УДК 681.03

А.О. Аронов, А.І. Дзюбенко

ПІДХІД ДО СТВОРЕННЯ СТУДЕНТСЬКОЇ ФАБРИКИ ПРОГРАМ

Розглядаються перші кроки з розроблення проекту фабрики програм, що зможе виконувати роль інформаційно-навчального інтернет проекту в галузі програмної інженерії та стати інструментом накопичення студентських наукових робіт у галузі кібернетики.

Вступ

Широке поширення видів і типів нової комп'ютерної техніки ставить перед науковими і програмістськими фахівцями важливі завдання з ефективного її використання при створенні програмних і прикладних систем масового застосування.

Водночас, за останні десятки років в інформаційному світі накопичено багато різного роду програм, які можуть використовуватися іншими, як готові продукти при вирішенні деяких нових завдань.

Враховуючи це, склався новий напрям у програмуванні (reusability), який орієнтований на створення нових програмних систем (ПС) з застосуванням готових програмних артефактів і компонентів повторного використання (КПВ) [1–5].

Програмний артефакт – це реальна порція інформації з програмування, яка може створюватися, змінюватися і використовуватися у діяльності, пов’язаній з виготовленням програмних продуктів (ПП) різного призначення. Артефакти є формалізованим результатом діяльності розробників програм, які відображають певну функціональність. Артефактами можуть бути:

– модель предметної області (ПрО) зі своїми термінами, поняттями та лексикою;

– готові функціональні КПВ або окремі частини (фрагменти) систем;

– проміжні продукти процесу розроблення (вимоги, завдання, моделі та ін.);

– специфікації (ресурсу, інтерфейсу, процесу і т. п.) окремих елементів, діаграм, паспортних даних і т. п.

Артефакти незалежні від платформ комп’ютерів, мають опис інтерфейсів і різних параметрів для взаємодії з іншими програмними ресурсами.

Кожний розроблений компонент або КПВ має такі властивості [1, 3, 5]:

– зв'язок через інтерфейси із зовнішніми компонентами в процесі розроблення ПС;

– інкапсульованість як «чорна скринька» без можливості втручання у вихідний код;

– успадкованість інтерфейсів, їхня зміна і налаштування на застосування;

– здібність до повторного використання його вихідного коду.

Із загальної точки зору компонент визначається по-різному залежно від середовища його створення або функціонування. Наведемо деякі існуючі визначення для КПВ.

Визначення 1. КПВ – це деяка функція з певними атрибутами, що забезпечують взаємодію з середовищем функціонування і його поведінку.

Визначення 2. Готовий КПВ – це сукупність методів із визначеною сигнатурою і типами даних, які передаються і повертаються після виконання відповідного викликуваного методу.

Компонент типу КПВ розглядається нами як самостійний програмний елемент, який задовольняє певним функціональним і архітектурним вимогам, інтерфейсу взаємодії у заданому середовищі. КПВ і його інтерфейс специфікуються деякою мовою програмування (МП) або мовою опису інтерфейсу IDL, що допома-

© А.О. Аронов, А.І. Дзюбенко, 2011

гає майбутньому користувачу використовувати і об'єднувати його з іншими компонентами в деякій новій системі, що створюється.

У зв'язку з появою компонентів зі спеціальним сервісним призначенням, що можуть виконувати певну сукупність прикладних сервісів, поширена роль інтерфейсів, які вказують на функції і порядок операцій звернення до іншого компонента. Виходячи з цього, модель компонента, зокрема і сервісу, має такий вигляд:

$$\text{КПВ} = (T, I, F, R, S),$$

де T – тип компонента; I – множина інтерфейсів; F – функціональність компонента; R – реалізація чи прихована частина – програмний код; S – сервіс, що забезпечує взаємодію компонента з середовищем щодо оброблення або розгортки.

Ці параметри моделі задаються у специфікаторі, у випадку, коли компонент записується у деяке загальне сховище.

