

К.М. ЛАВРІЩЕВА

Інститут програмних систем НАН України
просп. Академіка Глушкова, 40, Київ, 03187, Україна

ІНСТРУМЕНТАЛЬНО-ТЕХНОЛОГІЧНИЙ КОМПЛЕКС ДЛЯ РОЗРОБЛЕННЯ Й НАВЧАННЯ ПРИЙОМАМ ВИРОБНИЦТВА ПРОГРАМНИХ СИСТЕМ

Інструментально-технологічний комплекс (ІТК) призначений для розроблення й навчання технології виробництва програмних систем (ПС) із готових ресурсів і компонентів повторного використання (КПВ). У ньому реалізовані нові теоретичні засади технології програмування, які подано сукупністю теоретичних методів, засобів і інструментів. Ці засади оснащують технологію програмування новими засобами складання КПВ з урахуванням особливостей сучасних середовищ (MS.Net, Java, Corba, Eclipse та ін.), зберігання специфікованих КПВ у репозиторії для їх застосування в нових розробках ПС і досягнення їх якості. ІТК подано сукупністю простих ліній, які реалізують різні дії з опису доменів мовою DSL, її трансформації до мови програмування (МП) окремих компонентів, складання, тестування й оцінювання якості створеної з КПВ системи. Запропоновано і реалізовано нову концепцію взаємодії розроблених програм між собою в одному середовищі та в різних розподілених системах (Corba, Java, Microsoft.Net, Eclipse, Protégé). ІТК пропонує технологію навчання з лінії розроблення програм мовою C# Visual Studio (VS).Net, Java та вивчення електронного курсу «Програмна інженерія» за авторським підручником на сайті фабрики програм Київського національного університету імені Тараса Шевченка (<http://programsfactory.univ.kiev.ua>).

Ключові слова: програмні системи, компоненти повторного використання, генерувальне програмування, фабрика програм, технологія програмування, складальна технологія, взаємодія середовищ, інтерфейс, лінія продуктів, репозиторій, дистанційне навчання, електронний підручник.

В Інституті програмних систем НАН України (ІПС) впродовж багатьох років у межах фундаментальних проектів НАН «Розробка теоретичного фундаменту генерувального програмування (ГП) та інструментальних засобів його підтримки» (2007–2011) і попереднього проекту «Розробка теоретичних основ та методологічних засад компонентного програмування» (2001–2006) було взято курс на розвиток і вдосконалення різних аспектів технології виробництва програм з готових компонентів повторного використання (КПВ), спрямованої на індустріальний шлях розвитку майбутніх фабрик програм з конвеєрним

складанням, ідею якого сформулював академік В.М. Глушков ще в 1975 р.

Базовою основою для виконання цих проектів були: особисті й вітчизняні напрацювання в галузі індустрії програм, започатковані у СРСР ще в 70-х роках постановою Ради Міністрів СРСР «Программные средства как продукция производственно-технического назначения» (1974) для ініціювання розвитку відповідних теорій і систем автоматизації програм, пакетів прикладних програм у науково-дослідних інститутах СРСР та обговорення різних аспектів індустрії на багатьох наукових конференціях (1979–1992). На першому етапі індустрії Інститут кібернетики зробив великий внесок у створення науки і технології індустрії програмної

продукції (ПП) для великих ЕОМ. Було розроблено методи, технології й інструментальні засоби, а саме: Р-технологія, складальна технологія з КПВ та інтефейсу, технологія пакетів прикладних програм (Диспро, Планер, Вектор, СигмаДельта та ін.), системи автоматизації програм (Апроп, Мультипроцесист, Маяк та ін.) [1–5]. Це відповідало бурхливому розквіту технології програмування (ТП) в 90-х роках і використанню її в багатьох побудованих АСУ, СОД та системах автоматизації морського флоту СРСР тощо.

Сформувалася загальна концепція складання (об'єднання, комплексування, інтеграції, композиції), об'єктами якої були модулі за різними МП, інтерфейс їхніх зв'язків та операції складання з них різних структур програм. За цим програмуванням було захищено докторську дисертацію автора (1989), дві кандидатські дисертації (1990) й опубліковано монографію «Сборочное программирование» (1991) з теорією і практикою складального програмування та вирішення проблем несумісності типів даних, що передаються між об'єднаними різномовними модулями [6, 7]. Наступним кроком методу інтеграції були складання ПС з компонентів [8], монографія з ГП К. Чарнецькі та У. Айзенекера [9] з лейтмотивом «от ручного труда к конвейерной сборке», технологія І. Бея з частково автоматизованої взаємодії різномовних програм [10], фабрика розроблення програм Дж. Грінфілда [11] з потоковим складанням і багато інших фабрик, проаналізованих і обґрунтованих у [5] як автоматизовані лінії виробництва різних програм масового використання, в тому числі комерційні Software Product Lines Інституту програмної інженерії (Software Engineering Institute – SEI) США [12] і сучасні фабрики програм потужних індустріальних фірм з виробництва ПП (IBM, Microsystems, Sun Microsystems, Oberon, Unix, Intel та ін.).

Під час виконання наукових проектів в ПС стратегічну ідею В.М. Глушкова з конвеєрної індустрії програмних продуктів реалізовано за допомогою нових теоретичних методів, засобів та інструментів [13–18], які

оснащують сучасну ТП життєвим циклом (ЖЦ), новими засобами з досягнення якості КПВ і ПС та збагачують методи складання КПВ новими інструментами й особливостями операційних середовищ (MS.Net, Java, Corba, Eclipse та ін.), які підвищують рівень індустрії ПП, зокрема системного призначення, протягом багатьох років.

НОВІ НАУКОВІ І ПРИКЛАДНІ ЗАСАДИ З ВИРОБНИЦТВА ПРОГРАМ

У межах проекту розроблено теоретичні засади технології виробництва ПС, які докладно подано в електронній монографії [14], що містить сукупність теоретичних методів, засобів і інструментів, орієнтованих на технологію виробництва програм, застосувань, доменів і ПС із готових різномовних і різнорідних КПВ, а також на навчання спрощених ліній технології.

Навчання всіх технологічних прийомів розроблення різномовних програм та їх складання в різних поширених в інформаційному світі середовищах (VS.Net, IBM, Corba, Java, Sun Microsystems та ін.) сприяє підготовці у ВНЗ фахівців для індустрії ПП.

