

СТВОРЕННЯ СЕРЕДОВИЩА ВИКОНАННЯ НАУКОВИХ ДОДАТКІВ У ГРІД З ВИКОРИСТАННЯМ ІНСТРУМЕНТАРІЇВ GRIDSHERE ТА VINE TOOLKIT ЯК ЧАСТИНИ КОМПЛЕКСНОГО НАУКОВОГО ШЛЮЗУ

О.М. РОМАНЮК, А.С. САПГОРА, С.Я. СВИСТУНОВ, О.І. МАРЧЕНКО

Запропоновано метод реалізації виконання наукових додатків у Грід-середовищі в межах комплексного наукового шлюзу, який побудовано з використанням GridSphere та Vine Toolkit. Наведено модель архітектури такого наукового шлюзу. Розглянуто особливості використання технології Flex та процес побудови веб-додатків для визначеного наукового шлюзу на прикладі створення тестового портлету.

ВСТУП

Для проведення досліджень у будь-якій області науки, учені використовують спеціальні програмні додатки для наукових обчислень. На сьогодні існує достатня кількість програмних продуктів, яка здатна задовольнити ці потреби. Вони застосовуються лише в межах локального середовища і відсутня можливість використовувати розподілені обчислювальні ресурси. Більшість програмних пакетів мають досить складну функціональність, яку іноді неможливо подати у вигляді звичайної Грід-задачі [1]. З іншого боку, навіть виконання звичайної задачі в Грід вимагає від користувача знань основних принципів функціонування Грід-системи та інтерфейсу його проміжного програмного забезпечення (ППЗ). Окрім цього, використання додатка в такий спосіб не залишає можливості застосування графічного інтерфейсу додатка.

Для ефективного використання Грід-ресурсів потрібні інструментальні засоби, які здатні перетворити процес виконання додатка в спосіб, прийнятний для виконання в Грід-середовищі та зробити процедуру виконання додатка в Грід максимально простою та зручною.

У роботі [1] визначено термін «комплексний науковий шлюз» для характеристики операційного середовища, серед функцій якого є організація виконання наукових додатків у Грід-інфраструктурах та надання науковим додаткам зручних користувацьких інтерфейсів, які адаптовані до такого способу їх використання.

У цій роботі запропоновано варіант реалізації комплексного наукового шлюзу на основі інструментального засобу GridSphere та Vine Toolkit із використанням технології Flex для створення спеціального середовища виконання наукових додатків у Грід-середовищі під управлінням ППЗ gLite.

Мета роботи — розробка методів ефективного використання наукових додатків у Грід-середовищі в межах комплексного наукового шлюзу, побудованого за допомогою інструментаріїв GridSphere та Vine Toolkit.

ГРІД-ПОРТАЛ ЯК ОСНОВА НАУКОВОГО ШЛЮЗУ

Переважає більшість наукових шлюзів будується на основі Грід-порталів, які дозволяють організувати доступ до Грід-сервісів та Грід-ресурсів. Грід-портал визначається як клас серверних Інтернет-додатків, який забезпечують захищений інструментарій для on-line збору інформації про Грід-сервіси та Грід-ресурси, і який надає інструментальні засоби для використання цих Грід-сервісів і Грід-ресурсів для цілей виконання користувацьких завдань [2].

Організація Grid Computing Environment Research Group поділяє портали на дві групи: користувацькі портали і портали застосувань [3]. Перша група орієнтована на забезпечення доступу користувачів, об'єднаних у віртуальні організації для розв'язання спільних наукових завдань, які потребують значних обчислювальних ресурсів. При цьому цей доступ реалізується у вигляді заміни стандартного доступу до ресурсу за допомогою термінальної системи на систему доступу через портал. Другу групу порталів орієнтовано на забезпечення доступу не тільки до Грід-ресурсів, але й для полегшення підготовки даних для інженерних пакетів і використання цих пакетів у Грід-середовищі.

Прикладами вдалого використання портальних рішень для виконання наукових розрахунків можуть бути портали: Cactus [2], Astrophysics Simulation Collaboratory (ASC) [4], PACI HotPage [5].

Однак вже під час створення перших портальних систем розробники зрозуміли, що для полегшення програмування порталу та стандартизації роботи з функціями доступу до Грід-ресурсів необхідно користуватись набором відповідних інструментальних засобів. Використання таких інструментальних засобів дозволяє розробникам порталів сконцентрувати свої зусилля на розробці прикладних портальних додатків і на реалізації логіки використання інженерних пакетів, відокремлюючи функції взаємодії з ППЗ Грід в окремі стандартизовані бібліотеки.

