС, моделі характеристик СПС, архітектури, коду, документації, тестів тощо. Загалом, вона може бути реалізована як у СПС, так і в конкретних ПС. У першому випадку забезпечується варіабельність виробничої лінії, яка підтримує існування сімейства як множини ПС. Вона стосується множини базових елементів будь-якої природи (програмних і не програмних), які містяться у репозиторії виробничої лінії. У другому випадку забезпечується варіабельність складових ПС (включаючи документацію), що обумовлює еволюційність ПС після «відлучення» від СПС. В обох випадках реалізація варіабельності спрямована на забезпечення життєздатності СПС і ПС.

Variability – is ability of Software Product Line (SPL), single system or its component to be extended, modified, adjusted or configured to use in a particular application domain. Variability is provided for all levels of requirements for the software application, feature model, architecture, code, documentation, tests and more. In general, it can be implemented both in SPL and in specific application. In the first case variability is provided for a production line that supports the existence of the family as a set of software application. It applies a set of basic elements of any nature (software and non-software) which are contained in the repository of the SPL. In the second case variability provides variability of software components (including documentation), which causes software evolution after the "separation" from the SPL. In both cases the variability is aimed to ensure the viability of SPL and its applications.

Вступ

Ключовою проблемою інженерії сімейства ПС, як специфічного об'єкта програмної інженерії, є забезпечення належної варіабельності всіх артефактів процесу розроблення СПС та життєздатності ПС – членів СПС.

Варіабельність (синоніми – *варіативність*, *мінливість*) – здатність СПС до розширення, пристосування або конфігурування з метою породження нових ПС відповідно до потреб ПрО. Завдяки варіабельності, підвищується повторна використовність та чутливість СПС до передбачуваних змін ділових процесів ПрО та умов середовища функціонування, оскільки розширюється контекст застосовності ПС – членів СПС. Але водночас зростає складність інтегрованого середовища розроблення (ICP) та, як наслідок, тривалість і вартість складання ПС. Дотримання балансу між очікуваними перевагами та витратами від реалізації тих чи інших варіантів рішень з варіабельності обумовлює потребу оцінювання її рівня та керування нею.

Таким чином, метою керування варіабельністю СПС є максимізація охоплення й утримання сегментів ринку програмної продукції у ПрО через диверсифікацію та високу якість ПС при підвищенні віддачі від інвестицій у ПС і зменшенні витрат та тривалості розроблення ПС з КПВ СПС.

Аналіз теоретичних і практичних напрацювань щодо розроблення ПС в парадигмі СПС (мов, методів і інструментів) показав, що переважна більшість дослідників зосереджує увагу на окремих технологічних і реалізаційних аспектах забезпечення варіабельності, зокрема, механізмах забезпечення варіабельності в ресурсах СПС на окремих рівнях (варіабельності вимог, програмного коду тощо), залишаючи поза увагою проблему обґрунтованого кількісного оцінювання та моніторингу її рівня.

На підтримку вирішення цієї проблеми пропонується вдосконалення методології розроблення СПС теоретичним апаратом узгодженого моделювання варіабельності в структурі та артефактах СПС, а також функціями обґрунтованого керування варіабельністю у пропонованому модельному середовищі. Це модельне середовище дає можливість формулювати і розв'язувати оптимізаційні задачі ресурсно-ефективного планування розроблення СПС [1].

1. Поняття варіабельності СПС

1.1. Мета і задачі дослідження варіабельності СПС. Інженерія СПС принципово відрізняється від інженерії одиничних ПС у таких двох позиціях.

1. Виконання двох взаємодіючих процесів – інженерії ПрО й інженерії застосунків.

Характерна особливість процесу інженерії ПрО – виконання аналізу спільних і відмітних властивостей ПС, які утворюватимуть сімейство, зокрема, спільних і відмітних характеристик ПрО, які охоплюватиме кожна з ПС; спільних і відмітних вимог до кожної ПС, включаючи вимоги до її якості; спільних і відмітних частин еталонної архітектурної платформи, на основі якої будуватиметься архітектура кожної ПС, спільних і відмітних програмних компонентів, тестів тощо.

Властивість СПС, яка обумовлюється її здатністю відрізнити один член СПС від іншого на всіх рівнях його подання, є варіабельністю СПС.

©А.Л. Колесник, 2012

ISSN 1727-4907. Проблеми програмування. 2012. № 2-3. Спеціальний

Реалізація варіабельності СПС має на меті побудову моделей варіабельності СПС і вбудовування механізмів забезпечення варіабельності в артефакти СПС.

Характерна особливість процесу інженерії застосунків – побудова окремих ПС на засадах генерувального програмування, використовуючи спільні компоненти повторного використання – активи СПС (програмні і не програмні КПВ), моделі варіабельності й механізми зв'язування (усунення) варіабельності в відповідних (необхідних для ПС) активах СПС.

2. Необхідність впровадження в процес розроблення СПС процесів інженерії варіабельності, зокрема, керування варіабельністю СПС, з метою підтримання рівня варіабельності СПС, необхідного і достатнього для ефективного продукування ПС з урахуванням:

сфери охоплення ПрО – необхідна і достатня кількість ПС певного класу у функціональному сегменті ПрО, які відповідають потребам множини споживачів;

часу до випуску ПС – часу на створення ПС на основі СПС.

Предметом дослідження питань варіабельності у даному проекті є система моделей і методів підтримки ухвалення рішень при управлінні варіабельністю СПС, орієнтована на підвищення ефективності розроблення СПС, а метою дослідження – створення теоретичного апарату підтримки процесу керування варіабельністю.

