УДК 04.004

Індустріальний підхід до розробки і виконання прикладних систем в гетерогенних розподілених середовищах

Лавріщева К.М., Стеняшин А.Ю.

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

lavrischeva@gmail.com, andrey.stenyashin@gmail.com

Анотація. *Дано опис підходів з розвитку індустрії виробництва програм із компонентів повторного використання (КПВ). Запропоновано теоретичні ы практичні аспекти індустрії програм, обчислень та даних. Розглянуто підхід до їх реалізації відповідно принципу взаємодії програм і систем у сучасних операційних середовищах (Corba,Vs.Net, Java, Eclipse тощо). Сформульовані новітні концепції технології індустріального виробництва складних програм, за якими створено першу студентську фабрику програм в київському національному університеті для накопичення знань про наукові артефакти і програми, що розробляються в дипломних і магістерських роботах за науковими дисциплінами навчання. На фабрики діють три технологічні лінії щодо подання КПВ, програмування їх мовами VS.Net і вивчення основ програмної інженерії за підручником автора. До фабрики звернулося більш 5000 користувачів і студентів інших університетів ВНЗ України і СНГ.*

Ключові слова

Індустрія програм, компоненти повторного використання – КПВ, методика виробництва, фабрика програм, індустрія даних, обчислення, технологія, технологічні лінії, продуктові лінії, взаємодія, гетерогенне середовище, інструменти, засоби, фабрика КНУ, навчання , програмна інженерія.

Передмова

У даний час в інформаційному світі накопичено велику кількість якісних різноманітних програмних і інформаційних ресурсів у електронних бібліотеках, зокрема, наукових, як елементів індустрії наукового софтвера, потрібних Європейському проекту Grid, діють фабрики програм системного і прикладного характеру та сформувалися базові основи індустріального виробництва програмних продуктів (ПП) [1-4].

Перехід до індустрії програм і систем пов'язаний також бурним розвитком високоефективної елементної бази, нової багатоядерної, процесорної конфігурації комп'ютерів, кластерів тощо. Треба сказати,що комп'ютерна індустрія опереджає більш ніж на порядок розвиток як з теоретичної, так і з практичної точки зору інші види індустрії програмних систем, обчислень і даних. Така індустрія потрібна для виконання великомасштабних обчислень дуже складних наукових, фізико-технічних задач сучасності в e-sciences (фізики, математики та ін.), АСУ й в інших галузях промисловості , господарства та державних органів.

У роботі розглянуто базові основи індустрії програм, як механізмів фабрик програм, які зорієнтовані на індустріальне вирішення математичних задач наукового софтвера, що розробляються на кафедрах університетів та науково-дослідних інститутів України і зможуть бути використані при обчислюванні глобальних завдань. Наукові артефакти і програми, будуть накопичуватися на фабриках відповідно міжнародних стандартів їх специфікації, і стануть сховищами «заготовок» наукової продукції у галузі інформатики та Computer Science.

1 Сутність індустріального підходу

Термін індустрія визначає виробництво різних видів продуктів масового застосування і засобів їх вироблення промисловими підприємствами, фірмами та корпораціями, а в нашому випадку – програмно-технологічного типу.

Головним питанням деякої індустрії є не тільки випуск відповідної продукції, але й отримання прибутку від цього. Нині великі прибутки від випуску програмної продукції отримують такі світові фірми, як Microsoft, IBM, Sun, Unix, Intel та ін., а також індійські фірми по оновленню старіючих наслідуваних (legase) систем програм. В Україні за останні десятиріччя розвиток індустрії ПП фактично відсутній, немає державної і

наукової програми її підтримки. Одночасно діють невеличкі підприємства комерційного типу, які виконують роботи по виробленню не масових ПП на замовлення різних фірм, що фінансують їх.

У деяких наукових інститутах НАН України проводяться дослідження щодо принципів індустрії ПП, а в університетах ВНЗ цим теоретичним і прикладним аспектам індустрії ПП фактично не навчають (не має окремих навчальних курсів в Міносвіті) і тому можна констатувати, що наша держава відстає від світового рівня розвитку індустрії ПП більш ніж на 20 років. Тому дуже актуально розглядати всі нові підходи і технології з виробництва масових продуктів для вирішення завдань з інформатизації і комп'ютерізації.

Виробництво **ПП** виконується на технологічних лініях, за допомогою яких виконавці використовують відповідну теорію і інструментальні засоби підтримки для вироблення деякого ПП масового застосування. Ці лінії включають необхідні процеси життєвого циклу, орієнтовані на розроблення відповідного ПП. Прикладом є технології – Engineering application, family, domain, а також сформовані технологічні засоби оновлення застарілих систем в офшорних організаціях (наприклад, в Індії, Росії, Білорусії, України тощо),

Нині для підтримки процесів виготовлення ПП використовуються системні сервісні засоби (ОС, загальносистемні сервіси горизонтального і вертикального типу, нові мови, транслятори, компоузери тощо), а також готові програмні ресурси – ГОР, КПВ та методики проведення робіт з програмування та обчислювальня. Запропоновані нами *дисциплін програмної інженерії (ПІ),* які визначають різні аспекти діяльності по виробництву ПП, а саме наукові, інженерні, управлінські, економічні, виробничі. Вони потрібні при виробництві ПП й участі кваліфікованих фахових спеціалістів для виготовлення ПП і розробки нових засобів виробництва для фабрики програм [7–9].