ІНСТРУМЕНТАЛЬНІ ЗАСОБИ РОЗРОБКИ ГРІД-ПОРТАЛІВ

На початковому етапі інструментальні засоби являли собою прості API для безпосереднього використання функцій і сервісів Globus [6] з мов програмування таких як Perl, Python та Java. Найвдаліші розробки: Commodity Grid Kits [7], Grid Portal Development Toolkit (GPDK) [8], Open Grid Computing Environment [9], myGrid [10]. Загальними для всіх цих інструментальних засобів є те, що розроблені портали передбачають інтерактивну взаємодію безпосередньо із Грід-сервісами. Розробники таких порталів мають бути фахівцями з Грід-технологій і у веб-програмуванні.

Серед програмних інструментальних засобів відмітимо дві розробки, які надають можливість користувачам динамічно генерувати портали додатків із наступною публікацією без втручання у створення програмного коду: інструментальний засіб GridSpeed [11, 12] — розробка Tokyo Institute of Technology, TiTech та Grid Programming Environment [13] компанії Intel, призначений для створення універсального інтерфейсу доступу до різних

реалізацій ППЗ Грід. Ці інструментарії дозволяють користувачам і розробникам одержати прозорий доступ до розподілених обчислювальних ресурсів і ресурсів зберігання даних за допомогою технології веб-сервісів та дозволяють уникнути труднощів під час розробки програмного коду додатка.

Прийняття рішення про вибір певної технології не є тривіальним. У процесі вибору враховувались зусилля, які необхідні для встановлення, налаштування, обслуговування та підтримки програмного забезпечення (ПЗ). До цього ж мотивація вибору інструментарію була зумовлена заздалегідь через використання ППЗ gLite на Грід-ресурсах, з якими має взаємодіяти вибраний інструментарій. Виходячи з проведеного аналізу, а також оцінюючи функціональність інструментаріїв, в якості основного інструментарію було прийнято портальну систему на базі GridSphere.

ІНСТРУМЕНТАЛЬНИЙ ЗАСІБ GRIDSHERE

GridSphere було розроблено в 2002 р. в межах проекту GridLab, який фінансувався ЄС. Головною метою розробки порталу було створення надійного веб-інтерфейсу для європейських і світових користувачів Грід. Почавши зі специфікацій API, багато в чому ідентичних IBM Websphere® 4.2, проект згодом еволюціонував до порталу-контейнера, сумісного з JSR 168, який підтримує завантаження та взаємодію портлетів.

Інструментарій GridSphere 3.1 являє собою каркас веб-порталу з відкритим кодом для розробки модульних веб-компонентів, які називаються портлетами. Портлети — це візуальні компоненти, що можуть бути вбудовані у веб-сторінки порталу. Портлети містять «міні-додатки», які можуть відображати інформаційний контент або надавати доступ до різних сервісів. GridSphere реалізує безпосередньо контейнер портлету, набір основних портлетів і розширену бібліотеку інтерфейсу користувача, використання якої

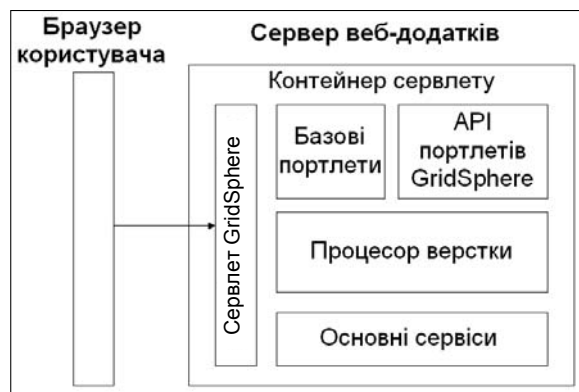


Рис. 1. Архітектура порталу GridSphere

спрощує процес розробки нових портлетів [14]. Архітектуру GridSphere зображено на рис. 1.

Схема архітектури містить дані щодо основних компонентів, які в поєднанні з контейнером сервлетів складають сервер веб-додатків. Запит веб-браузера користувача до порталу викликає сервлет GridSphere, який виступає в якості контрольного диспетчера процесору верстки, що відповідає за генерацію вихідних даних для браузера користувача. Сервлет GridSphere та процесор верстки використовують основні сервіси, включаючи реєстр портлетів, для відображення набору відповідних портлетів на портальній сторінці користувача.