Для створення теоретичного апарату підтримки процесу керування варіабельністю, у проекті розв'язувалися такі задачі:

- аналіз методів моделювання і механізмів реалізації варіабельності в СПС;
- побудова моделі процесу керування варіабельністю в СПС з використанням визначених задач;
- розроблення взаємозв'язаних моделей для підтримки вирішення сформульованих задач;
- розроблення системи метрик варіабельності СПС;
- розроблення математичних методів розв'язання формалізованих задач;
- автоматизована підтримка функцій процесу ресурсно-ефективного керування варіабельністю;
- програмна реалізація окремих елементів створеного теоретичного апарату і його апробація в рамках інструментально-технологічного комплексу (ІТК).

Наукова новизна проведених досліджень полягає у розвитку теоретичного апарату інженерії СПС для підвищення ефективності керування розробками СПС через застосування:

- злагоджених моделей: моделі варіабельності; моделі СПС, орієнтованої на керування варіабельністю; моделі процесу керування варіабельністю;
- системи метрик варіабельності;
- математичних постановок задач керування варіабельністю і методів (алгоритмів) їх розв'язання.

Очікуване практичне значення досліджень полягає у тому, що застосування моделей, методів і інструментальних засобів в організаціях розробників СПС забезпечуватиме підвищення рівня задоволеності споживачів ПС за рахунок поліпшення якості наданих ПС при скороченні витрат на їх розроблення.

1.2. Прояви варіабельності. Проявами варіабельності та, водночас, об'єктами керування нею, є такі:

- 1) варіантна характеристика (ВХ) – абстракція, що описує сукупності варіантів і точок варіантності;
- 2) точка варіантності – формальне подання елементарного артефакту СПС певного типу. Точка варіантності може бути зовнішньою або внутрішньою. Зовнішня точка варіантності пов'язується із середовищем, в якому працюватиме ПС (периферією, технічною платформою, операційною системою). Внутрішня точка варіантності – з варіантами функцій ПС або реалізацій. Властивості якості також можуть мати внутрішні і зовнішні точки варіантності, причому точки варіантності функцій можуть спричинювати побіжну варіантність властивостей якості ПС;

3) варіант – елементарний артефакт СПС одного типу з відповідним йому варіантним артефактом;

4) обмеження – обмеження значень варіантів у точках варіантності;

5) залежність – обмеження значень варіантів, якими «закривається» точка варіантності у конкретній ПС.

1.3. Види варіабельності у СПС. Виділяються такі види варіабельності у СПС:

– варіабельність функціональних характеристик – варіанти визначення і реалізації специфікованої характеристики у ПрО або додаткових характеристик;

– варіабельність в архітектурі (структурі) СПС – платформи реалізації ПС;

– дискретна варіабельність встановлюються правила вибору характеристик у точці варіантності, які зводяться до того, вибирати характеристику чи не вибирати (вибір типу «так/ні»);

– неперервна варіабельність прийняття рішень на стадії компіляції, збірки або виконання.

1.4. Рівні варіабельності в структурі СПС.

Варіабельність на рівні СПС визначає, якими компонентами відрізняються окремі ПС.

Варіабельність на рівні ПС визначає архітектуру і компоненти, які можуть бути вибрані для певної ПС.

Варіабельність на рівні компонента визначає його реалізацію, які можуть бути вибрані і включені у ПС.

Варіабельність на рівні підкомпонента пов'язана з видаленням та додаванням частин компонента.

Варіабельність на рівні коду пов'язана з реальною реалізацією відмінностей на всіх рівнях.

У подальших викладах у цьому розділі звіту варіабельність розглядається на рівнях вимог до ПС, моделі характеристик СПС, архітектури, коду, даних, тестів (рис. 1).

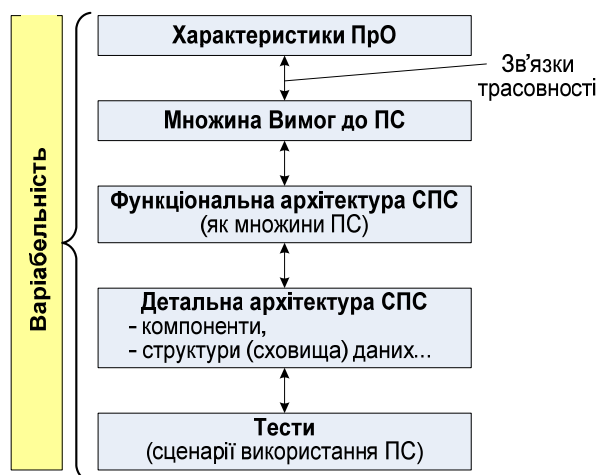


Рис. 1. Рівні розгляду варіабельності в СПС

1.5. Керування конфігурацією та варіабельність СПС. Наразі два аспекти керування конфігурацією – керування версіями однієї ПС у часі (історичний аспект) і керування версіями паралельно створених частин ПС, періодично об'єднаних в одну версію (аспект кооперативної розробки) – отримали належну методичну та інструментальну підтримку (наприклад, ClearCase, CVS). Третій аспект керування конфігурацією – керування одночасно існуючими варіантами окремих робочих продуктів (РП) ПС, які відбивають їх логічну варіантність (наприклад, варіанти коду для різних платформ) – розглядається, як правило, лише з позицій пошуку «обхідних шляхів» для застосування існуючих інструментів керування конфігурацією (наприклад, ведення версій кількох копій ПС для різних платформ, використання умовної компіляції коду тощо).

Проте саме аспект конфігурування множини варіантів є актуальним в інженерії сімейств систем, оскільки варіанти кожного КПВ відповідають контекстам його використання у різних ПС і мають знаходитися під контролем конфігурації одночасно. Та й сама ПС, що є членом СПС, розглядається як його варіант. Таким чином, у контексті процесів життєвого циклу СПС [2] керування варіабельністю доцільно вважати складовою процесу керування конфігурацією.