В Україні нині діють окремі організації і фірми з розроблення ПП, які орієнтовані на випуск продуктів для окремих замовників, як правило, закордонних. Фірми з успадкування ПП, що покривають деякий фрагмент ринку ПП теж працюють в основному для закордонних держав.

Для першого попиту з індустрії запропоновано створити *студентську фабрику програм і артефактів* (Http://programsfactory.univ.kiev.ua) на факультеті кібернетики в КНУ імені Тараса Шевченка з готових артефактів, розроблених студентами на відповідних кафедрах, їх апробація на технологічних або продуктових лініях (Product Lines) для розповсюдження їх не тільки на вітчизняному науковому ринку, але з часом і закордонному.

Фабрика КНУ базується на лініях, обладнаних набором засобів, комплектуючих «деталей», інструментів і сервісів для автоматизованого виконання їх процесів у операційному середовищі.

З погляду інформаційних технологій фабрики дають набір інструментів для переходу від індустрії окремих ПП, до індустрії складних програмних систем з підвищенням продуктивності простих елементів продукту на кожному процесі життєвого циклу (ЖЦ). Лінії розробки простих продуктів на фабрики реалізовані в середовищі MS.Net з використанням готових КПВ, мови DSL (Domain Specific Language) й ін. Тобто, основу діяльності фабрики містять окремі лінії та готові КПВ і артефакти, що орієнтовані на конкретну предметну область і накопичені в репозитарію, для збирання з них, як на конвеєрі цільовитх ПС.

Фабрики програм. Під *фабрикою програм* розуміється інтегрована інфраструктура зі зборкою готових ресурсів у ПП, потрібних державним, науковим, комерційним й іншим замовникам. Фабрика обладнається продуктовими лініями, набором засобів, інструментів і сервісів для полу – чи автоматизованого виконання процесів на цих лініях в операційному середовищі. Фабрика софтвера – це погоджений набір процесів, засобів і інструментів для цілеспрямованого створення нових програмних об'єктів.

Фабрика базується на середовище, що містить автоматизовані засоби і інструменти виробництва ПП за відповідними лініями. З погляду інформаційних технологій фабрика дає набір інструментів для переходу від ремесла до індустрії ПП із метою збільшення продуктивності розробки продукту на кожному процесі ЖЦ лінії із заданими функціями, архітектурою і якістю.

Фабрики містять лінії й відповідний набір засобів розробки простих і складних ПП. Лінії розробки простих елементів ПП, як правило, відповідає відповідний ЖЦ, наприклад, реалізований в середовищі MS.Net. Лінія розробки складних ПП на фабрики розглядаються збирального типу із готових програмних ресурсів бібліотек чи репозиторієв глобального типу.

2 Базові основи індустрії програм

Виходячи з отриманої практики автоматизованого збирання різнорідних програм у мовах програмування (МП) в середовищі ОС ЄС [1] і аналізу сучасних закордонних фабрик програм з індустрії ПП (IBM, OMG, Microsoft, Oberon тощо) [3–8] нами сформовані загальні елементи індустрії виробництва програм, а саме:

– *готові програмні ресурси* (артефакти, програми, системи, reuses, assets, компоненти повторного використання – КПВ) тощо;

– *інтерфейс,* як специфікатор готових ресурсів, незалежно від мов програмування (МП), в мові інтерфейсу (IDL, API, SIDL, WSDL, RAS тощо) [6, 7];

– *технологічні лінії (ТЛ), продуктові лінії* (Product Lines) [1] з виробництва ПП;

– *зборочний конвеєр* (атоматизовані лінії) фабрики програм;

– *методики з прийомів* проведення робіт на лініях фабрики програм;

– *середовище* виробництва програм на фабрики.

Ці складові елементи сформульовані нами і розвиті в рамках фундаментальних проектів Інституту кібернетики (1980–1991) і ІПС НАНУ (1992–2010) [1, 8]. В них уперше визначено концепцію *інтерфейсу* (1982 р.) [10]), метод зборки різномовних програм, технологічні лінії (ТЛ, 1987–1991)) та засоби автоматизації випуску ПП [1–5]. Все це попередило появу індустріальних складових елементів з виробництва ПП закордонними фірмами, наприклад, IBM, Microsoft, Oberon та ін.

Далі дається формальне тлумачення наведених складових елементів індустрії ПП і обчислень.

2.1 Опис складових елементів

Готові програмні ресурси (ГОР). Програмні ресурси, які можуть використовуватися багаторазово, звуться reuse, assets, компоненти, КПВ, програми тощо. Вони відображають реалізацію різних прикладних або математичних функцій деякої наукової (фізика, математика, біологія тощо) або прикладної предметної області.

Головна парадигма КПВ – «писати один раз, виконувати багато разів, де завгодно». Архітектура, в яку вбудовується готовий ресурс, використовує стандартні механізми роботи з ним, як із «будівельними» блоками. Щоб забезпечити високу ефективність повторного використання КПВ, вони мають бути якісними і надійними при обчисленні [8].

Готові ресурси відображають деякі артефакти з діяльності розробників програм. *Артефакт* – це реальна порція інформації з програмування, яка може створюватися, змінюватися і використовуватися при виготовленні ПП, їм може бути:

– модель ПрО зі своїми термінами, поняттями та лексикою;

– готові КВП або окремі частини (фрагменти) елементи систем;

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

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

Артефакти незалежні від платформ комп'ютерів мають опис інтерфейсів і параметрів для взаємодії з іншими готовими ресурсами. Усі ресурси і їх інтерфейси зберігаються у сховищах (бібліотеки, репозиторії) для подальшого їх пошуку іншими фахівцями з метою подальшого використання. Повторне використання є капіталомістким видом діяльності з ПП на фабрики.