Усі програмні КПВ і артефакти, якими можуть користуватися інші спеціалісти як готовими, мають зберігатися у загальних сховищах (бібліотеках, репозиторіях) для подальшого пошуку іншими фахівцями і застосування у нових ПС.

Тобто повторне використання готових ресурсів стає капіталомістким видом діяльності у галузі програмної інженерії, особливо при розробці складних програм із готових КПВ за умов сучасних фабрик програм.

У наш час надзвичайно важко створити деяку необхідну та корисну програмну систему з готових компонентів і різних наукових артефактів на рівні навчання в університеті і створення їх власноруч, а також апробувати ті КПВ, що розроблені іншими студентами раніш. Їх використання дозволяє підвищити продуктивність праці інших студентів з програмування будь-якого рівня, а також полегшити групове програмування. Ідея КПВ застосовується в багатьох системах-фабриках автоматизації ПС.

Концепцію фабрик програм уперше сформулював академік В.М. Глушков на науковому семінарі 5 березня 1975 р. в Інституті кібернетики [1]. Суть нової для того часу парадигми Глушкова – прискорити перехід від мистецтва програмування до промислових методів виробництва ПП для розв'язання різних господарських задач народного господарства СРСР. Основними аргументами академіка Глушкова на користь переходу до промислового програмування були:

– великий арсенал обчислювальної техніки, вітчизняні математичні машини, виготовлені самостійно в Україні;

– готові програми у мовах програмування 4GL (Алгол, ПЛ-1, Кобол, Фортран, Модула-2 та ін.) й алгоритми, накопичені в республіканських фондах алгоритмів і програм для повторного використання у нових розробках за конвеєрною зборкою;

– побудовано важливі автоматизовані системи, деякі АСУ керування підприємствами господарського та військово-промислового комплексу, АСУ ТП тощо.

У КНУ імені Тараса Шевченка професор Лавріщева К.М. читає курс лекцій «Технологія програмування ІС» для студентів 4-го курсу на кафедрі інформаційних систем.

Студенти виконували лабораторні і практичні заняття з наукових артефактів кібернетики, використовуючи різні процеси життєвого циклу технології побудови програм мовою C# у середовищі VS.NET.

Зважаючи на це, протягом лекцій обґрунтовувався шлях індустріального виробництва програм на прикладі діючих в даний час фабрик програм, які пропонують могутні закордонні фірми (Microsoft, IBM, Sun, COM, CORBA тощо) для випуску програмних продуктів на основі загальносистемних властивостей операційних середовищ [3–5]. За допомогою загальних засобів побудовано більше сотні різних типів ліній виробництва програм, що підтримують технологію розроблення у середовищах зазначених фірм з використанням сучасних МП. В університетах США сумісно з цими фірмами викону-

ються пілотні проекти з студентами, в тому числі зі створення ліній продуктів (Product lines) різного призначення.

Виходячи з цього К.М. Лавріщевою був запропонований експериментальний проект зі створення студентської фабрики наукових артефактів і програм в КНУ імені Тараса Шевченка на факультеті кібернетики. Спочатку на фабрики будуть розміщуватися наукові виконані лабораторні й дипломні роботи за допомогою лінії продуктів, призначених для розробки програм і артефактів.

Лінія продуктів – це збірковий конвеєр з готових ресурсів для створення програмних продуктів спеціального призначення на задоволення потреб ринку.

Кожна лінія має включати наступні властивості та засоби:

– умови й обмеження на застосування різних ресурсів на лінії;

– зразки, каркаси, готові компоненти, КПВ, артефакти тощо;

– стратегії та методи програмування;

– засоби та інструменти виробництва проміжного результату продукту на різних процесах життєвого циклу (ЖЦ) лінії;

– сервіс контролю планів робіт та виявлення різного роду ризиків (програмних, людних тощо);

– механізми вимірювання та оцінки показників якості створюваного продукту;

– засоби сертифікації ПП.