1.6. Варіабельність СПС та конфігурування ПС. Задачами забезпечення варіабельності СПС є:

- моделювання конфігурованих вимог до СПС під час аналізу ПрО. Варіант конфігурації вимог має відповідати вимогам замовника до конкретної ПС як члена сімейства. Вимоги можуть пред'являтися до функцій ПС та показників якості ПС, елементів архітектури ПС, програмних компонентів ПС та не програмних компонентів ПС;

- реалізація механізмів розширення ядра КПВ у точках варіантності [3]. Точки варіантності на одному рівні абстракції реалізують варіабельність на більш високому рівні абстракції. Прикладами механізмів реалізації варіабельності є паттерни проектування, агрегація, успадкування, параметризація, макроси, умовна компіляція, динамічно приєднані бібліотеки тощо. Реалізація варіабельності здійснюється на рівні архітектури та компонентів СПС. Архітектура СПС визначає всі можливі комбінації реалізацій програмних компонентів. Архітектура ПС є варіантом архітектури СПС, позбавленим точок варіантності.

Задачами конфігурування ПС з КПВ СПС є:

- ототожнювання вимог до ПС з вимогами до СПС у моделі вимог до СПС;
- підтримка операцій: поповнення моделі вимог, трасування вимог до варіантів існуючих КПВ всіх рівнів, породження нових варіантів архітектури та компонентів у точках варіантності для реалізації раніше не встановлених вимог до ПС у ПрО. Ці операції мають виконуватися з урахуванням обмежень ПрО;
- підтримка прийняття ефективних рішень щодо розроблення нових варіантів КПВ через надання інформації для проведення аналізу «витрат/переваг» з урахуванням досвіду, накопиченого у базі знань проектів СПС.

Керування варіабельністю СПС полягає у плануванні розширення КПВ новими варіантами та контролі повноти та несуперечливості варіантів КПВ для потреб СПС, а керування конфігуруванням ПС – у наданні засобів обґрунтованого вибору узгоджених варіантів КПВ при побудові ПС.

Для розв'язання поставлених задач ІСР має бути доповнене не лише інформаційними структурами щодо варіантів КПВ, функціями введення (вилучення) варіантів, зіставлення між собою варіантів одного КПВ («по КПВизонталі») та варіантів КПВ різних рівнів («по вертикалі»), але й моделями та методами, призначеними для підтримки прийняття ефективних рішень розробниками СПС і ПС щодо вибору окремих

варіантів і їх композицій з урахуванням вимог до показників якості ПС, вартості тощо на кожному кроці побудови ПС у парадигмі генерувального програмування.

Адекватне керування конфігурацією СПС потребує узгодженої інтеграції до інженерії Про та інженерії застосунків відповідного процесу. На відміну від процесу керування конфігурацією одиничних ПС, який має чітко визначений профіль, нормативно-методичну та інструментальну підтримку, процес керування конфігурацією та варіабельністю в СПС потребує формалізації та забезпечення всіх складових профілю [2].

1.7. Кроки процесу забезпечення варіабельності СПС. Для того щоб забезпечити бажану варіабельність СПС, треба виконати наступну послідовність кроків [3].

Крок 1. Ідентифікація варіабельності СПС. Найпростіше ідентифікувати варіабельність через моделювання систем в термінах характеристик і визначення характеристик, які матимуть варіанти, тобто варіантних характеристик.

Крок 2. Обмеження варіабельності СПС. Спрямоване на забезпечення достатньої гнучкості СПС з урахуванням поточних потреб і перспектив еволюції у найбільш ефективний за витратами спосіб. На цьому кроці ВХ проходить процедуру введення в СПС і перестає бути неявною. Введена в СПС характеристика має подання в архітектурі і реалізації СПС у формі множини точок варіантності – місць в архітектурі і реалізації КПВ, які разом надають механізми, необхідні для забезпечення варіабельності характеристики. На момент введення у СПС конкретні варіанти характеристики можуть бути відсутніми.

Обмеження варіабельності стосується прийняття рішень з таких питань:

- час введення варіантної характеристики в архітектуру і реалізацію СПС;
- час і спосіб приєднання варіантів до ПС;
- вибір моменту зв'язування варіанта для кожної точки варіантності, тобто моменту, коли точка варіантності конкретизується певним варіантом ВХ (і перестає існувати як така у ПС).

Таким чином, на цьому кроці визначаються всі особливості варіантних характеристик, які впливають на спосіб їхньої реалізації.

Крок 3. Реалізація варіабельності. Залежно від прийнятих обмежень щодо варіабельності може бути вибрана відповідна (найбільш ефективна) технологія реалізації точок варіантності, пов'язаних з певною варіантною характеристикою. На цьому кроці для введеної ВХ визначаються долучені варіанти, для кожного з яких реалізуються програмні сутності, які разом здатні «закрити» всі точки варіантності. Залежно від того, як реалізована точка варіантності, вона буде відкритою для приєднання варіантів на різних стадіях розроблення ПС, або закритою. Тобто, нові варіанти вводяться лише на певних етапах розроблення ПС.

На наступній стадії, приймається рішення стосовно того, який варіант ВХ буде використовуватися, і ПС конкретизується до вибраного варіанта через посилання на конкретні програмні сутності.

Таким чином, ЖЦ варіантної характеристики, як об'єкта СПС, включає наступний перехід її станів (рис. 2): концептуальне подання (неявна ВХ), множина точок варіантності у СПС (введена ВХ), множина варіантів програмних сутностей (ВХ як множина варіантів реалізації), конкретний фрагмент ПС, який реалізує задану характеристику (зв'язана ВХ).