Нові теоретичні засади технології виробництва ПС з простих програм, КПВ і наукових артефактів охоплюють [8–31]:

1. Теоретичний апарат – методи, моделі й операції компонентної алгебри, засоби маршалінгу даних для їх перетворення при реалізації зв'язків компонентів між собою в МП, моделі проектування архітектур ПС (MDD, MDA, GDM, PIM, PSM тощо), методи збирання, трансформації і конфігурування, мови опису специфіки й понять предметної області – DSL (domain specific language), засоби автоматизованого виробництва ПС з можливістю адаптувати їх до інших умов виконання, керувати варіантами (варіабельністю) сімейств систем, оцінювати правильність, якість, витрати та вартість виготовлення ПП [8, 14, 24].

2. Теорію взаємодії програм і систем для реалізації зв'язків операційних розподілених систем між собою MS.Net↔Eclipse, Corba↔Eclipse, Java↔MS.Net через апробацію програм C#↔Java, Basic↔C++, Java↔↔C#, що працюють у розподілених системах мережі й отримують дані з СУБД або сучасних віртуальних сховищ даних он-лайн, які потребують їх перетворення до форматів архітектур на перетині середовищ виконання [21–24, 29, 30].

3. Нові, оригінальні моделі варіабельності й життєздатності, які в сукупності призначені для забезпечення адаптивності різних властивостей розроблених ПС (змінюваності, інтероперабельності та відмовостійкості) для поліпшення їх супроводу в сучасних розподілених середовищах [14, 23–25].

4. Онтологічні засади для відображення знань про прикладні домени (математика, медицина, біологія тощо) в інфраструктурі репозиторію шляхом подання опису специфікації КПВ та їхніх паспортів, уніфікованих за стандартом WSDL, для їх накопичення, організації пошуку, відбору й застосування в нових розробках ПС [14, 26].

5. Методологію проектування ПС з використанням ЖЦ, моделювання архітектури ПС за відомими моделями (MDA, MDD, GDM та ін.), генерацію і складання ПС з готових ресурсів (reuses, assets, services, artefacts, КПВ) [14–16].

6. Методи й засоби інженерії тестування й оцінювання деяких показників компонентів ПС (зокрема, якості й витрат на розроблення проектів ПС) [14, 27, 28].

7. Спектр технологій для виробництва різних КПВ, програм і ПС, реалізований в ІТК з простих ліній, а саме: накопичення готових КПВ у репозиторії ГП; розроблення і специфікація КПВ та їхніх паспортів; складання різномовних програм і компонентів у ПС за даними FDT і GDT (стандарту ISO/IEC 11404–2007) та теорії конвертування даних до форматів платформ обраних розподілених систем середовища ГП ІТК; розроблення ПС засобами DSL; конфігурування КПВ; інженерія якості і

витрат; реалізація моделі взаємодії програм, систем у середовищах VS.Net, Corba, Java, Eclipse з можливістю їх міграції з однієї розподіленої системи середовища в іншу тощо [29–35].

8. Технологію навчання студентів прийомів розроблення програм за ЖЦ у середовищах MS.Net, Java та проведення електронного вивчення курсу «Програмна інженерія» [14–19] на сайті (<http://sestudy.edu.ua>), сайтах фабрики програм КНУ імені Тараса Шевченка (<http://programsfactory.univ.kiev.ua>) [21, 22, 31] і Національного відкритого університету «ИНТУИТ» (<http://www.intuit.ru/>) [18].

Оснору ІТК становить побудований репозиторій готових КПВ та інших артефактів, які можна накопичувати й застосовувати під час складання з них нових систем чи проектів різного призначення. До середовища ГП входить Eclipse [44], який забезпечує зв'язок усіх інструментів ГП, підключення до нього механізмом плагінів необхідних загальних, прикладних інструментів (наприклад, VSTool DSL) та інших засобів для підтримання технології виробництва ПС з КПВ із навколишніх розподілених систем.

У процесі реалізації ліній виробництва в ІТК застосовано сучасні інструментальні розподілені системи:

- загальносистемні засоби (Corba, Java, VS.Net, IBM, Protégé, MCF і Tools DSL MS.Net, Eclipse–DSL та ін.);

- засоби спеціального призначення (Eclipse, Protégé) для створення моделей предметних областей і використання їх під час опису ПС у DSL і додавання КПВ до репозиторію [37, 38];

- системи програмування з МП Visual Basic, C++, Java і мови, що належать до загальносистемних середовищ, наприклад VS.Net (C#, F#, C++, Visual Basic), мови системи Corba (C++, C, Lisp, Smalltalk, Java, Pascal, PL/1, Python), Java/RMI для опису різних програм, доменів або ПС [38–43];

- нові засоби з підтримання мови DSL (Tools DSL VS.Net і Work Flow, Eclipse–DSL)

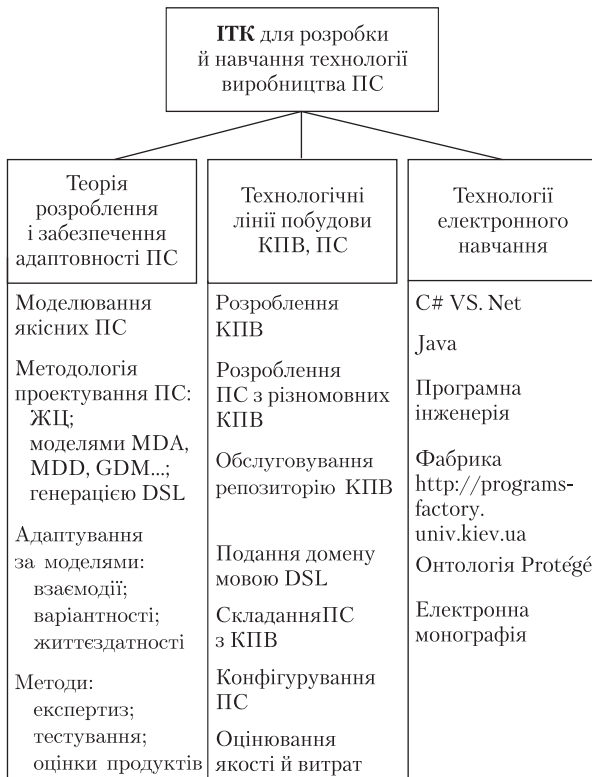


Рис. 1. Структура ІТК

для подання описів прикладних доменів [45–47];

— засоби тестування й оцінювання програм і КПВ [14, 40].

Базову платформу інтегрованого середовища ІТК — Eclipse використовують як ядро і механізм розширення за плагінами середовища новими потрібними для ГП системними засобами і компонентами (наприклад, AppFabric VS.Net і IBM, Apache Ant, Azure та ін.) для підтримання окремих ліній виробництва ПС. Фактично всі залучені до інтегрованого середовища ГП інструментальні засоби, такі як VS.Net, Corba, Protégé [37], є незалежними і їх використання зумовлене не лише потребами виробництва ПС в ІТК, а й особливостями реалізації розроблених у цьому проекті теоретичних засад [14] і ПП.