У студентському проекті фабрики запропоновані три загальні лінії виробництва програмного продукту наступного змісту: ***Перша лінія*** – це загальна схема процесів ЖЦ для побудови деякого виду програмного артефакту шляхом (рис. 1):

– вивчення завдань предметної області, виявлення серед них загальних властивостей і функцій та методів породження з них програмних артефактів і елементів;

– специфікація цих елементів мовами програмування і паспортних даних їх інтерфейсів;

– зберігання артефакту або програми з їх специфікатором у бібліотеку, або репозиторій;

– тестування програм, їх інсталяція і верифікація.

Реалізація процесів такої лінії закінчується визначенням сформованих артефактів у класі задач ПрО, специфікацією паспортних даних та розміщенням їх описів у репозиторії для подальшого застосування. Такого типу лінії розробляється на студентській фабриці у вигляді ЖЦ створення програм у середовищі Visual Studio .NET (рис. 1).

Друга лінія – це механізми підбору готових програм із репозиторію за їхніми функціями і відповідними критеріями (рис. 2) з метою оцінки можливості їх застосування при розробці нового програмного артефакту або ПС.

Третя лінія – це збіркова лінія, що забезпечує конструювання нових програмних продуктів методом збірки з застосуванням знов розроблених і підібраних програм і артефактів у репозитарії (рис. 2).

Ця лінія може давати прибуток за рахунок заощадження трудовитрат від застосування готових артефактів, КПВ та

Студентські артефакти є науковими, тобто їх продуктом можуть бути деякі методи або наукові алгоритми з математики, фізики, біології тощо, що описані засобами однієї з МП [1–7].

Рис. 1

Типи фабрик програм

На даний час діють типи фабрик для виробництва програм наступного призначення:

– системні програми, такі як ОС, інструменти редагування, трансляції, композиції тощо;

– сімейства систем (АСУ, АСУТ), які виробляються за новими мовами опису специфіки розділів доменів у DSL, що відрізняються функціональністю та засобами їх підримки;

– бізнесові та комерційні програми, що реалізуються як незалежно розміщувані частини коду з публічним інтерфейсом і можуть бути скомбіновані з іншими для отримання більш складної функціональності (наприклад, у процесах керування підприємством);

– бортові системи, що використовуються як вбудовані програми спеціального призначення у космічних кораблях, літаках, сучасному колайдері тощо;

– застарілі системи (Legacy System), що повторно можуть використовуватися після реінженерії їх складових або усієї системи до конкретних умов застосування;

– студентські програми, що створюються за загальним ЖЦ, починаючи з опису алгоритмів наукових задач;

– утилізаційні і перебудовані програми під нове середовище функціонування або призначення;

– окремі інструменти (космічні, медичні прибори, мобільні телефони, комп'ютери тощо), що виконують спеціальні функції.

Завдання і структура сайта фабрики програм

Нами були зроблені перші кроки з розроблення проекту фабрики програм, що зможе виконувати роль інформаційно-навчального Інтернет-проекту в галузі програмної інженерії та основою накопичення студентських наукових робіт у галузі кібернетики.

Головне завдання: розробити систему обміну сертифікованими програмними продуктами та науковими артефактами студентів КНУ імені Тараса Шевченка для підвищення кваліфікації студентів, а також підвищення якості, надійності створюваних ними систем та продуктів.

Кожна програмна система за час свого існування проходить визначену послідовність періодів від ідеї створення до реалізації, використання та завершення. Таку послідовність називають життєвим циклом.

З точки зору виробництва програм модель ЖЦ – схема виконання робіт і задач у рамках процесів, які забезпечують проектування, розробку, експлуатацію і супровід програмного продукту. Ця схема відповідає стандарту ЖЦ – ISO/IEC 12207 і зображує еволюцію ПС, починаючи від формулювання вимог і закінчуючи припиненням або завершенням користування нею.