Крок 4. Керування варіабельністю. В результаті виконання кроків 1 – 3 із забезпечення варіабельності у середовищі СПС накопичуються артефакти, створювані впродовж ЖЦ ВХ для всіх розроблюваних ПС.

Щодо них мають застосовуватися супроводжувальні дії, як-от поповнення варіантних характеристик новими варіантами, видалення не використовуваних варіантів тощо, а також, видалення ВХ разом з усіма варіантами при зміні вимог, видалення «слідів» старих ПС у СПС тощо.

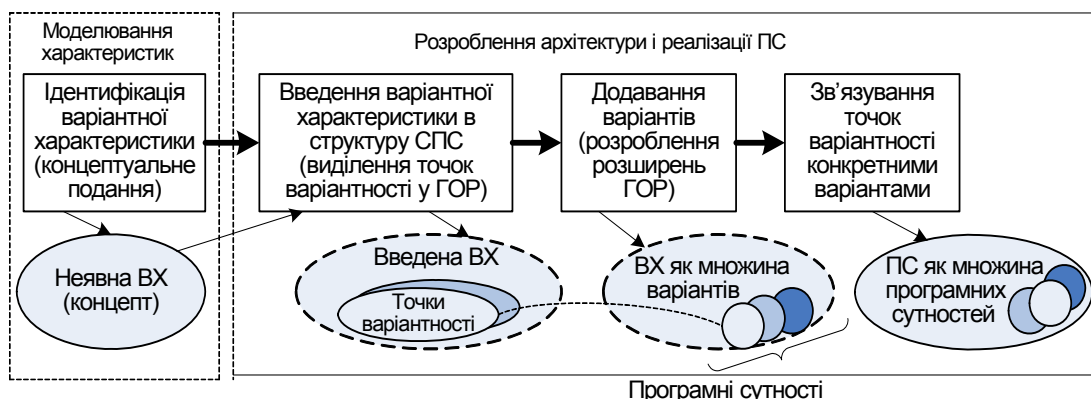


Рис. 2. Процес забезпечення варіабельності

Разом ці дії стосуються регулювання рівня варіабельності СПС і пов'язані з іншими діями з керування варіабельністю – плануванням, обліком і моніторингом варіабельності.

1.4. Проблеми забезпечення варіабельності. Проблеми при забезпеченні варіабельності пов'язані з введенням варіантної характеристики в архітектуру, дорученням варіантів у точках варіантності, зв'язуванням варіанта в точці варіантності.

Введення варіантної характеристики в архітектуру. Варіантні характеристики можуть бути введені на всіх етапах ЖЦ системи, від архітектурного та детального проекту, до реалізації, компіляції і компонування (лінкінгу, від linking) [4]. Одна ВХ може відбиватися на множину різнотипних програмних сутностей, які разом забезпечують бажану функціональність. Вона проявляє себе у СПС множиною точок варіантності на різних рівнях абстракції. Рішення щодо введення ВХ, приймаються з огляду на таке:

- доступність (готовність) технології реалізації, яка задовольняє потребам щодо моментів зв'язування та приєднання варіантів;
- розміри програмних сутностей, яких стосується введення ВХ;
- кількість точок варіантності;
- вартість супроводження введених точок варіантності.

Якщо точка варіантності введена на ранніх стадіях ЖЦ ВХ, вона потрапляє під контроль на всіх наступних етапах розроблення, в іншому випадку – вона контролюється лише протягом нетривалого часу.

Якщо точка варіантності відкрита для приєднання нових варіантів до їх колекції у цій точці, в будь-який момент часу варіанти можуть бути долучені або вилучені. Якщо точка варіантності закрита, колекція варіантів зафіксована і не може змінюватися.

Приєднання варіантів у точках варіантності. Рішення щодо того, коли і як долучати варіанти, приймається з огляду на бізнес стратегію і модель виробництва ПС. Якщо бізнес стратегія передбачає можливість пізнього приєднання варіантів, наприклад, сторонніми постачальниками, це обмежує вибір технологій реалізації точок варіантності, оскільки може виникнути потреба залишати їх (точки) відкритими для приєднання нових варіантів після компіляції або навіть у період виконання ПС.

На те, як і коли долучати варіанти, впливає також процес розроблення ПС і інструменти, використовувані для розроблення [5].

Приєднання варіантів може виконуватися двома шляхами, залежно від того, як реалізовано точку варіантності.

Варіанти можуть бути долучені неявно, і це означає, що не існує подання колекції варіантів у ПС. Колекцією керують за межами системи з використанням, наприклад, простих переліків наявних варіантів. У цьому випадку забезпечення належного варіанта з колекції наявних варіантів покладається на розробників або користувачів. При явному долученні варіантів колекція варіантів присутня у складі початкового коду ПС, і це означає, що в системі достатньо інформації для того, щоб за потреби самостійно знайти необхідний варіант.

Зв'язування варіанта в точці варіантності. Головна мета введення варіантних характеристик – відкладення на подальші стадії розроблення ПС рішення щодо конкретної характеристики, яку треба реалізувати. Конкретизація СПС до необхідного варіанта асоціюється зі зв'язуванням варіанта в точці варіантності.

Розрізняють внутрішнє і зовнішнє зв'язування. При внутрішньому зв'язуванні програмна система включає функціональність, асоційовану із зв'язуваним варіантом. Таке зв'язування відбувається у період виконання. Зовнішнє зв'язування передбачає, що існує особа або інструмент, які фактично виконують зв'язування.

Рішення щодо того, яким має бути зв'язування – внутрішнім або зовнішнім, обумовлюється тим, виконують його розробники ПС чи кінцеві користувачі, а також тим, чи має воно бути прозорим для користувачів.