Інструментарій GridSphere [15] включає такі основні портлети: для авторизації користувача; налаштувань; персоналізації профілю та зовнішнього вигляду; адміністрування (створення користувачів, груп, управління портлетами та відображенням контенту); підтримки контролю доступу на основі ролей. Реалізація API портлетів повністю сумісна зі стандартом JSR 168 та має гнучкий XML-подібний формат опису відображення контенту порталу, який легко модифікується з метою створення спеціальних компоновок контенту.

Сервіси портлетів створені з метою відокремлення функцій, які реалізовані портлетами від сервісів, з якими портлети мають взаємодіяти. Сервіси портлетів реалізують інкапсуляцію логіки, яка багаторазово повторюється з можливістю подальшого використання одним або декількома портлетами. У GridSphere сервіси використовуються для управління всім — від властивостей відображення контенту до сервісів Грід. У такий спосіб зв'язок із Грід-середовищем стає можливим шляхом створення портлетів та сервісів портлетів, які реалізують логіку взаємодії з окремими сервісами ПЗ забезпечення Грід. Прикладом такої реалізації є інструментарій GridPortlets [15].

ІНСТРУМЕНТАЛЬНІ ЗАСІБ VINE TOOLKIT

Інструментарій Vine Toolkit [16] є результатом еволюції проекту GridPortlets для GridSphere у напрямку покращення API та реалізації доступу до Грід-ресурсів. З програмної точки зору Vine Toolkit являє собою Java каркас, який пропонує розробникам простий у використанні високорівневий API для розробки Грід-додатків. Однак найбільшою перевагою є те, що Vine Toolkit — це загальна бібліотека, яка може бути досить легко розгорнута для використання в настільних додатках, Java Web Start, Java Servlet 2.3 та Java Portlet 1.0 середовищах. Це стає можливим завдяки спрощеній системі інсталяції, налаштування та комплектування Vine Toolkit для використання з різними типами програм [17]. Наприклад, Vine може бути розгорнутий як веб-додаток у контейнер сервлету, такий як Apache Tomcat, для використання з каркасом порталу GridSphere.

Інструментарій Vine Toolkit надає розширені сервіси даних BlazeDS з метою поліпшення комунікації клієнт-сервер для розробки інтерактивних і динамічних веб-додатків. Vine Toolkit підтримує Adobe Flex/Flash-технологію, яка дозволяє створювати сучасні та складні веб-інтерфейси, схожі на більшість автономних GUI та інші веб GUI технології, такі як HTML та JavaScript із підтримкою Ajax.

Інструментарій Vine Toolkit забезпечує єдиний загальний API, відкритий користувачеві, який абстрагує різні реалізації ППЗ. Нарешті, інструментарій Vine Toolkit реалізує узагальнену модель на основі ресурсів — будь-які сервіси та джерела даних можуть бути інтегровані за допомогою веб-додатків із використанням API.

Однак основною властивістю Vine Toolkit є підтримка різних ППЗ Грід: gLite, Globus Toolkit, GRIA, OGSA-DAI, SRB, UNICORE [18].

Інструментарій Vine Toolkit побудовано за ієрархію проектів, яка включає основний проект (Root Vine), що визначає базовий API і модель

програмування, на яких будуються підлеглі проекти. Кожен з підлеглих проектів вирішує проблеми конкретної галузі. Проект Grid Vine будується на основі Root Vine для визначення загальніших понять і розширюваних елементів. Проект Globus Toolkit 4 Vine відповідає за підтримку конкретних бібліотек сторонніх виробників і сервісів [17]. Кожному проекту відповідає певна структура файлів, яка визначає яким чином вихідний код має бути побудований, та які бібліотеки сторонніх виробників і конфігураційні файли мають бути упаковані й розгорнуті. Користувачі можуть обирати конкретні проекти Vine Toolkit, необхідні для реалізації їх додатків враховуючи певні залежності між проектами (рис. 2).