Отже, інтеграція деяких інструментів у середовище ГП має стандартний шлях, у якому

розроблено принципово нові способи зв'язку систем і середовищ (VS.Net, Corba, Java, Eclipse) між собою, що створює розширене гетерогенне середовище з різномовними ПП, потрібними різним користувачам Інтернету.

ОПИС ФУНКЦІЙ ІТК

ІТК містить мовні й інструментальні засоби для підтримання технологічних засад і функцій, орієнтованих на реалізацію ліній розроблення програм і ПС з повторних КПВ, накопичених у репозиторії ГП. Він побудований як сайт у корпоративній мережі ІПС НАН України.

Розділ «Головна сторінка». На ній наведено (рис. 1) перелік функцій ІТК, основні схеми наукових і проектних рішень щодо методології виробництва програм у середовищі ГП і методик реалізації ліній складального виробництва в ІТК, взаємодії програм, систем і технології подання деяких доменів засобами інструментальних систем середовища ГП і КПВ репозиторію.

У реалізації окремих ліній технології виробництва програм в ІТК брали участь студенти КНУ імені Тараса Шевченка і МФТІ, особливо в галузі створення експериментальної фабрики програм [17, 21, 22, 31], програмних засобів підтримання моделей взаємодії програм і систем [14, 24, 29, 30] та варіабельності ПС [14, 23–25]. Залучений до ІТК інструментарій Protégé [37] підтримує технологію онтологічного створення моделей деяких доменів і завдань програми інформатизації НАН України [38].

На цій сторінці можливо задати огляд розміщених текстових документів проекту — заключний звіт у двох книгах, електронну монографію (2007–2011), перелік підручників і наукових публікацій співробітників відділу. Вони інформують всіх користувачів про сутність наукових робіт, підготовлених у рамках фундаментальних проектів, що виконувалися у відділі «Програмна інженерія» впродовж останніх років [14, 40].

Розділ «Технології». В ньому подано розроблену інтегровану технологію виробни-

цтва ПС і СПС із КПВ за наступними спрощеними лініями:

- фабрика програм, яка містить специфікації КПВ, їхні паспорти, курси навчання програмуванню C# VS.Net і програмної інженерії;
- обслуговування репозиторію КПВ;
- складання різномовних компонентів у ПС з конвертуванням несумісних типів даних;
- конфігурування КПВ у складну структуру ПС за точками варіантності;
- опис доменів мовою DSL;
- оцінювання якості й витрат на розроблення ПС;
- онтологія предметної області;
- генерація загальних типів даних до фундаментальних;
- проектування прикладних ПС із сервісних ресурсів Інтернету.

Ці прості технології реалізуються відповідними лініями, зміст яких наведено нижче.

Фабрика програм. Лінії розроблення програм в ІТК будуються після об'єктного аналізу відповідної Про і компонентного методу реалізації об'єктів з завданням інтерфейсів, необхідних під час складання компонентів і КПВ у складні ПС. Нині відповідно до [5, 10, 11], діє багато різних фабрик програм (рис. 2), що ґрунтуються на різних методах програмування і підходах до опису інтерфейсів складання КПВ.

Ідея і концепція створення студентської фабрики програм виникли в автора у процесі викладання курсу «Технологія програмування ІС» на факультеті кібернетики КНУ імені Тараса Шевченка.

Відповідно до неї студенти розроблятимуть наукові артефакти під час виконання лабораторних і дипломних робіт за *лініями продуктів*, подібними до Software Product Lines SEI [5, 12], стандарту WSDL для їх опису і подання в студентську фабрику програм. У ній реалізовано спочатку три лінії побудови окремих КПВ [5, 21, 31].

Перша лінія — це загальна схема процесів ЖЦ для побудови деякого виду артефакту, КПВ, за якою розроблено лінію створення

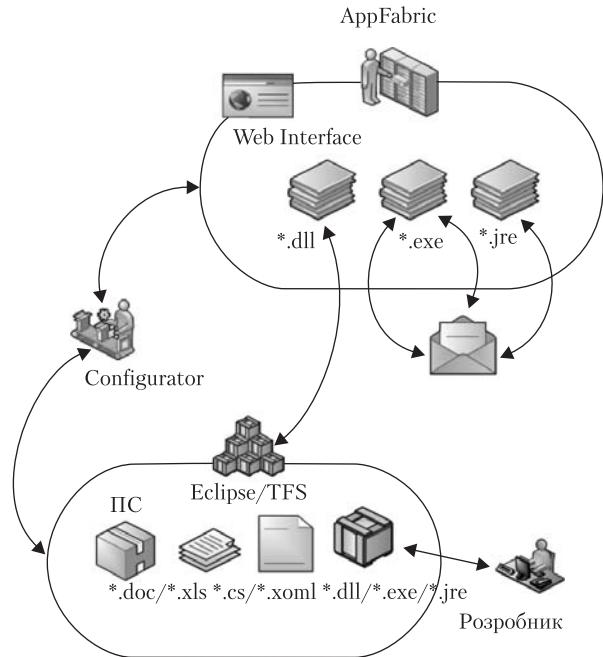


Рис. 2. Загальна структура фабрики програм

програм у VS.Net. Реалізація процесів лінії закінчується визначенням артефактів у класі задач Про, специфікацією паспортних даних і розміщенням їх описів у репозиторії для подальшого застосування в середовищі Visual Studio.Net.

Друга лінія — це добір готових програм, артефактів із репозиторію за їхніми функціями з метою визначення можливості їх застосування для розроблення нового програмного артефакту або ПС.

Третя лінія — складальна, що забезпечує складання ПС з готових ресурсів із застосуванням знов розроблених і дібраних програм і артефактів у репозиторії.

Студентські артефакти є науковими, їхнім продуктом можуть бути деякі методи або наукові алгоритми й КПВ у МП з математики, фізики, біології тощо.