На сайті фабрики представлені моделі ЖЦ, лінії зборки програмних продуктів, лінія виробництва окремих програм. В MS.NET поданий приклад студентської програми, наведені необхідні вимоги для сертифікації програмного продукту для його розміщення у репозиторії (бібліотечному фонді) фабрики.

Рис. 2

Основний напрям діяльності сайта – ознайомлення студентів, розробників із засобами та методами створення програм та програмних систем, представлення студентських програмних продуктів у репозиторії, запропонувати літературу для вивчення даної області діяльності та ознайомитись з особливостями. А головне, це залучити до сайта нових студентів, які будуть специфікувати свої артефакти і накопичувати їх у репозитарії для ознайомлення іншими студентами.

Приклади фабрик програм

Відповідно до аналізу робіт [1, 6, 7], в сучасному інформаційному світі діють наступні фабрики програм.

Система АПРОП (Інститут кібернетики), що працювала у середовищі ОС ЄС і об'єднувала різномовні модулі через інтерфейсні посередники за методом зборки.

Система Sun Microsystems (IBM) зі зборкою різномовних програм у всіх сучасних мовах програмування (наприклад, Ruby, Script) та побудова нових ліній виробництва складних ПП з використанням модельного підходу (SOA), Web-сервісів тощо.

OMA-архітектура або система CORBA (OMG), яка забезпечує взаємодію клієнта і сервера через модулі-посередники Stub (для клієнта), skeleton і Dill (для сервера), що передають зовнішні дані, включаючи інтерфейс, брокеру для виконання їх клієнтом або сервером.

Фабрика «ручної» зборки різномовних програм Інга Бейя з використанням інтерфейсних посередників та конфігураційних файлів для забезпечення взаємодії різномовних програм у різних середовищах (VC++, VBasic, Matlab, Java, Visual Works Smalltalk тощо).

Фабрики бізнесних програм Дж. Гринфільда, яка підтримує виробництво програм, сценарії дій яких задаються use case і діаграмами мови UML.

Сучасне колективне мережне середовище – MS.VSTS для виробництва програм і ПП різного призначення за контрактами фахівців з різних держав світу і керування їхніми роботами за спеціальним тулом – Project Managment.

Фабрика програм Г. Ленца [5], яка забезпечує масове створення ПС за їх описом мовою DSL, генерацію цього опису до вихідного коду, конфігурування окремих програм, які створюються за спеціальною схемою розробки відповідно ЖЦ, прийнятого у середовищі системи .NET.

Інфраструктура системи Grid з підсистемою Etics, яка працює за принципом фабрики і забезпечує тестування, зборку та сертифікацію програм з області e-science, що використовуються для обчислення за ними різних наукових завдань.

Реалізація сайта керування фабрикою програм

Перший варіант Веб-сайта фабрики програм розділений на підрозділи, кожний з яких використовується для певних задач, а саме розробленню програм в VS.NET, специфікація артефактів чи готових програм, занесення їх у репозиторій, пошук готових КПВ або артефактів та зборка з КПВ нових продуктів. Користування сайтом відбувається шляхом використання відповідного підрозділу. Назви основних підрозділів виглядають наступним чином:

1) головна,
2) оголошення,
3) новини,
4) документи,
5) галерея,
6) контакти,
7) зв`язок.

Розділ 1) слугує для відображення оглядової та ознайомчої інформації про фабрику програм, відображає мету та сутність проекту. Також, цей розділ пов`язаний із розділом 2), який слугує для інформування користувачів проекту про останні зміни в його роботі, важливі повідомлення тощо.

Розділ 3) присвячений для різного роду публікацій щодо проблем програмної інженерії для вивчення сучасного стану проблематики фабрик програм. Аналізуючи цей розділ протягом життєвого циклу проекту можна буде відстежувати тенденції виробництва і подальшого розвитку.