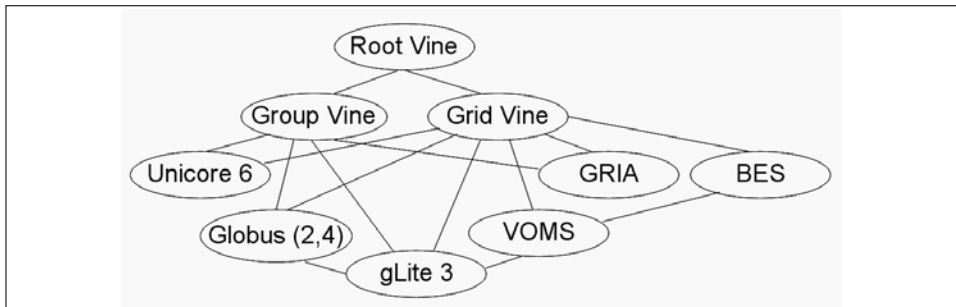


Рис. 2. Ієрархія проектів в Vine Toolkit

Проект Root Vine включає базове управління сесіями, управління сервісами, управління ресурсами та керування завданнями за допомогою визначення таких інтерфейсів: Vine (представляє додаток), Session (представляє сесію діяльності агента або кінцевого користувача), Service Context (інтерфейс для моніторингу завдання), Service (контекст у межах сесії для створення сервісів), Resource (сутність або запис даних), Resource Module (інтерфейс або дескриптор ресурсу) та Resource Manager (відповідальний для управління інформацією про ресурси) [17].

Ресурси (Resource) наразі є найважливішим поняттям, яке моделюється в Vine Toolkit. Комп'ютер, додаток, бібліотека, особа, завдання — усе це є ресурсами Vine. Фактично, ресурси визначають додаток так само, як вони визначають Грід. На самому базовому рівні, Грід представлено як колекцію ресурсів із політиками використання цих ресурсів.

Більш розширені інтерфейси та класи надаються в межах проектів Group Vine та Grid Vine. Group Vine включає в себе управління обліковими записами користувачів, управління реєстрацією користувачів, організацій і контактів та профілів. Vine підтримує реєстрацію користувачів з використанням сторонніх бібліотек та сервісів, тим самим спрощуючи процес створення облікових даних шляхом реєстрації цих повноважень у ППЗ Грід, або шляхом створення облікових записів на віддалених системах у модулях реєстрації, визначених провайдером. Grid Vine визначає поняття, які мають безпосереднє відношення до підтримки розробки Грід-додатків. Окрім моделювання основних концепцій, таких як комп'ютери, ПЗ та сервіси, Grid Vine моделює інформаційні сервіси, сервіси управління файлами послуг та сервіси управління задачами. Проекти, які розширюють Grid Vine, повністю або частково реалізують Grid Vine API. Брокер ресурсів зберігання Vine, наприклад, реалізує файлові аспекти Grid Vine API.

Окрім розглянутих проєктів, Vine Toolkit також включає такі проєкти (будуються на основі розглянутих): UNICORE Vine (підтримка клієнта UNICORE 6), VOMS Vine (підтримка безпеки VOMS), BES Vine (підтримка подання задачі BES), SRB Vine (підтримка управління файлами SRB), RUS Vine (підтримка збору інформації RUS).

У Vine Toolkit реалізовано можливість підключення та інсталяції лише тих проєктів, які необхідні для використання в цільовому середовищі.

СТРУКТУРА НАУКОВОГО ШЛЮЗУ НА БАЗІ GRIDSHERE TA VINE TOOLKIT

Під час використання GridSphere сервером веб-додатків є Apache Tomcat версії 5.0–5.5, який реалізує контейнер сервлету для веб-додатків порталу. Інструментарій GridSphere реалізує контейнер портлету, який діє в якості інтелектуального диспетчера, перенаправляючи запити браузера клієнта відповідним портлетам порталу протягом циклу запитання–відповідь, а також відповідає за створення, використання та знищення портлетів. Така функціональність необхідна для роботи інструментарію Vine Toolkit через відсутність власного портлет-контейнера.

У процесі інсталяції Vine Toolkit не стає частиною GridSphere, а працює паралельно з ним, використовуючи його графічний інтерфейс для запуску та доступу до своїх компонентів. На рис. 3 наведено архітектурну модель наукового шлюзу, побудованого на основі інструментарію Apache Tomcat, GridSphere та Vine Toolkit. На рис. 3 вказано лише ті портлети, які необхідні для організації наукового шлюзу з використанням інструментарію Vine Toolkit.