Технологія обслуговування репозиторію КПВ. Репозиторій розроблений у середовищі VS.Net на основі MS SQL Server 2005 з такими функціями в ІТК:

- запис компонента і його паспорта в репозиторій;

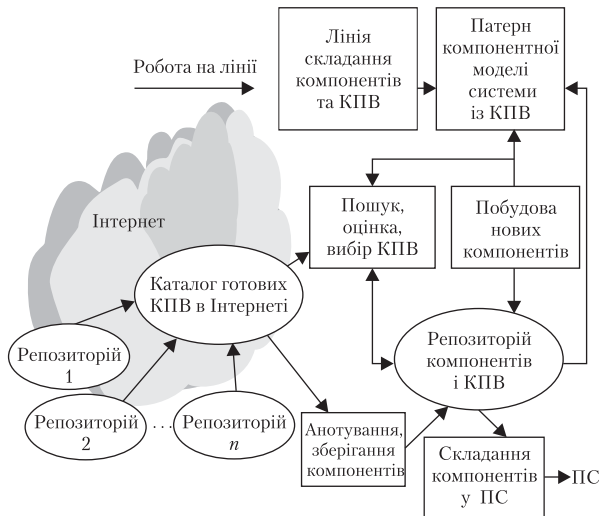


Рис. 3. Схема роботи репозиторію на фабриці

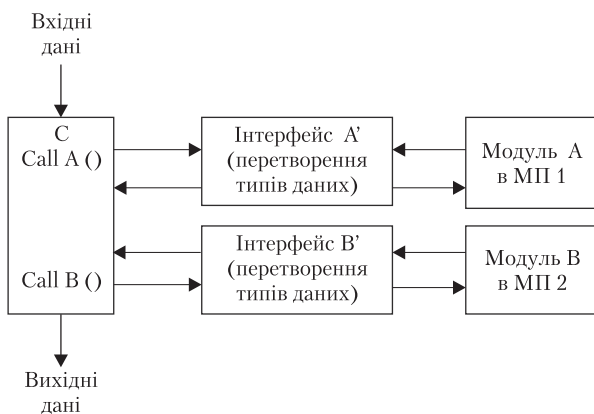


Рис. 4. У центрі схеми – інтерфейсний посередник двох різномовних компонентів А, В

- вибір КПВ для аналізу його застосування в новій РС як готового;
- обчислення КПВ за поданням значень параметрів;
- виведення результату виконання КПВ;
- внесення змін у КПВ;
- документування КПВ (друк опису програми КПВ і паспорта);
- контроль версій КПВ.

Паспортизація КПВ виконується відповідно до стандартного шаблону WSDL, використаного у системі Etics проекту Grid, узятя нами як еталон і слугує засобом по-

дання КПВ і відповідних паспортів у каталозі репозиторію ГП (рис. 3).

У загальному випадку репозиторій – це система засобів для зберігання, поповнення напрацьованих КПВ. Він містить у собі інфраструктуру розробки РС з компонентів, організацію доступу до КПВ, які розташовані в ньому для подальшого їхнього застосування в нових проектах. Наприклад, репозиторії Інтернету типу бібліотек GreenStone і Matlab надають величезну кількість готових програм наукового, зокрема математичного, типу. Вони орієнтовані на математиків, фізиків і інших фахівців предметних областей.

Складання різномовних компонентів. На сьогодні є багато різних підходів зі складання (інтеграції) різномовних і різноплатформених компонентів [6, 8, 40–44]. Насамперед складання КПВ здійснюють викликами типу RPC, RMI, IContract для забезпечення зв'язків між компонентами РС через:

- *інтерфейсний посередник* як перехідний міст (stub, skeleton) між КПВ у різних МП у загальних системах (IBM, VS.Net, Corba, COM тощо);
- *інтерфейси* мовами IDL, API, IContract та ін. для програм у сучасних МП;
- *науковий інтерфейс* (SIDL) для взаємодії програм e-science (C++, C, Fortran 77, Java тощо);
- *примітивні функції* спеціальних бібліотек системи VS.Net (CLR – common language runtime, CTS – common type system і CLS – common language specification).

Різні варіанти складання з використанням теорії перебудови несумісних типів даних, що передаються між КПВ [6–8], реалізовано в ІТК на спрощених лініях:

- складання одно- та різномовних програм у середовищі VS.Net (C# і Java, Basic і C++) з використанням бібліотек CRL, CTS і CLS;
- складання різномовних програм з КПВ у середовищах Corba–Eclipse та Apache Ant за інтерфейсним посередником типу ORB, IDL, SIDL тощо (рис. 4).

Таким чином, в ІТК реалізовано кілька варіантів складання різномовних програм для сучасних розподілених систем Visual

Studio, Corba, Eclipse через інтерфейсний посередник за параметрами і типами даних, що передаються між цими програмами.

Реалізація зв'язків між середовищами Visual Studio ↔ Eclipse і Corba ↔ Eclipse в ІТК продемонстрована відповідними прикладами в розділі сайту «Взаємодія».

Конфігурування КПП з точками варіабельності ПС. Конфігурування є механізмом об'єднання КПП у варіабельну структуру з точками змінювання деяких КПП за вимогами замовника [23, 34, 46].

КПП, позначені точками в моделі варіабельності, можуть бути *обов'язковими* — наявними в усіх ПС сімейства; *необов'язковими* — що наявні лише в деяких ПС або в кількох точках можливих варіантів; *індивідуальними* — які створюють на замовлення для заміни або додавання в ПС.

Точки варіантності виділяють позиції в описі КПП або ПС, в яких можна виконати зміни деяких функцій ПС. *Варіант* — це відокремлена функціональність, якій відповідає точка варіантності, що задається ідентифікатором як можлива альтернатива.

В ІТК наведено програму розв'язування квадратного рівняння з трьома точками варіабельності з використанням Work Flow MS.Net. За цим прикладом будується відповідний конфігураційний файл, який забезпечує виконання цієї програми, і в майбутньому буде реалізовано операції змінювання програм за точками варіантності.

Лінія опису і генерації доменів DSL. ГП об'єднало всі парадигми програмування (об'єктне — ООП, компонентне — КОП, сервісне — СОП тощо). Воно орієнтоване на створення (генерацію, складання, композицію) ПС (applications), доменів (domains), сімейств програмних систем — СПС (software product families) тощо. Новим у ГП порівняно з ООП, КП, СОП є метамова опису домену за термінологією, яка визначає сутність простору понять і завдань домену засобами DSL [40, 49], зокрема модель ПрО, модель характеристик зі змінними параметрами і функціями, що залежать від умов функціонування системи, платформ і засо-

бів трансформації типів даних залежно від місця їх розташування в середовищі. Як приклад в ІТК реалізовано домен ЖЦ стандарту ISO/IEC 12207–2007, який подано графічно і текстово в XML за допомогою нового інструменту Eclipse–DSL [47].