Як і з приєднанням варіантів, час, коли потрібне зв'язування, обмежує вибір можливих шляхів реалізації точки варіантності. Якщо варіантна характеристика вводиться через багато точок варіантності, можуть виникнути проблеми, оскільки точки варіантності мають зв'язуватися або одночасно (як у випадку зв'язування в період виконання), або синхронізовано (коли, наприклад, точка варіантності, зв'язувана при компіляції, асоціюється з варіантом, який був зв'язаний у точках варіантності на рівні архітектури).

При прийманні рішень щодо часу зв'язування слід керуватися правилом: чим пізніше виконується зв'язування, тим воно витратніше. Відкладання зв'язування з часу побудови архітектури на час компіляції зумовлює необхідність для розробників керування всіма варіантами під час реалізації, а відкладання зв'язування з часу компіляції на час виконання зумовлює необхідність включати функціональність зв'язування і це коштує дорожче в термінах ефективності виконання зв'язування.

2. Модельне середовище керування варіабельністю

2.1. Модель циклу керування Е. Дьомінга. Керування варіабельністю розглядається у контексті циклу керування У. Шухарта – Е. Дьомінга [6] або *циклу PDCA* (від Plan (плануй), Do (виконуй), Check (перевірй), Act (дій, вживай заходи)), який відомий з 1939 року та застосовується для ефективного керування практичною діяльністю на системній основі. Цикл включає такі кроки процесу керування:

- планування – ідентифікація і аналіз проблеми; оцінка можливостей і планування необхідних змін;
- виконання – пошук вирішення проблеми і здійснення запланованих заходів;
- перевірка – оцінка результатів і формулювання висновки відповідно до поставленої задачі;
- дії та заходи – ухвалення рішення на основі зроблених висновків; якщо зміна не вирішує поставлену задачу – коригування планів та повторення циклу.

Пізніше Е. Дьомінг ввів модифікацію циклу PDCA – цикл PDSA ("study" – вивчай), яка й береться за основу при визначенні функцій керування варіабельністю.

2.2. Функції керування варіабельністю та вимоги до їх виконання. Адаптація циклу Е. Дьомінга до сучасних уявлень про ефективну організацію програмної індустрії [7], дозволяє виокремити у процесі розроблення СПС чотири функції кількісного керування варіабельністю:

- 1) планування реалізації варіабельності в артефактах СПС (на рівнях інженерії Про та інженерії застосунків [8]) (F_1);
- 2) реалізація варіабельності в артефактах СПС [3] (F_2);
- 3) системний моніторинг задовільності стану СПС в аспекті варіабельності (F_3);
- 4) актуалізація СПС за результатами моніторингу (F_4).

Для уніфікації керування варіабельністю у процесах інженерії Про та інженерії застосунків, доцільно прийняти такі вимоги до виконання функцій $F_1 - F_4$:

- обґрунтованість – наявність підстав прийняття рішень щодо $F_1 - F_4$ (D_1);
- узгодженість – однаковість способу вироблення й реалізації рішень на всіх рівнях абстракції та на всіх етапах процесу розроблення СПС (D_2);
- масштабовність – незалежність способу вироблення й реалізації від обсягу функціональних можливостей, охоплених СПС (D_3);
- трасовність – можливість відстеження зв'язків між проявами варіабельності на всіх рівнях абстракції та на всіх етапах процесу розроблення СПС (D_4);
- візуалізованість зазначених проявів варіабельності та зв'язків між ними (D_5).

Забезпечення сформульованих вимог $D_1 - D_5$ для функцій $F_1 - F_4$ потребує адекватного модельного середовища їх реалізації.

2.3. Склад модельного середовища керування варіабельністю. На підтримку функцій процесу керування варіабельністю пропонується внести до складу модельного середовища три взаємопов'язані моделі:

- 1) модель СПС для споживача, орієнтовану на варіабельність (VDF);
- 2) модель варіабельності (OVM), яка розвиває ортогональну модель варіабельності;
- 3) модель СПС для розробника (фактично модель артефактів у репозиторії) (VDS).

Модель СПС для споживача, орієнтована на варіабельність (VDF), надає інтенціональне подання СПС. Модель СПС для розробника (VDS) пропонує екстенціональне подання моделі VDF, відповідне інтенціональному.

Саме модель VDS являє собою множину формалізованих подань тих КПВ, які вже створені, наразі створюються або можуть бути створені на підставі поточної моделі VDF, з підмножиною програмних продуктів для замовника(ів):

$$VDS = \{G_i, i \geq 1\}, G_i = (CAD_i, b_i, t_i, s_i, fu_i, cu_i, r_i, id_i), VDP \subseteq VDS, \quad (1)$$

де CAD_i – опис i -го КПВ G_i в термінах властивих йому характеристик (features);

$b_i \in B$ – підстава виокремлення G_i як КПВ СПС;

$t_i \in T$ – тип G_i ;

$s_i \in S$ – стан КПВ G_i ; $S = (SP \cup SR) \times SE$;

fu_i – частота використання G_i в ролі готового продукту типу t_i для замовника;

cu_i – частота використання G_i як КПВ при розробленні ПС для замовника;

$r_i(t_i, s_i) \in [0;1]$ – рейтинг G_i ;

id_i – реєстраційні реквізити G_i у складі персоналій і ролей суб'єктів його формування, унікального ідентифікатора та дати реєстрації у VDS (невизначені при $b_i =$ "Потенційна реалізованість");

B, T і S – поповнювані множини підстав виокремлення КПВ, типів КПВ та їх станів;

SP , SE і SR – підмножини припустимих станів КПВ, які, відповідно, можуть бути створені, наразі створюються та вже створені (останні утворюють відповідний репозиторій КПВ СПС).