Інтерфейси ГОР. Під *інтерфейсом* розуміється зв'язок двох окремих сутностей. Інтерфейси бувають програмні, апаратні, мовні, користувальницькі, цифрові й т. п. Програмний (API) і/або апаратний інтерфейс (port) є способом перетворення вхідних/вихідних даних під час зв'язку комп'ютера з периферійним устаткуванням. У програмуванні інтерфейсом є програма або її частина, в якій визначаються дані (константи, змінні, параметри й структурні типи даних тощо) для передачі їх іншим програмам з завданням значень типів даних їх параметрів [6, 7].

У ролі інтерфейсу виступають оператори виклику до процедур і функцій у МП з завданням імен процедур або функцій і списку формальних параметрів з фактичними значеннями для виконання. Послідовність і число формальних параметрів відповідають фактичним параметрам. Коли програма, процедура або функція специфіковані різними МП і вони розташовані на різних комп'ютерах, то вирішується проблема неоднорідності поданих їх типів даних, відмінності архітектур комп'ютерів або операційних середовищ та несумісності параметрів за їх кількістю та порядком розташування.

Інтерфейсу фактично відповідає *посередник* або *перехідник* (stub, skeleton Corba) між двома модулями чи програмами. Він передає дані і виконує необхідне пряме й зворотне перетворення даних у випадку їх неоднорідності та невідповідності. Він визначається мовою IDL або APL для об'єкту-клієнта і об'єкту-сервера, має окрему реалізацію і доступний різномовним програмам. Інтерфейсний посередник включає опис формальних і фактичних параметрів програм, їх типів і порядок завдання операцій передачі параметрів і отримання результатів після виконання. Іншими словами, такий опис є не що інше, як специфікація інтерфейсу двох різномовних програм, які взаємодіють між собою через виклик, який реалізований практично в різних загальних системах (наприклад, COM, CORBA, JAVA тощо). У функції інтерфейсного посередника (stub у системі CORBA) для клієнта входить:

– підготовка зовнішніх параметрів клієнта для звернення до сервісу сервера;

– посилка параметрів серверу і його запуск для отримання результату або відомостей про помилки.

Загальні функції інтерфейсного посередника (skeleton) сервера:

– отримання повідомлення від клієнта, запуск видаленої процедури, обчислення результату і підготовка (кодування або перекодування) даних з формата клієнта;

– повернення результату клієнтові через параметри повідомлення з результату виконання) та ін.

Таким чином, інтерфейсні посередники задають зв'язок між клієнтом і сервером (*stub – для клієнта і skeleton* – для сервера), як це зроблено в системі CORBA.

Сучасні підходи до реалізації інтерфейсу. До підходів з вирішення проблеми інтерфейсу належать наступні:

1) зв'язок готових, різнорідних програм за допомогою інтерфейсу IDL, в якому визначені вхідні й вихідні дані цих програм;

2) спеціальний інтерфейс – JNI (Java Native Interface), що допускає звернення з Java-класів до функцій і бібліотек на інших МП з пошуком прототипів звернень до функцій на C/C++, генерацією заголовкових файлів компіляторами C/C++ і звернення з Java-класів до COM-компонентам;

3) технологія Bridge2Java, що генерує оболонку для COM-компонентів у вигляді проксі-класа і забезпечує необхідне перетворення даних для різних МП засобами стандартної бібліотеки перетворень типів;

4) зв'язок за допомогою мови CLR(Common Language Runtime) платформи .Net для будь-яких її МП, в якій транслюються об'єкти в ЯП (C#, Visual Basic, C++, Jscript) з використанням бібліотеки стандартних класів і засобів генерації в представлення .Net-компонентів;

5) стандартне рішення ISO/IEC 11404–1996 щодо специфікації типів даних незалежно від МП за допомогою мови LI (Language Independent) для різномовних компонентів, що містить усі існуючі типи даних МП або засоби їх конструювання. У мові LI описуються параметри виклику, як елементи інтерфейсу, вони перетворюються у типи даних конкретних МП спеціальними правилами і операціями генерації складних типів даних цього стандарту до більш простих фундаментального типу;

6) XDL-стандарт опису структур даних довільної складності і перетворення форматів даних, що передаються з однієї платформи комп'ютера на іншу за допомогою спеціальних процедур та як стандарт забезпечення взаємозв'язків і перетворення типів даних ЯП, зрозумілого багатьом розподіленим середовищам.

Рішення по конкретному перетворенню даних цими варіантами не вичерпуються, вони ще з'являтимуться при впровадженні нових платформ комп'ютерів і середовищ гетерогенного типу, особливо Cloud Computing.

Стандарту WSDL пропонує специфікувати інтерфейс програмних об’єктів (як паспорт) виду:

– *назва* інтерфейсу;
– *ID* – ідентифікатор ресурсу;
– *зміст* програми (функції) з інтерфейсом;
– *параметри* виклику інших програм;
– *інструменти* підтримки виконання програми тощо;
– *необов'язкові атрибути* (дата, стан, версія, право доступу, автор, дата створення, термін придатності, правила придбання і т. п.).