Оцінювання якості і витрат продукту виконують програмним модулем Softest, призначеним для прогнозування оцінок трудовитрат і вартості розроблення ПС за відомою моделлю СОСОМО II [14, 28]. Остання модель містить: *попередню модель (application composition model)* з оцінювання трудовитрат ПС на *ранніх* стадіях розроблення; *передпроектну модель (early design model)* з оцінкою трудовитрат, починаючи з перших стадій ЖЦ з розподілом витрат на процеси ЖЦ; *детальну модель (post architecture model)*, яка уточнює попередні оцінки трудовитрат при супроводі ПС. Кожна з моделей має аналітичний вигляд, реалізований у модулі Softest за вхідними параметрами, що задаються на сайті для отримання результату, і розраховує *номінальну (середню) тривалість* розроблення за формулою $D_{\text{ном}} = 3,67 \cdot T_{\text{ном}}^{(0,38 + 0,2 \cdot (B - 1,01))}$.

У майбутньому до модулю буде підключено оцінювання якості КПП і ПС за стандартною методикою.

Таким чином, реалізація ліній поступового виробництва ПС з простих ліній за допомогою сучасних засобів і інструментів дає можливість зробити висновок про наближення ТП до потреб сучасних фабрик програм.

Розділ «Взаємодія». Спектр технологій за наведеними лініями доповнено засобами взаємодії програм, систем і середовищ відповідно до розробленої в межах проекту теорії взаємодії [21, 24].

Сутність її полягає в реалізації інтерфейсів інтеперабельності шляхом удосконалення і загальних методів, і засобів доступу до середовищ Corba, VS.Net, Java й Eclipse у напрямі забезпечення перенесення програм через перетин середовищ в інше середовище для взаємодії програм цих середовищ між собою з розміщенням цих програм у репозиторію ІТК [5, 13, 14].

Ці операційні системи середовища ГП підтримують процеси ЖЦ розроблення різно-рідних програм і методи їх об'єднання в різні структури ПП через спеціальні механізми зв'язку, реалізовані в кожній із них.

Створені в середовищах різно-рідні програми не можуть взаємодіяти між собою з багатьох причин, пов'язаних з різними підходами до реалізації їхніх інтерфейсів і форматів архітектур комп'ютерів тощо. Для вирішення цієї проблеми в ІТК запропоновано додаткові оригінальні засоби взаємозв'язку за допомогою моделі взаємодії з параметрами інтеперабельності, перенесення й адаптивності програм і середовищ. Головне, що практично зроблено в цьому ІТК, — це реалізація нової моделі взаємодії, що розширює «водорозділ» між середовищами для повільного «плавання» програм і систем з метою виконання чи змінювання деяких пов'язаних між собою КПВ [23, 28, 29].

Реалізовані *нові засоби взаємодії* поки не мають прототипу. Практично вони перевірені програмно на сайті за наступними варіантами взаємодії.

Взаємодія програм у різних МП встановлюється зв'язками з іншими програмами через інтерфейсні мови (IDL, API, SIDL) і відповідними механізмами виклику: RPC системи Sun Microsystems, RMI системи Java, ORB системи Corba, MSIL системи VS.Net, IContract системи MCF тощо.

Виклики RPC, RMI містять параметри, що передаються між програмами або компонентами для виконання обчислень з поверненням результату програмі, що викликає.

В ІТК реалізовані конкретні інтерфейси взаємодії на прикладі програм мовами Visual Basic і C++, що виконуються в середовищі VS.Net в ІТК. Як приклад для ІТК взято програми з монографії І. Бея [10] з опису псевдовипадкової послідовності та розрахунку максимального, мінімального й середнього значень цієї послідовності із взаємодією програм через інтерфейсний посередник — проміжний ланцюг між програмами, що обмінюються даними між собою і перетворюють несумісні типи даних.

Взаємодія забезпечується за допомогою розроблення програм типу DLL в середовищах Visual C++ - Visual Basic і реалізації цих програм з використанням файлів *.cpp, *.h, DEF і LIB. Зазначені програми обчислюють максимальне, мінімальне та середнє значення цієї послідовності і видають результат на дисплей сайту ІТК.

Взаємодія розподілених систем Corba ↔ Eclipse, Java, занурених у середовище ІТК, реалізована на прикладі програми авторизації ключів мовою Java [4, 20]. Цією програмою продемонстровано реалізацію моделі взаємодії двох середовищ з використанням механізму RMI передавання повідомлень у системі Corba [43]. Під час реалізації використовується проміжний шар для об'єднання програмного коду об'єктів, інтерфейс доступу до яких описується мовою IDL. Типи даних об'єктів у МП відображаються в типи IDL і зворотно за допомогою брокера ORB щодо компіляції IDL-описів компонентів, створення клієнтських і серверних stub- і skeleton-посередників.

Реалізований механізм відсутній на платформі .Net, і тому для взаємодії тут використовують сторонні пакети (наприклад, IIOPNet з відкритим вихідним кодом). За допомогою утиліти з цього пакета IDL-опис об'єкта перетворюється в функції CLS-сумісної бібліотеки, а потім залучається до застосування.

Зв'язок з брокером ORB здійснюється через канал зв'язку, який відіграє роль клієнтського стабу, виконуючи маршалінг і демаршалінг переданих через нього даних.

Таким чином, в ІТК продемонстровано реалізацію завдання взаємодії прикладних компонентів у середовищах Microsoft.Net (клієнтська частина застосування) і Java (серверна частина) засобами системи Corba з використанням MS.Net і Java.

Взаємодія систем і середовищ. Взаємодія Eclipse і VS.Net реалізована в системі ГП на прикладі деякої програми мовою C# VS.Net, що передається через плагін у репозиторій Eclipse. Засобами цих середовищ розроблено програму складання з компонентів,

специфікованих мовами C# і Java. Зв'язок між ними реалізовано за допомогою бібліотек типів CTS і класів CLR системи Visual Studio і розміщення програм мовою C# у репозиторії Eclipse. Для взаємодії Eclipse із платформою .Net мовою C# використовують плагін Emonic і програму NAnt. Створений проект програми імпортується в Eclipse і конфігураційний exe-файл складання, побудований за допомогою External Configuration Tools.

Отже, в ІТК, на відміну від розглянутих систем і середовищ, створено можливість відкритого доступу до нових механізмів взаємодії у середовищах Corba, VS.Net, Java, що розширює функції розробленого ІТК ГП із взаємодії програм, систем, середовищ.

Розділ «Інструменти». До інструментальних засобів ІТК належать Protégé й Eclipse.

Система Protégé призначена для створення моделей різних предметних областей за онтологією і знаннями про поняття і зв'язки між ними. Цей засіб підключений до Eclipse, проходить процес апробування на прикладі деякої Про і моделювання знань у базі знань цього інструменту [26]. Систему Protégé в ІТК використовують для розроблення онтологій предметних областей за допомогою *слотів і фасет*.