Пропонується такий первинний склад запроваджених множин з (1):

$B = \{\text{“Аналіз ПрО”}, \text{“Аналіз результатів процесу розроблення”}, \text{“Вимоги Замовника”}, \text{“Потенційна реалізованість”}\};$

$T = \{\text{“Програмний компонент”}, \text{“Документ”}, \text{“БД”}, \text{“файл певної структури”}\};$

$SP = \{\text{“Передбачений до розроблення за мінімуму достатніх ресурсів”}, \text{“Очікує рішення щодо розроблення”}, \text{“Не передбачений до розроблення”}\};$

$SE = \{\text{“Описаний вимогами”}, \text{“Спроекований”}, \text{“Закодований”}, \text{“Тестований”}, \text{“На дослідній експлуатації”}\}$ – відображає стадії процесу розроблення продуктів;

$SR = \{\text{“Створений на підтримку розроблення СПС”}, \text{“Створений на підтримку бізнес-процесів ПрО”}, \text{“Створений для Замовника”}\}.$

Зазначимо, що $\forall G_i | s_i \in SP \ f u_i = c u_i \equiv 0$.

Запропоновані моделі відображають поточний стан процесу розроблення СПС і підлягають актуалізації:

– при виявленні нових потреб бізнес-процесів у ПрО;

– при висуванні суб'єктами бізнес-процесів у функціонально-цільових сегментах ПрО нових вимог до програмних продуктів (у складі реактивного розроблення цих продуктів як елементів СПС).

У першому випадку за схемою $(VDF \xrightarrow{M} OVM \xrightarrow{P} VDS) \wedge (VDF \xrightarrow{P} VDS)$, реалізуються прямі зв'язки між моделями, а саме зв'язки типу Моделювання (\xrightarrow{M}) і Реалізація (\xrightarrow{P}), а в другому – обернений зв'язок типу Актуалізація (\xrightarrow{A}) за схемою $(VDS \xrightarrow{A} OVM \xrightarrow{A} VDF) \wedge (VDS \xrightarrow{A} VDF)$.

Трьом запровадженим зв'язкам надається такий зміст:

– моделювання полягає в узгодженому вилученні з моделі СПС для споживача елементів, обов'язкових для всіх членів СПС, для подання у моделі OVM тільки точок варіабельності та варіабельних характеристик;

– реалізація передбачає застосування до моделей VDF і OVM операцій побудови узгоджених із ними формальних подань КПВ;

– актуалізація охоплює застосування до моделі VDS операції несуперечливого відображення в моделі VDF характеристик нових продуктів, відповідних вимогам замовників і/або потребам бізнес-процесів ПрО.

Використання моделей VDF , OVM і VDS при розв'язанні визначених задач керування варіабельністю потребує розроблення для кожної з них трьох взаємно узгоджених подань:

– формального, придатного для створення й опису методів розв'язання зазначених задач, а також зовнішнього і внутрішнього подання;

– зовнішнього, що забезпечуватиме ефективне оперування цими моделями (у середовищі інтерфейсу відповідних інструментальних засобів) з боку виконавців основних ролей у процесі розроблення СПС, які використовують ці моделі в межах своїх сфер відповідальності;

– внутрішнього – у середовищі розроблення СПС, використовуваному інструментальними засобами підтримки процесу керування варіабельністю.

2.4. Модель структури СПС, орієнтованої на керування варіабельністю. Модель структури СПС для споживача, орієнтованої на керування варіабельністю, має відповідати таким вимогам:

– бути придатною для подання артефактів у різних взаємопов'язаних функціональних сегментах ПрО;

– бути придатною для подання артефактів (КПВ) СПС *відповідно до сфери відповідальності учасників розроблення СПС* (як мінімум, замовників, аналітиків у ПрО, бізнес-аналітиків, архітекторів, програмістів);

– включати *нотації*, прийнятні для моделювання артефактів (КПВ) СПС розробниками СПС відповідно до типу артефактів;

– бути прийнятною для *ідентифікації спільних* для всіх ПС і *відмінних* (притаманних окремим ПС) характеристик артефактів, та *відображення залежностей* між варіабельними артефактами.

Модель СПС, яка має декілька взаємопов'язаних рівнів подання артефактів у відповідних нотаціях, є гібридною і відповідає поставленим вимогам, схематично показано на рис. 3.

Перший рівень максимально описує характеристики ПрО і є найближчим і найбільш зрозумілим клієнту (замовнику ПС), оскільки подається в термінах ПрО. Береться до роботи в момент визначення функціональності майбутньої ПС або в разі необхідності розширення існуючої функціональності.

Другий рівень описує вимоги до нової функціональності ПС з боку бізнес аналітиків, які аналізують ринок і побажання замовників та вносять відповідні зміни до першого рівня (рівня характеристик ПрО).

На третьому рівні працюють архітектори СПС, які аналізують вимоги до ПС та визначають засоби реалізації ПС.

На четвертому рівні працюють програмісти, тестувальники та інші розробники СПС, разом з архітекторами. Вони створюють нові та підтримують актуальність існуючих компонентів СПС у відповідності з вимогами до ПС та архітектурою СПС.

На п'ятому рівні працюють адміністратори БД, які підтримують актуальність структур БД і їх відповідність потребам ПС.

Подання моделі структури СПС доцільно розглядати в трьох аспектах – зовнішнє візуальне подання; формальне подання в нотаціях, запропонованих обраними методами моделювання; та внутрішнє подання в середовищі розроблення СПС.

2.5. Формальна модель структури СПС. Елементарні артефакти СПС поділяються на чотири типи (t):

- 1) характеристика ПС, передбачена до реалізації ($t=1$);
- 2) компонент архітектури ПС ($t=2$);
- 3) програмний артефакт – (компонент, аспект тощо) ($t=3$);
- 4) таблиця бази даних ($t=4$).