Розділ 4) – основний розділ проекту, який містить всю документацію та робоче наповнення репозиторію фабрики програм. Розділ матиме чітку внутрішню структуру, орієнтовану на швидку навігацію по документації та репозиторіям.

Серед інших компонентів важливою складовою буде шаблон специфікатора. Саме він заповнюється користовучем сайта, заноситься їм у репозиторій і слугує забезпеченню роботи репозиторію фабрики.

Розділ 5) міститиме всю необхідну графічну інформацію, забезпечуючи мультимедійні розширення сайта фабрики. Планується тісна інтеграція з розділом 4).

Останні два розділи є інструментом для діалогу між користувачами проекту та його адміністрацією. Розділ 7) зворотного зв`язку забезпечуватиме комунікативні можливості з метою покращення та розвитку проекту.

Важливим компонентом систему сайта буде інтегрований модуль пошуку, що надаватиме можливість якнайшвидше знаходити необхідні документи та програми на фабриці, які є найбільш релевантними до пошукових запитів.

Перспективи розвитку

Враховуючи потреби індустрії програмних продуктів та систем, можна зробити висновок, що вдосконалення та розвиток нових, потужніших механізмів забезпечення взаємодії програмних компонентів при їх зборці є необхідним і важливим напрямком розвитку програмних ресурсів, що накопичуються у репозиторії.

У перспективі розвитку сайт фабрики програм може стати реальним «конвейєром» програмних продуктів, що міститиме репозиторій компонентів великого розвитку, з можливістю збирання нових програмних проектів на відповідній лінії виробництва ПП.

Удосконалення фабрики буде супроводжуватись нарощенням її ресурсів, збільшенням гнучкості системи, а отже, надання все більших можливостей розробникам. У майбутньому фабрики можуть стати потужним інструментом для створення різних програмних проектів з готових КПВ.

Висновки

Першим результатом нашої діяльності є запущений у тестовому режимі сайт, що можна знайти за адресою http://programsfactory.univ.kiev.ua/, де на даний момент уже розміщені деякі матеріали, щодо створення експериментального варіанта фабрики студентських програм, який буде розвиватися і надалі.

За ідею створення сайта, як спроби побудови першого інструменту фабрики програм і артефактів автори вдячні професору К.М. Лавріщевій.

1. *Лавріщева К.М.* Збіркове програмування. Теорія та практика. // Кибернетика и системный анализ. – 2009. – № 6. – С. 3 – 12.
2. *Лаврищева Е.М., Грищенко В.Н.* Сборочное программирование. Основы индустрии программных продуктов.– К.: Наук. думка, 2009. – 371 с.
3. *Андон П.І., Лавріщева К.М.* Розвиток фабрик програм в інформаційному світі // Вісник НАН України. – 2010. – № 10. – С. 15–41.

4. *Гринфильд Дж.* Фабрики разработки программ. – М., СПб., К.: Изд. дом «Вильямс», 2007. – 591 с.
5. *Лавріщева К.М.* Програмна інженерія. – Академперіодика, 2008. – 319 с.
6. *Лаврищева Е.М.* Проблема интероперабельности разнородных объектов, компонентов и систем. Подходы к ее решению // Матер. 7 Міжнародної конференції з програмування "Укрпрог – 2008" – С. 28 – 41.
7. *Лавріщева К.М.* Перспективні дисципліни програмної інженерії // Вісник НАН України. – 2008. – № 9. – С. 12 – 17.

Отримано 10.05.2011

Про авторів:

Аронов Андрій Олексійович,
студент 4-го курсу факультету кібернетики,

Дзюбенко Артем Ігорович,
студент 4-го курсу факультету кібернетики.

Місце роботи авторів:

Київський національний університет імені Тараса Шевченка.
e-mail: j-a-g-u-a-r@gala.net
e-mail: asmer@asmerok.org.ua