Специфікації інтерфейсу зберігаються у бібліотеки інтерфейсів для майбутньої зборки відповідних об’єктів. Для включення студентських програмних артефактів розроблено шаблон специфікації інтерфейсів за стандартом WSDL, названі позиції якого заповнюють розробники програм.

Прикладом розвитку різних типів взаємодії різнорідних програм є керівництво І. Беєві («Взаимодействие разноязыковых программ», 2005), представив більше 100 спроектованих варіантів модулів-посередників для програм в класі сучасних мов : C/C++, Visual C++, Visual Basic, Matlab, Smalltalk, Lava, LabView, Perl їх різних предметних областей. Варіанти інтерфейсів практично перевірені ним у відповідних середовищах функціонування. Один з його прикладу використано нами для фабрики програм.

Нові сучасні підходи автоматизації взаємодії різномовних програм реалізовані нами у новому інструментально-технологічному комплексі (ІТК), як веб-сайту для навчання фундаментальним аспектам з програмної інженерії (http://sestudy.edu-ua.net).

Роль цих концепцій інтерфейсу для взаємодії КПВ і шляхів їх реалізації зростає завдяки накопиченню величезного запасу цих КПВ у світі і інтерфейсів для різних предметних областей, використання яких без визначення їх інтерфейсів, принципів їх взаємодії і засобів їх підтримки не можливо.

2.2 Технологічні і продуктові лінії

Технологічна лінія – ТЛ будується з процесів ЖЦ після аналізу ПрО й виявлення її основних задач і функцій. Для процесів підбираються готові ресурси, а також засоби й інструменти породження функцій в програми. Крім того додається планування, контроль й керування процесами ТЛ з формуванням шаблонів і заготовок для фіксації проектних рішень, зміни станів і оцінювання якості ПП [2, 4].

Набір процесів ТЛ будується з урахуванням міжнародних стандартів ISO/IEC 12207– 2007 і ДСТУ 9126–99, що підкріплюються відібраними методами, засобами й інструментами для здійснення змін станів проміжних об'єктів. Цей підхід до побудови ТЛ апробований в АІС "Юпітера" (1987–1991 рр.) на прикладі побудованих шості ТЛ обробки даних.

На новому витку історії розвитку ліній індустрії програм виникли поняття продуктових *ліній* в SEI США (SEI), принципи побудови й виконання яких аналогічні ТЛ і відображають індустрію виробництва систем, сімейств систем комерційного типу.

Поняття *продуктової лінії (*product line (лінія продуктів) і product family (сімейство продуктів, надалі СПС) визначені у словнику ISO/IEC FDIS 24765:2009(E) – Systems and Software Engineering Vocabulary як «група продуктів або послуг, які мають спільну керовану множину властивостей, що задовольняють потреби

певного сегмента ринку або виду діяльності» («*product line* – group of products or services sharing a common, managed set of features that satisfy specific needs of a selected market or mission. Synonym: *product family*»).

Тобто с*ім'я продуктів* або *лінія продуктів* для деякої ПрО – це множина ПС, розроблених з використанням набору ГОР, які мають спільний набір властивостей і рис для деякої ПрО. Під *ресурсами* розуміються артефакти, КПВ, проектні рішення, коди, вимоги тощо. Кожна лінія ініціює первинний процес або зборочний конвеєр з виро ПП із ГОР з метою задовольняння потреб деякого ринка ПП. Наведені типи ліній виробництва ПП не обмежені. Деяка фабрика може бути зорієнтована на спеціальні лінії функціонального типу, наприклад, на створення програм із класу задач статистичної обробки, чисельних методів тощо.

Інженерія продуктових ліній. Створення *ліній виробництва продуктів,* запропоноване інститутом SEI з систем, каркасів, готових програм і КПВ, з яких формується кінцевий продукт, що задовольняє певним потребам ринку програмної продукції [10, 12].

Поняття фреймворка для лінії програмних продуктів (Framework for Product Line Practice) сформувалося як деяка автоматизована реалізація інженерії ПрО, в завдання якої входить побудова різних видів програмних продуктів для ринку за допомогою методів і засобів ліній виробництва продуктів.

Для побудови ліній виробництва вони досліджували ринок і потреби майбутніх покупців, створювали виробничий план, визначали процеси та організацію їх виконання. На основі аналізу потреб ринку і інтересу до певного виду продукту вони побудували лінію продукції, в яку включаються необхідні методи розробки, тестування і оцінки процесів, продуктів ліній.

У інфраструктуру розробки лінії продуктів, окрім необхідних методів і засобів побудови, експлуатації ліній продуктів, входять матеріали і методики по керівництву. Хоча лінії і сімейство продуктів ототожнюють, лінії продуктів (на ринок) може бути побудована на базі певного представника сімейства продуктів (що розробляється, наприклад, для певного замовлення).

При побудові конкретної лінії для деякого представника сімейства у ПрО визначаються:

– технологічні і виробничі обмеження, властиві продуктової лінії;

– зразки і каркаси, які можуть використовуватися на лінії;

– набір засобів, методів і інструментів для розробки продукту на лінії.

Ці лінії ґрунтуються на використанні КПВ, які відбираються з з репозиторію Про. Вони підтримують процес складання готових функцій, компонентів і КПВ в єдиний продукт на лінії виробництва.

Нами розроблені три базові лінії загального призначення для фабрик програм КНУ [2].