Слоти слугують для опису властивостей класів та їхніх екземплярів (і атрибутів). Згідно з фреймовою моделлю, слот — це фрейм. Слоти визначаються незалежно від класу, і один і той самий слот може належати до різних класів. *Фасети* дають змогу вводити обмеження на типи і діапазони значень екземплярів (атрибутів) мовою XML. Для слоту задається кількість значень, обмеження на значення (ціле, рядкове тощо) і граничні значення (мінімальне і максимальне). За допомогою слотів і фасет будують моделі Про за технологією проектування, реалізованою в ІТК.

Eclipse містить різні IDE для розроблення ПС у різних МП з використанням стандартного відкритого інтерфейсу API для доступу до нього. Середовище розроблення має набір розширень RPC: редактори, панелі, перспек-

тиви, модуль CVS, модулі Java Development Tools (JDT) і Java IDE, спрямовані на групове розроблення. Це середовище інтегроване із системами керування версіями CVS для інших систем (наприклад, Subversion, MS SourceSafe), C/C++ Development Tools (CDT) і засобами мов Fortran, PHP тощо.

Множина розширень доповнює середовище Eclipse менеджерами для роботи з базами даних, серверами застосувань тощо. Eclipse написаний мовою Java, використовує стандартні Java-бібліотеки Swing і підключає через плагіни не лише мовою Java, а й іншими мовами — C/C++, Perl, Ruby, Python, PHP, Erlang, Pascal тощо. Eclipse розширено засобом Eclipse-DSL для побудови моделей доменів мовою DSL і розподіленими системами VS.Net, Corba та Java для об'єднання різномовних програм і перенесення їх у репозиторій ГП.

Розділ «Презентації». Презентація «Прикладна система» на сайті ІТК демонструє розроблену в межах програми інформатизації НАН України інформаційну систему для підтримання повсякденної діяльності відділу міжнародних зв'язків Президії НАН України. Для доступу до неї розроблено автоматизований веб-комплекс, який забезпечує формування довідок про відряджених за кордон працівників Президії НАН України, звітів про відрядження. Усі дані зберігаються в єдиній базі даних цього комплексу. Його опис, інструкція щодо роботи з ним користувачів, а також слайди про нього містяться в ІТК. Ними можна користуватися, оскільки цей веб-комплекс готовий для застосування в підрозділах інститутів і Президії НАН України та інших організаціях. Сайт передано до Державного департаменту інтелектуальної власності при МОН [38].

Крім того, розміщено дві презентації-довідки автора з програмної інженерії і фабрики програм та індустрії програм на міжнародному науковому конгресі при Кабміні України (18 листопада 2011 р.).

Розділ «Навчання». Враховуючи міжнародну практику навчання студентів, а саме, навчання за електронними посібниками, в

ІТК використано підручники (українською й російською мовами) з таких навчальних дисциплін:

– «Програмна інженерія» [14, 17, 18], що доступний на сайті фабрики програм КНУ (<http://programsfactory.univ.kiev.ua>) і Національного відкритого університету «ИНТУИТ» (www.intuit.ru);

– технологія розроблення програм мовою Java («Самовчитель Java» І.Ш. Хабібулліна);

– технологія розроблення програм засобами C# VS.Net (опис цієї технології див. далі).

Електронне навчання охоплює курси за програмою Міністерства «Програмна інженерія та Java» для навчання студентів, аспірантів, деяких співробітників ІПС НАН України й користувачів фабрики програм, корпоративного сайту ІПС (<http://192.168.220.87/проект-2011/index.html>), що реалізує спектр технологій за лініями поступового виготовлення СПС із готових КПВ і складання їх у готовий ПП у середовищах VS.Net, Corba, Java, Eclipse тощо, а також курси навчання.

Технологія розроблення програм на платформі VS.Net. В розробленні цієї технології приймала участь с.н.с. ІПС і доцент МНТУ ім. Юрія Бугая Т.М. Коротун, яка викладала курс програмування в системі MS.Net з мови C# на прикладі завдань лабораторних і дипломних робіт.

Платформа VS.Net забезпечує підтримання проектування і реалізації ПС з використанням різних МП, що входять до її складу. Вона містить *загальномовне середовище* часу виконання CLR, ідентичне Java Runtime Environment. Будь-яка програма, написана для цієї платформи, є *керованим кодом* (managed code) і компілюється в бінарний вигляд, зрозумілий для .Net runtime. У середовищі Visual Studio MS.Net реалізовано модель програмування за шаблоном з генерацією заготовок коду для програм різних типів: консольних, бібліотек, Windows-, Web-форм, Web-сервісів тощо.

В ІТК розроблено лінії створення консольних програм мовою C#, локальних Windows-застосувань та обслуговування бібліотек компонентів DLL. Ці лінії навчання вдосконалюватимуться у напрямку розширення тех-

нології складання готових програм у цьому середовищі, а також додавання нових МП.

ВИСНОВКИ

Статтю подано в період, коли Кабінет Міністрів України звернув увагу на проблеми розвитку вітчизняної індустрії програмних продуктів для отримання доходів від неї, оскільки частка національного доходу в міжнародному обсязі ПП становить 1,3%. Було проведено міжнародний науковий конгрес (17–18 листопада 2011 р.) з проблем індустрії ПП і слухання на засіданні Верховної Ради (14 грудня 2011 р.) щодо економічних, комерційних і навчальних питань індустрії ПП в Україні. Це означає початок другого етапу в історії програмної індустрії, оскільки такі питання в незалежній Україні досі не розглядалися. Після розпаду СРСР курс було взято на аутсорсинг і придбання готових закордонних ПП різного призначення для впровадження їх у державні організації і підготовку студентів для роботи в закордонних фірмах, яких у країні налічується близько 2000.