Артефакти типу $t=1$ поділяються на дві групи: враховані потреби ділових процесів ПрО та вимоги до ПС з боку аналітиків СПС.

Ієрархію рівнів варіабельності артефактів СПС, відповідних типам $t = 1, \dots, 4$, встановлює

Означення 1. Модель структури СПС – це кортеж

$$SV = \langle \langle CF; \langle DR, TC \rangle; \langle AR, TD \rangle; \langle CM, FR, TS, TA \rangle; \langle ER, TF \rangle \rangle; Constr; Dep \rangle, \quad (2)$$

де $CF = \langle SF, LF \rangle$ – діаграма характеристик, яка подає множину сталих і змінних властивостей ПС (SF), пропонованих їх потенційним споживачам у ПрО, та обмеження ($Constr$) і залежності (Dep) між ними за допомогою зв'язків обов'язкового і варіантного підпорядкування (LF);

$DR = \langle SF \cup SR, LF \cup LR \rangle$ – аналогічна діаграма, де характеристики $f \in SF$ деталізовано і/або доповнено характеристиками $r \in SR$, що подають сталі та змінні "технічні" вимоги до ПС з боку аналітиків СПС, обмеження й залежності яких подаються зв'язками LR ;

$TC = \{(r, f), r \in SR, f \in SF\}$ – двосторонні зв'язки трасовності між "технічними" вимогами до ПС та їх властивостями для споживача, які конкретизуються вимогами;

$AR = \langle AC, LC \rangle$ – еталонна архітектура СПС, тобто опис взаємозв'язків (LC) між формальними поданнями програмних компонентів (AC) для реалізації характеристик $f \in DR$ у певній нотації для програмної архітектури (UML, xADL тощо);

$TD = \{(ac, f), ac \in AC, f \in SF \cup SR\}$ – двосторонні зв'язки трасовності між поданнями компонентів $ac \in AC$ для реалізації характеристик $f \in DR$ та цими характеристиками;

CM – формальні подання базових компонентів для реалізації архітектури при складанні за допомогою елементів компонентного каркаса (FR) [1];

$TS = \{(ts, cm), cm \in CM\}$ – формальні описи тестів (ts) для компонентів з поданнями cm ;

$TA = \{(cm, ac), cm \in CM, ac \in AC\}$ – двосторонні зв'язки трасовності між програмними компонентами та компонентами архітектури AR ;

$TL = \{(fr, lc), fr \in FR, lc \in LC\}$ – двосторонні зв'язки трасовності між елементами каркаса для складання компонентів, та взаємозв'язками компонентів архітектури;

ER і TF – відповідно, ER-модель БД для оброблення ПС з СПС, та двосторонні зв'язки трасовності її елементів з компонентами $cm \in CM$.

Кожна з наведених рис. 3 п'яти "горизонтальних" площин відображає подання потреб потенційних споживачів, формоване у відповідному під процесі розроблення СПС в артефактах відповідного типу – від характеристик потреб ПрО до підмножин полів таблиць БД.

Варіантність цих артефактів має два аспекти: розбіжність варіантів їх втілення в ПС для певного функціонального сегменту ПрО і в ПС для різних функціональних сегментів. Два незалежні напрями приєднання вершин до формальних подань варіантності – графів $DR, AR, (FR, CM), ER$, відповідні аспектам варіантності, – утворюють виміри "горизонтальних" площин на рис. 3).

Третім виміром моделі (2) є трасовність артефактів суміжних рівнів, реалізована зв'язками з множин TD, TA, TF і подана на рис. 3 вертикальними стрілками.

Зауваження 1. У моделі структури СПС (2) зв'язки зі складу множин TF, TA, TD реалізують неперервну трасовність довільного артефакту СПС типу $t > 1$ знизу догори, зіставляючи йому хоча б один артефакт-відповідник кожного з типів t^* ($1 \leq t^* < t$), для реалізації (безпосередньої або опосередкованої) якого призначено трасований артефакт.