Перша лінія для побудови деякого програмного ресурсу шляхом:

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

– специфікація цих елементів мовами програмування (МП) і паспортних даних інтерфейсів;

– зберігання їх у репозиторію.

Друга лінія – це механізми підбору готових ресурсів із репозиторію з метою їх застосування у відповідної предметної області

Третя лінія – це лінія об єднання (зборка) різних КПВ, що забезпечує конструювання нових ПП методом зборки з готових підібраних ресурсів. Ця лінія буде давати прибуток за рахунок заощадження трудовитрат від застосування готових артефактів та КПВ за оцінкою ефективності їх застосування і обсягів повернення цього вкладення в ПП. ГОР може бути науковим, прикладним і загальносистемним. Перший ресурс – це метод, алгоритм як артефакт (з математики, фізики, біології тощо), що описаний загальною МП, другій ресурс – це реалізація окремій задачі або функції ПрО (бізнесу, комерції, економіки і т. п.) і загальносистемний ресурс – це готові засоби середовища виконання ліній (транслятор, редактор текстів, генератор, інтегратор, завантажувач, сервіс тощо). Зміст даної лінії відповідає лінії зборки програм.

Оскільки кожна лінія ПП складається з процесів, які виконуються за участю одного чи більше розробників, кожний з них завдає необхідні вхідні дані для процесу виробництва програм (наприклад, готовий КПВ при зборці його з іншими).

2.3 Середовище виробництва програм на фабрики

Середовище виробництва – це засоби, інструменти і методики забезпечення роботи фабрики програм при виробництві ПП методом зборки, функції яких забезпечують їх віртуалізацію, синхронізацію, виконання та підтримку продуктових та технологічних ліній з виробництва різнорідних програм в МП.

Засоби – це різні МП, які використовуються для опису програм і інструменти, типу транслятори, редактори обов'язково будуть на фабрики, як необхідний засіб виробництва програм.

Інструменти – це системні, прикладні програмні компоненти (Eclipse, Protégé, Eclipse-DSL тощо) підтримки процесів побудови прикладних ПС чи СПС та сучасні фреймворки (Jaspect, Jbeans, Ant, WCF, Web-service, Amazon, Sky-Driven тощо), які мають комплекс засобів для побудови окремих програмних об'єктів і даних в технології виробництва окремих особливих елементів, та загальносистемні засоби (VS.Net, Corba, Java,

IBM Vsphere тощо), які самі як фабрики з великим набором різних Tools. І засобів підтримують командну розробку складних програм і проектів.

Методики розроблення і виконання процесів виробництва ПП включає документацію і інструкції по організації ТЛ і ТП та принципам проектування на них різних окремих елементів. Сюди відноситься нормативні і регламентовані методи виконання різних робіт з розробки, зборки, керування експертизами, вимірами і оцінками артефактів..

Таким чином, ці базові елементи індустрії програм створюють процес виробництва ПП..

3 Індустрія обчислень програм і даних

Нові індустріальні підходи до обчислення програм ы даних з'явилися у зв'язку з бурним розвитком високоефективної елементної бази, нової багатоядерної, процесорної конфігурації комп'ютерів тощо. Це сприяє великомасштабним обчисленням дуже складних задач сучасності в e-sciences (геології, біології, фізики, математики та ін.), АСУ й інших галузях промисловості. Комп'ютерна індустрія більш ніж на порядок опереджає розвиток як з теоретичної, так і з практичної точки зору інші види індустрії, а саме, індустрію обчислень систем і програм [3–5].

Індустрія обчислень у теоретичному плані іде по шляху застосування нових комп'ютерних можливостей щодо швидкості дій, розподілення пам'яті для обчислення задач з великими обсягами даних та інтероперабелності [6–8].

Індустрія обчислення програм у Grid. Вона отримала також новітні базові засоби з організації прозорих обчислень різного роду складних задач за рахунок розвитку нових моделей: *взаємодії* OSI, *генерації* GDM, *архітектури* SOA,MDA, *розробки* DDM, *хмарних обчислень* (Cloud Computing), що підтримуються Веб–стандартом HTML5 з високо пропускними протоколами доступу до загальних on-line сховищ даних з деякого місця нашої планети.

Одним з практичних рішень забезпечення *індустрії обчислень* є поява системи Grid (2005 р.) в межах Європейського проекту. Цю систему було розроблено як інфраструктуру для глобальних обчислень суперскладних задач з області e–science. Для обчислень таких задач застосовуються знов розроблені або готові програмні і системні ресурси виду: розподілені процесорні потужності; системи сховищ даних; новітні засоби комунікації; фонди reuses з багатьох доменів; нові технічні і загальні «тули» (конвертори, генератори, трансформатори, тощо); моделі та схеми взаємодії програм у середовищі гетерогенних платформ, географічно розташованих у віддалених адміністративних доменах тощо.

Cloud Computing чи *хмарні обчислення* - це нові системні засоби для підтримки обчислень, якими є Google Apps, IBM-VSphere та системами Microsoft – WCloud, Azure, Amazon, Mech, WApps, SkyDriven тощо. Функції деяких систем розглядаються далі. Google Apps забезпечує обчислення бізнесу в режимі on-line за допомогою Інтернет-браузера до великих обсягів даних на серверах Google, виконує веб-застосування з сховищами даних через інтерфейс API .в мовах Java і Python за допомогою стандартних протоколів [3, 4].