Разом з тим в ІПС НАН України впродовж багатьох років виконували наукові проекти в напрямі послідовного опрацювання теоретичних і прикладних питань розвитку технології виробництва різних програмних продуктів. У результаті розроблено новітні *теорії* і *технології* виготовлення ПП у сучасних операційних *середовищах* (на прикладі діючих фабрик програм), оригінальні автоматизовані *лінії розроблення* окремих КПВ і ПС зі спектру комплексних технологій та механізми накопичення *готових КПВ* і *наукових артефактів* у репозиторіях і е-бібліотеках, як необхідних комплектуючих «деталей» для розвитку індустрії ПП і навчання її методів студентів ВНЗ. Отже, в ІПС НАН України в напрямі індустрії ПП розроблено:

– *комплексну технологію виробництва СПС* за найпростішими лініями: розроблення окремих КПВ; їх сервісного обслуговування в репозиторії; складання і конфігурування КПВ у СПС; взаємодії програм, систем і середовищ між собою; тестування КПВ; оцінювання якісних показників СПС і витрат на

розроблення. Ця технологія є сучасною і актуальною з позицій індустрії програмної продукції, надає всі необхідні теоретичні й прикладні засади для регламентованого її використання в колективах-розробниках України;

– *інструментально-технологічний комплекс* як сучасний веб-сайт корпоративної системи НАН України (<http://192.168.220.87/проект-2011/index.html>), що введений в Інтернет на сервері Академії педагогічних наук (<http://sestudy.edu-ua.net>), що реалізує наведений у роботі спектр комплексних технологій виробництва СПС у середовищі систем VS.Net, Corba, Java, Eclipse тощо; сайт *експериментальної фабрики програм* на факультеті кібернетики КНУ імені Тараса Шевченка (<http://programsfactory.univ.kiev.ua>) з навчання технологій програмування C#, Java і курсу програмної інженерії (за жовтень-листопад 2011 р. було зареєстровано понад 3000 звернень викладачів та студентів);

– *методику виробництва*, висвітлену в електронній монографії «Нові теоретичні засади технології виробництва сімейств програмних систем у контексті генерувального програмування» (К.М. Лавріщева, Г.І. Коваль, Л.П. Бабенко, О.О. Слабоспицька, П.П. Ігнатенко), здану у ДРНТІ України за № 67, жовтень 2011 р.

Електронна монографія, яка буде подана у розділі «Презентації» веб-сайту (<http://sestudy.edu-ua.net>), є початком впровадження індустріального напрямку виготовлення програмної продукції в НАН України.

СПИСОК ЛІТЕРАТУРИ

1. *Сергиенко И.В.* Некоторые вопросы разработки общесистемного программного обеспечения ЭВМ и систем. – К.: ИК им. В.М. Глушкова, 1987. – 29 с.
2. *Сергиенко И.В.* Вопросы разработки прикладного математического обеспечения ЭВМ и систем. – К.: ИК им. В.М. Глушкова, 1987. – 36 с.
3. *Лаврищева Е.М.* Основы технологической подготовки разработки прикладных программ СОД. – К., 1987. – 30 с.
4. *Лаврищева Е.М.* Становление и развитие модульно-компонентной инженерии программирования в Украине. – К.: ИК им. В.М. Глушкова, 2008. – 33 с.
5. *Андон П.І., Лаврищева К.М.* Розвиток фабрик програм в інформаційному світі // Вісник НАН України. – 2010. – № 10. – С. 15–41.
6. *Лаврищева Е.М., Грищенко В.Н.* Сборочное программирование. – К.: Наук. думка, 1991. – 213 с.
7. *Лаврищева Е.М., Грищенко В.Н.* Сборочное программирование. Основы индустрии программных продуктов. – К.: Наук. думка, 2009. – 371 с.
8. *Грищенко В.М.* Метод об'єктно-компонентного проектування програмних систем // Проблеми програмування. – 2007. – № 2. – С. 113–125.
9. *Чарнецки К., Айзенкер У.* Порождающее программирование. Методы, инструменты, применение. – СПб: Питер, 2005. – 730 с.
10. *Бей И.* Взаимодействие разноразличных программ. – М.: Вильямс, 2005. – 868 с.
11. *Гринфилд Дж., Шорт К., Кук С. и др.* Фабрики разработки программ. Поточковая сборка типовых приложений, моделирование, структуры и инструменты. – М.: Вильямс, 2007. – 591 с.
12. Framework for Software Product Line Practice // http://www.sei.cmu.edu/productlines/frame_report/index.html.
13. *Андон Ф.И., Лаврищева Е.М.* Методы инженерии распределенных компьютерных приложений. – К.: Наук. думка, 1997. – 228 с.
14. *Лаврищева К.М., Коваль Г.І., Бабенко Л.П. та ін.* Нові теоретичні засади технології виробництва сімейств програмних систем у контексті генерувального програмування. – К., 2011. – 377 с.
15. *Бабенко Л.П., Лаврищева К.М.* Основы программной инженерии. – К.: Знання, 2001. – 269 с.
16. *Лаврищева Е.М.* Методы программирования. Теория, инженерия, практика. – К.: Наук. думка, 2006. – 451 с.
17. Основы инженерии качества программных систем / Ф.И. Андон, Г.И. Коваль, Т.М. Коротун и др. – К.: Академперіодика, 2007. – 672 с.
18. *Лаврищева Е.М., Петрухин В.А.* Методы и средства инженерии программного обеспечения. – М.: МОН Российской Федерации, 2007. – 415 с.
19. *Лаврищева К.М.* Програмна інженерія. – К.: Академперіодика, 2008. – 319 с.
20. *Коваль Г.І., Колесник А.Л., Лаврищева К.М., Слабоспицька О.О.* Удосконалення процесу розроблення сімейств програмних систем елементами гнучких методологій // Проблеми програмування. – 2010. – № 2–3. – С. 261–270.
21. *Лаврищева К.М.* Концепція індустрії наукового софтвера і підхід до обчислення наукових задач // Проблеми програмування. – 2011. – № 1. – С. 3–17.
22. *Анісімов А.В., Лаврищева К.М., Шевченко В.П.* Про індустрію наукового софтвера // Theoretical and Applied Aspects of Cybernetics. Proc. Int. Sci. Conf. Students and Young Scientists (21–25 Feb. 2011, Kyiv, Ukraine). К.: Bukrek, 2011. – P. 65–67.