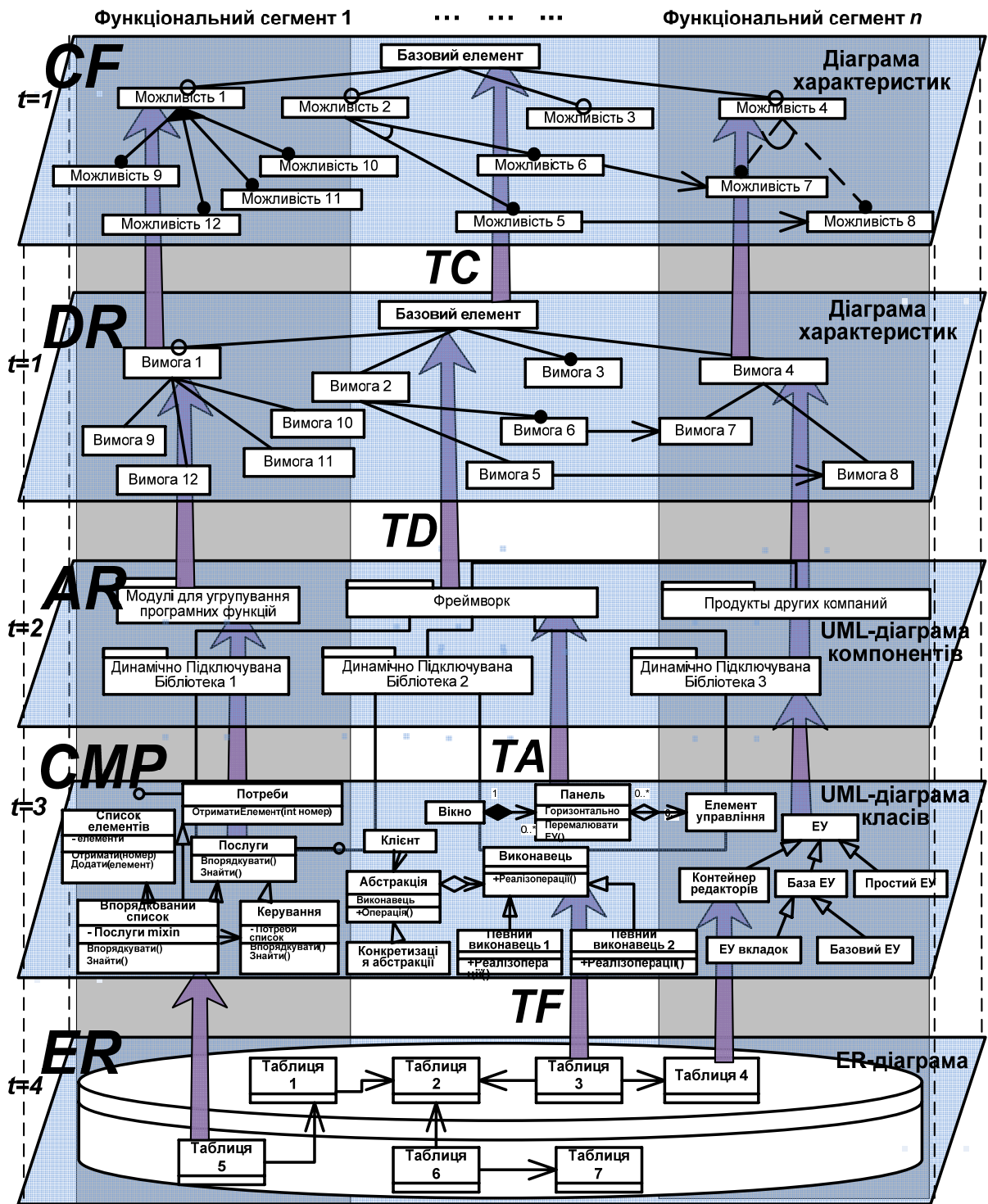


Рис. 3. Гібридна модель СПС

Зуваження 2. У моделі структури СПС (2) може існувати артефакт СПС довільного типу $t = 1, 2, 3$, для якого не визначено жодного артефакту-відповідника типу t_* ($t < t_* \leq 4$) за зв'язками зі складу множин TD , TA , TF , призначеного для його реалізації (безпосередньої або опосередкованої).

Такий артефакт зветься надалі нетрасовним (якщо $t_* = t + 1$) або нетрасовним на рівні t_* .

Із сформульованих зауважень випливає неможливість появи на будь-якому рівні моделі (2) вершин, не підпорядкованих певній характеристиці $f \in SF \cup SR$, а також дублювання вершин (за необхідності, долучаються нові зв'язки трасовності). Це означає неможливість дублювання артефактів у процесі розроблення СПС та створення "зайвих" артефактів, не призначених для реалізації в ПС певних характеристик з меж СПС.

Висновки

Головним результатом виконання проекту в контексті варіабельності СПС є вдосконалення методології розроблення СПС теоретичним апаратом узгодженого моделювання варіабельності в структурі та артефактах СПС та обґрунтованого керування варіабельністю у пропонованому модельному середовищі, яке дає можливість формулювати і розв'язувати оптимізаційні задачі ресурсно-ефективного розроблення СПС.

Отримано такі основні результати:

1. Визначено основні функції, задачі та структуру процесу керування варіабельністю.
2. Запропоновано модельне середовище ресурсно-ефективного керування варіабельністю, яке включає узгоджені моделі варіабельності в структурі та артефактах СПС; тривимірну модель варіабельності, яка з'являє вимірам її ортогональної моделі третій вимір – вкладену оціночну модель, яка надає можливість накопичення оцінок варіабельності та визначення ступеню її відповідності потребам Про.
3. Запропоновано низку механізмів забезпечення варіабельності програмних активів (готових ресурсів) СПС та сформульовано підхід до їх вибору у контексті парадигм програмування, операційних систем та мов програмування.

1. Гриценко В.Н., Лаврищева Е.М. Методы и средства компонентного программирования // Кибернетика и системный анализ.– 2003.– № 1.– С. 39 – 55.
2. Лаврищева К.М., Коваль Г.І., Коротун Т.М. Підходи інженерії якості сімейств програмних систем // Проблеми програмування (Спецвипуск конференції УкрПРОГ-2008). – 2008. – № 2-3. – С. 219 – 228.
3. Колесник А.Л. Механізми забезпечення варіабельності в сімействах програмних систем // Проблеми програмування. – 2010. – № 1. – С. 35 – 44.
4. Svanberg M., van Gurp J. A. Taxonomy of Variability Realization Techniques // Practice and Experience . – 2006. – N 8. – P. 705–754.
5. Лаврищева К.М., Коваль Г.І., Слабоспицька О.О., Колесник А.Л. Особливості процесів керування при створенні сімейств програмних систем // Проблеми програмування. – 2009. – № 3. – С. 40 – 49.
6. Walton M. The Deming Management method // New York : Dodd, Mead, 1986. – 262 p.
7. Андон П.І., Лаврищева К.М. Розвиток фабрик програм в інформаційному світі // Вісник НАН України. – 2010. – № 10. – С. 15 – 41.
8. Trinidad P., Benavides D., Ruiz-Cort'es A. Improving Decision Making in Software Product Lines Product Plan Management // in: Proc. of the ADIS 2004 Workshop on Decision Support in Software Engineering, CEUR Workshop Proceedings. – 2004. – Vol. 120. – P. 51 – 58.