Amazon Machine Image (AMI) підтримує застосування, бібліотеки за інструментами зберігання AMI і образів у сховищі, вибір ОС для запуску і контролю декількох AMI з використанням Веб-сервісу, інструментів управління та платою за надані для вживання ресурси, такі як час за кількість переданих даних. Azure (Windows Cloud, 2008) для створення розподілених «хмарних» веб-застосувань і Інтернет-сервісів з використанням технології .Net.

Amazon Web Services — це інфраструктура Web Services для надання таких послуг: зберігання даних (файловий хостінг, розподілені сховища даних), надання обчислювальних потужностей тощо. При чому, час обчислення 12 центів, гігабайт даних на сервері – 15 центів, кожні 10 тисяч транзакцій – 10 центів.

Ці системи зорієнтовані на збереження даних, доступ до глобальних сховищ даних on-line, синхронізацію даних великих розмірів і викладання їх на віддалені сервери тощо. Тут головна проблема організації обчислень потребує визначення нових методів, таких як координація, кооперація та взаємодія різних сервісів і інших готових ресурсів через конфігураційний файл для виконання обчислень відповідних задач. Накопичення готових ресурсів у різних глобальних сховищах потребує розвитку індустріальних методів і моделей з урахуванням специфіки наукових задач, готових звичайних компонентах (артефактах, reuses, assets і даних) багаторазового використання, що знаходяться у різних бібліотеках, репозиторіях та нових «хмарних» сховищ Інтернету.

Головним методом індустрії наукових продуктів є удосконалений метод зборки наукових різнорідних ресурсів, що належить до напрямків e–science в структури ПП, після виготовлення вони забезпечуються моделлю взаємодії для організації обчислень на різних глобальних серверах. Запити до виконання використовується віртуальний сервер з метою віртуалізації і балансування навантаженнями засобами розподілу віртуальних задач по організації обчислень на реальних платформах і великих комп'ютерах чи кластерах.

Індустрія даних Сучасний банк сервісів для збереження даних on-line за останні п'ять років є прозорим і вбудований в продукти IBM, Microsoft що забезпечують синхронне подання даних і передачі даних на віддаленні сервери. Сервіси взаємодіють зі хмарними сховищами для отримання необхідних даних,

виконувати обчислення і створювати резервні копії даних. Поряд з таким механізмом доступу до хмарних даних з'являються нові безпровідної технології типу WiMAX подібно безпровідному Інтернету.

Сервіс даних on-line сховищ включає:

– синхронізацію даних засобами Windows Live Mech, SkyDriven тощо;

– передачу даних на віддаленні сервери і резервування різних копій даних (Office WebApps);

– використання нових веб-технології (Flash, Silverlight, Amazon);

– компоузер завдань, робота з БД (Windows Azure, MS SQL, Services, Live Services);

– бібліотеки примітивів з перетворення типів даних GDT<=>FDT із бібліотек.

Бібліотека примітивів для перетворення типів даних GDT (примітивних, агрегатних і генерованих) до FDT типів даних (простих, структурних і складних) МП використовується при передачі даних між різномовними компонентами, підсистемами і проектами. Бібліотека складних типів даних GDT (контракт, портфель тощо) використовується для генерації простих типів даних FDT і накопичення у базі даних. Бібліотека функцій для перебудови форматів не релевантних даних інтерфейсних посередників (stub, skeleton), що передають їх на інші платформи взаємодіючим компонентам і зворотно. Вона описана [12].

Організація обчислень. Для функціонування різних інструментів і засобів фабрики використовуються моделі організації зв'язків між компонентами ї [3, 4, 13]. До них відносяться такі:

1) брокер запитів між програмами клієнта і сервера з отримання даних від stub клієнта, їх обробку під формати даних сервера складених через skeleton;

2) проміжний прошарок загальносистемних середовищ підтримки процесів виконання або взаємодії об'єднаних різнорідних програм;

3) пряма зборка транслюваних програм за різними МП і інтерфейсами з бібліотеки MS.Net.

Перша модель взаємодії практично реалізується брокером ORB для класу МП у середовищі CORBA. Зборка компонентів базується на класах: Client class з викликами інших, Stub class з конвертування даних, ORB class з передачі даних за викликами; Server class зі створення сервентів та посиланням даних ORB; Skeleton class з конвертування форматів даних та адаптера для породження різних видів серверів.

Друга модель забезпечує взаємодію між компонентами, розміщеними на різних комп'ютерах, через проміжний прошарок (middleware), та неоднорідність їхніх форматів даних шляхом повідомлень (наприклад, Java Message Queue), віддаленого виклику процедур RPC в MS Windows, ONS IBM, CORBA та виклику методів RMI у Java. Модель ISO/OSI також забезпечує проміжну взаємодію на рівнях (фізичному, канальному, мережному і транспортному) через мережну ОС. Верхній рівень моделі (прикладний) забезпечує перетворення даних до транспортного рівня шляхом їх маршалінгу й демаршалінгу за інтерфейсним stub. Сеансовій рівень моделі передає дані через транспортний канал за механізми посилань іншим рівням.

Третя модель реалізована в MS.Net і базується на бібліотеках: CLR (Common Language Runtime), CTS (Common Type System) і CLS (Common Language Specification). CLR забезпечує виявлення і завантаження типів даних, керування ними у відповідності до завдань розробника програми. CTS визначає принципи взаємодії із іншими, відповідно представлених форм метаданих. Зборка різномовних програм виконується за правилами CLS бібліотеки. Усі компілятори з МП цього середовища створюють модуль DLL або EXE у проміжну мову MSIL (Microsoft Intermediate Language) для зборки отриманого коду, незалежно від платформи, на якій він буде виконуватися рішення. При запуску цей код конвертується в специфічний код CPU івиконується на різних архітектурах комп'ютерів.