23. Лаврищева К.М., Слабоспицька О.О., Коваль Г.І., Колесник А.Л. Теоретичні аспекти керування варіабельністю в сімействах програмних систем // Вісник КНУ, сер. фіз.-мат.наук. — 2011. — № 1. — С. 151–158.
24. Лаврищева К.М. Взаємодія програм, систем й операційних середовищ // Проблеми програмування. — 2011. — № 3. — С. 13–24.
25. Ігнатенко П.П. Життєздатні програмні системи. Концептуалізація підходу до автоматизації систем організаційного керування // Проблеми програмування. — 2006. — № 3. — С. 33–44.
26. Бабенко Л.П. Онтологічний підхід к специфікації свойств програмных систем и их компонентов // Кибернетика и системный анализ. — 2009. — № 1. — С. 180–187.
27. Коротун Т.М. Моделі і методи тестування програмних систем // Проблеми програмування. — 2007. — № 2. — С. 76–84.
28. Андон П.І., Суслов В.Ю., Коротун Т.М. та ін. Визначення витрат на створення ПЗ АС // Проблеми програмування. — 1998. — № 3. — С. 23–34.
29. Островский А.И. Подход к обеспечению взаимодействия программных сред JAVA и MS.Net // Проблеми програмування. — 2011. — № 2. — С. 37–44.
30. Радецький І.О. Один з підходів до забезпечення взаємодії середовищ MS.Net і Eclipse // Проблеми програмування. — 2011. — № 2. — С. 45–52.
31. Аронов А.О., Дзюбенко А.І. Підхід до створення студентської фабрики програм // Проблеми програмування. — 2011. — № 3. — С. 42–49.
32. Коваль Г.І. Підхід до моделювання якості сімейств програмних систем // Проблеми програмування. — 2009. — № 4. — С. 49–58.
33. Слабоспицька О.О. Технологічна модель процесу автоматизованого виробництва сімейств програмних систем // Проблеми програмування. — 2011 — № 1. — С. 39–48.
34. Колесник А.Л. Підхід до конфігурування компонентів повторного використання // Проблеми програмування. — 2011. — № 4. — С. 57–66.
35. Стеняшин А.Ю. Про формальний опис типів і структур даних в різномірних програмах // Проблеми програмування. — 2011. — № 2. — С. 50–61.
36. Нейгел К., Івсен Б., Глини Дж. и др. С# 2008 и платформа .NET 3.5 для профессионалов / пер. с англ. — М.: Вильямс, 2009. — 1392 с.
37. Зінкович В.М. Онтологічне моделювання предметної області з проблематикою e-science // Проблеми програмування. — 2011 — № 3. — С. 91–99.
38. Грищенко В.М., Куцаченко Л.І. Автоматизована інформаційна система підтримки міжнародної діяльності НАНУ // Державний департамент інтелектуальної власності. Свідоцтво № 2304 від 23.12.2009.
39. Нэш Т. С# 2010. Ускоренный курс для профессионалов / пер. с англ. — М.: Вильямс, 2010. — 592 с.
40. Лаврищева К.М. Генерувальне програмування програмних систем і сімейств // Проблеми програмування. — 2009. — № 1. — С. 3–16.
41. <http://nant.sourceforge.net>.
42. Эммерих В. Конструирование распределенных объектов. Методы и средства программирования интероперабельных объектов в архитектурах OMG/CORBA, Microsoft/COM и Java/RMI / пер. с англ. — М.: Мир, 2002. — 510 с.
43. Naughton P., Schildt H. Java 2: The Complete Reference. — Berkeley: Osborne McGraw-Hill, 1999. — 1108 p.
44. Гамма Э., Бек К. Расширения Eclipse: принципы, шаблоны и подключаемые модули / пер. с англ. — М.: КУДИЦ-ОБРАЗ, 2005. — 384 с.
45. Protégé-Frames User's Guide // http://protegewiki.stanford.edu/wiki/PrF_UG_all.
46. Колесник А.Л. Механізми забезпечення варіабельності в сімействах програмних систем // Проблеми програмування. — 2010. — № 1. — С. 35–44.
47. Walkthrough. Domain-Specific Language (DSL) Tools, 2005.

Е.М. Лаврищева

Институт программных систем
Национальной академии наук Украины
пр. Академика Глушкова, 40, Киев, 03187, Украина

ИНСТРУМЕНТАЛЬНО-ТЕХНОЛОГИЧЕСКИЕ
КОМПЛЕКСЫ ДЛЯ РАЗРАБОТКИ
И ОБУЧЕНИЕ ПРИЕМАМ ПРОИЗВОДСТВА
ПРОГРАММНЫХ СИСТЕМ

Инструментально-технологический комплекс (ИТК) предназначен для разработки и обучения технологии производства программных систем (ПС) из готовых ресурсов и компонентов повторного использования (КПИ). В нем реализованы новые теоретические основы технологии программирования, которые представлены совокупностью теоретических методов, средств и инструментов. Эти принципы оснащают технологию программирования новыми средствами составления КПИ с учетом особенностей современных сред (MS.Net, Java, Corba, Eclipse и др.), хранения специфицированных КПИ в репозитории для их применения в новых разработках ПС и достижения их качества. ИТК представлен совокупностью простых линий, которые реализуют различные действия по описанию доменов на языке DSL, его трансформации в языках программирования (ЯП) отдельных компонентов, сборки, тестирования и оценки качества созданной из КПИ системы. Предложена и реализована новая концепция взаимодействия разработанных программ между собой в одной среде и в различных распределенных системах (Corba, Java, Microsoft.Net, Eclipse, Protégé). ИТК предлагает технологию обучения по линии разработки программ на языке C # Visual Studio (VS). Net, Java и изучение электронного курса «Программная инженерия» по авторскому

учебнику на сайте фабрики программ Киевского национального университета имени Тараса Шевченко (<http://programsfactory.univ.kiev.ua>).

Ключевые слова: программные системы, компоненты повторного использования, генерирующее программирование, фабрика программ, технология программирования, сборочная технология, взаимодействие сред, интерфейс, линия продуктов, репозиторий, дистанционное обучение, электронный учебник.

K.M. Lavrisheva

Institute of Software Systems
of National Academy of Sciences of Ukraine
40 Glushkov Ave., Kyiv, 03187, Ukraine

INSTRUMENTAL AND TECHNOLOGICAL
COMPLEX FOR DEVELOPMENT
AND TRAINING METHODS
OF MANUFACTURING SYSTEMS SOFTWARE

Instrumental and technological complex — ITC is intended for development and learning technology aircraft with ready Reuses. It reflected new theoretical frame-

work developed technology programming, submitted a set of theoretical methods, tools and instruments. These principles giving programming technology means new collection Reuses considering the specifics of modern environments (MS.Net, Java, Corba, Eclipse, etc.), storage Reuses specified in the repository for their use in new aircraft development and achievement of quality reuses. Technology ITC filed a set of simple lines that implement various actions to realize the domains of language DSL, transforming it to the PL of individual components, assembly, testing and quality evaluation system was created with Reuses. Proposed in ITC technology training for a new line of programming language C# VS.Net, Java and e-learning course «Software Engineering» for the copyright of electronic books online KNU (<http://programsfactory.univ.kiev.ua>).

Keywords: programs system, reuses, generative programming, assembling or compositional programming, programs factory, interconnection or interoperability, interface, product line, repository, distance learning, electronic textbooks.

Стаття надійшла 17.10.2011 р.