Для побудови програм для студентської фабрики програм використовуються Visual Studio, Веб-сервіси різного призначення, пакети VSTS для керування конвеєрним методом зборки та засобів вимірювання і оцінювання якості створених програмних продуктів.

Таким чином**,** в залежності від цілей фабрики програм в якості середовища вибирається те середовище, що найбільш підходить для організації процесу побудови і зборки відповідних програмних ресурсів у межах деякої ПрО. Коло проблем зменшується, коли різні середовища взаємодіють між собою, створюючи одне велике середовище з поширеними можливостями щодо виробництва кінцевого об'єднаного продукту.

Новим способом взаємодії між програмами типу клієнт і сервер забезпечується інтерфейсом *Icontract* у системі WCF (www. wcf.org), який призначений для опису атрибутів та операцій з передачі даних від сервісного об'єкта (*Service consumer*) до клієнтського (*Service provider*). Передача інтерфейсів виконується за контрактами:

– *сервісів* опису операцій з виклику клієнта і *передачі даних* за типами даних (*int, float, string* тощо) між клієнтом і сервером;

– *повідомлень, як* механізми забезпечення взаємодії об'єктів;

– *помилок*, що утримуються сервісною службою для передачі їх клієнтам.

Повідомлення специфікується мовою XML і подається протоколом SOAP. При взаємодії середовищ через протоколи WCF виникають конфлікти (див. е-сайт wcf), які пов'язані з несумісністю типів даних, переданих в інтерфейсах клієнтів та форматами архітектури платформ об'єктів клієнта і сервера, які реалізуються за допомогою запропонованої бібліотеки примітивів [12].

На сайті http://sestudy.edu-ua.com проведено первінну реалізацію наступні моделі взаємодіїї.

Модель взаємодії програм – це схема зв'язків окремих частин програм між собою. Як зв'язки виступають оператори звернення (типу CALL, RPC й ін.) до процедур і функцій програм з формальними параметрами, які записуються в одній МП [4, 13, 14].

Моделі взаємодії систем найбільш пов'язані з використанням готових програм у процесі розроблення нових ПС методом зборки. Програми подані в різних МП і розташовані на різних комп'ютерах мережі і їх збирання пов'язано з вирішенням проблеми взаємодії, пов'язаних з неоднорідністю гетерогенних типів даних, структур пам'яті комп'ютерів і середовищ, де вони виконуються. В принципі зібрані КПВ враховують формати даних програм з платформ сучасних комп'ютерів, структури вихідного коду програм після компіляторів з МП з використанням примітивів з спеціальних бібліотек *data types* і *routines*. Реально існуючі розходження в апаратній частині платформ і у вихідному коді програм відмічається у конфігураційному файлі ПП, який використовується при організації його виконання у сучасних обчислювальних або гетерогенних середовищах. [4, 13, 14].

Модель взаємодії середовищ – *це* послідовність дій з перенесення програми, створених у одному середовищі в інше для їх виконання. Деякі середовища, наприклад, VS.Net і Java вже мають безпосередні зв'язки і створюють одне інтегроване середовище. В межах фундаментального проекту проведено реалізацію моделей взаємодії пар наступних операційних середовищ Visual Studio ↔ Eclipse та CORBA↔ Visual Studio за участю студентів КНУ імені Тараса Шевченка та МФТІ [8, 13, 14]. Кожне з середовищ базуються на своїх інтерфейсах взаємодії і включає загальні методи і засоби доступу до даних мережного середовища. Практичні приклади пар взаємодії середовищ: – *Visual Studio.Net, Eclipse,– Visual Studio.Net, Eclipse Corba, Java, MS.Net, IBM VSphere, МП, Eclipse* розглянути на вказаному сайті ІТК на прикладі програм в цих МП. Тут Eclipse виконує функцію адміністратора репозиторію по збереженню, відбору і застосуванню готових програмних ресурсів типу КПВ, а також для виконання ПП, виготовлених у цих середовищах.

Тобто при переході в середовище Eclipse використовуються вихідні файли програми, *dll*–бібліотки VS та файли ресурсів (*.resx*). Коли необхідно знов перейти із цього середовища в Visual Studio, весь проект імпортується туди. Обчислення програми та її зміни можна виконувати як у середовищі Eclipse, так і у Visual Studio. Модель IBM Eclipse буде продовжено надалі при використанні сервісів для взаємодії програм і проведення обчислень за ними у розширеному середовищі.

Реалізації ідеї індустрії. Виходячи з розглянутих концепцій і теорії індустрії, а також викладання лекцій автором в КНУ імені Тараса Шевченка по курсам «Технологія програмування» і «Програмна інженерія» студентами була підтримана ідея створення експериментальної фабрики програм з декількома лініями по поданню наукових артефактів, що розробляються як нові артефакти і програми, які кращі і їх можливо стандартизовано оформляти і накопичувати у репозиторії фабрики програм (http://programsfactory.univ.kiev.ua). Крім того, за участю студентів підготовлені курси навчання з дисциплін за такими технологічними лініями:

– розроблення засобами C# VS.Net (опис цієї технології на наведеному сайті фабрики) консольних програм на мові C#, бібліотек компонентів DLL, локальних Windows-застосувань;

– розроблення програм мовою Java (самовчитель І.Ш.Хабібуліна);

– електронне навчання курсу "Програмна інженерія " (SE) за підручниками на сайті КНУ фабрики програм (укр.), також раніше створеного цього підручника російською мовою на сайті Інтернету (www.intuit.ru).

На фабрики використано ліцензійну в КНУ VS.Net як фундамент функціонування фабрики програм і можливості платформи MS.Net щодо *багатомовної розробки* ПС із компонентів в МП (C#, C++, VBasic тощо) та засобів підтримки колективного виробництва програмних проектів. Як результат, зовнішній розробник програм для фабрики не обмежується вибором однієї якої-небудь МП, а може в межах однієї ПС використовувати різні МП використовуючи сайт фабрики програм [9, 13, 14].

Лінії навчання на фабриках програм застосовуються студентами і підвищують рівень підготовки IT-спеціалістів та навчання співробітників ІПС НАНУ за допомогою ІТК [8, 9].

4 Висновки

Визначені базові основи індустрії виробництва програм із КПВ. Запропоновані головні елементи індустрії програм, обчислень і даних. Описано теорію індустрії програмних продуктів та наведені практичні варіанти моделі взаємодії складних систему сучасних гетерогенних середовищах.

Сформульовані нові підходи до створення технологічних ліній електронного навчання студентів на фабрики програм в КНУ та розроблення програмних артефактів засобами Visual Studio.Net, технології яких реалізовані в ІТК з 15 простішими лініями та лініями взаємодії програм в сучасних розподілених системах (VS.Net, Corba, Java, Protégé, Eclipse) та обчислень у середовищах хмарних обчислень на прикладі обчислювальної геометрії і задач перебудови релевантних типів даних. Ці засоби використовують студенти університетів Україні та СНГ за спеціальністю «програмна інженерія», інформатика і Computer Sciences.

Література

1. *Лаврищева Е.М.* Становление и развитие модульно-компонентной инженерии программирования в Украине. – Київ, 2008. – 33С. – (Препр. Ин-т кибернетики им. В.М. Глушкова; 2008–1).
2. *Андон П.І., Лавріщева К.М.* Розвиток фабрик програм в інформаційному світі // Вісник НАН України. – 2010. – № 10. – С. 15–41.
3. *Лаврищева Е.М.* Концепція індустрії наукового софтвера і підхід до обчислення наукових задач // Проблеми програмування. – 2011. – № 1. – С. 3–17.
4. *Лаврішева К.*М. Взаємодія програм, систем й операційних середовищ // Проблеми програмування. – 2011. – №3. – С. 13–24.
5. *Лаврищева Е.М., Грищенко В.Н.* Сборочное программирование. Основы индустрии программных продуктов. – Київ: Наук. Думка, 2009.– 371 с.
6. *Лаврищева Е.М.* Интерфейс в программировании // Проблеми програмування. – 2007. – № 2. – С. 126–139.
7. *Лаврищева Е.М.* Проблема интероперабельности разнородных объектов, компонентов и систем. Подходы к ее решению // Матер. 7 міжнар. конф. з програмування "УкрПрог –2008"– С. 28–41.
8. *Лавріщева К.М., Коваль Г.І., Бабенко Л.П., Слабоспицька О.О., Ігнатенко П.П.*. Нові теоретичні засади технології виробництва сімейств програмних систем у контексті ГП .– Електронна монографія, ДРНТІ.– № 67–УК–2011 від 05.10.11. – 377 с.
9. *Лавріщева К.М.* Інструментально-технологічний комплекс для виробництва програмних систем.– Вісник НАН, 2012.– № 3.– С. 19–23.
10. *Лавріщева К.М.* Програмна інженерія.– Підручник.– Академперіодика. – 2008. – 319 с.
11. *Лаврищева Е.М.* Классификация дисциплин программной инженерии // Кибернетика и системный анализ. – 2008. – № 6. – С. 3–9.
12. *Лавріщева К., Стеняшин А.* Підхід щодо трансформації загальних типів даних стандарту ISO /IEC 11404 для використання в гетерогенних середовищах//2nd Int. Conf. on HPC, October 8– 10, 2012, Kyiv.– Ukraine.– с.227– 234.
13. *Островський А.И.* Подход к обеспечению взаимодействия программных сред JAVA и Ms.Net.– Проблеми програмування, 2011.–№2.–с.37–44.

Industrial approach of application systems development and implementation in the heterogeneous environments

Lavrischeva K.M., Stenyashin A.U

Abstract. *The description of approaches is given to the development of industry of programs production from the components of the repeated use (reuses). The theoretical and practical aspects of programs industry are offered, calculations and information. An approach to realization according to principle of co-operation of the programs and systems in modern operating environments (Corba, VS.Net, Java, Eclipse and others) is considered. The newest conceptions of technology of industrial production of the difficult programs, after which the first student factory of the programs is created in the Kiev an national university for the accumulation of knowledge's about scientific artefacts' and programs which are developed in diploma and master's degree works after scientific disciplines of studies, are formulated. On factories three technological lines operate in relation to presentation of reuses, programming their languages of VS.Net and study of bases of the software engineering after the textbook. To the factory 5000 users and students of other universities of higher of Ukraine and SNG appealed more.*