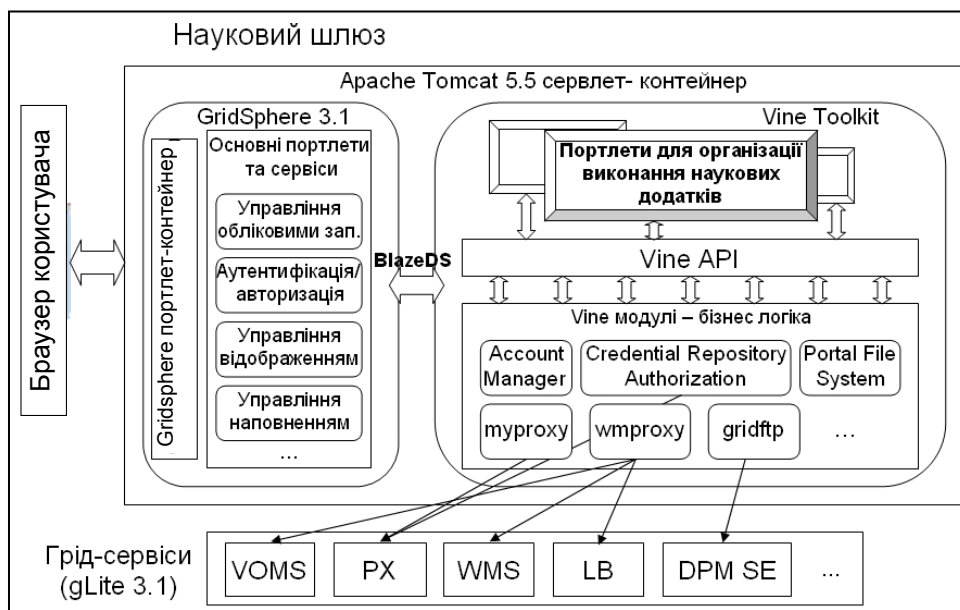


Рис. 3. Архітектура наукового шлюзу

Окрім реалізації контейнера портлетів, GridSphere надає набір портлетів і сервісів портлетів для управління обліковими записами користувачів, аутентифікації, авторизації, управління шаблонами графічного інтерфейсу та системою управління контентом.

Інструментарій Vine Toolkit реалізує функціональність модулів взаємодії з Грід-середовищем, яка може бути використана за допомогою API (так званого Vine API). На рис. 3 наведено основні модулі, використання яких дозволяє відправити задачу в Грід на виконання, проконтролювати її стан та отримати результати. До них відносяться:

- Account Manager — реалізує контроль повноважень користувачів та встановлює порядок авторизації на ресурсах;
- Credential Repository Authorization — реалізує механізм отримання делегованих повноважень (проху-сертифікат) із сервера MyProху (які були туди попередньо делеговані користувачем) для подальшого використання в якості облікових даних (проху-сертифікату) користувача;
- Portal File System — реалізує локальну файлову систему;
- MyProху — реалізує всі допустимі операції з сервером gLite Proху;
- WMPроху — реалізує запуск та моніторинг ходу виконання задачі в Грід шляхом взаємодії з gLite серверами Workload Management та Logging and Bookkeeping. Також взаємодіє з gLite сервером VOMS для доповнення проху-сертифікату користувача VOMS розширенням (необхідна умова взаємодії з деякими gLite серверами, у тому числі WM);
- GridFTP — реалізує всі операції з файлами в межах доступних файлових систем, а також передачу вихідних файлів WMS-сервісу під час запуску задачі.

Vine Toolkit передбачає створення портлетів з використанням технології Adobe Flex, яка дозволяє створювати більш складні та сучасні інтерфейси. Однак при цьому веб-додаток (портлет) поділяється на серверну та клієнтську частину. Для поєднання цих частин використовується технологія BlazeDS, яка, у свою чергу, поліпшує комунікацію завдяки використанню протоколу Action Message Format (AMF), який здатний передавати повідомлення значно більші за розміром, ніж XML-повідомлення.

Для розгляненої архітектури засоби для організації виконання наукових додатків у Грід-середовищі мають бути створені у вигляді портлетів стандарту JSR 168, що дасть можливість реалізувати веб-інтерфейси додатка в середовищі Gridsphere. Реалізація взаємодії з потрібним Грід-середовищем має бути реалізована шляхом використання відповідних API Vine Toolkit. Для цього портлети мають бути представлені у вигляді проекту Vine Toolkit, який будується на основі певної ієрархії базових проектів Vine Toolkit, які включають усі необхідні проекти для реалізації бажаних функцій додатка в цільовому середовищі (у цьому випадку — ППЗ gLite).

Таким чином, середовищем виконання наукових додатків у межах наведеної моделі наукового шлюзу буде деяка підмножина портлетів, яка реалізує користувацькі інтерфейси та всю необхідну функціональність для запуску наукового додатка, моніторингу ходу виконання задачі та отримання результатів у Грід-середовищі.

ЗАСТОСУВАННЯ FLEX-ТЕХНОЛОГІЇ В VINE TOOLKIT ТА GRIDSHERE

Розглянемо ключові особливості використання технології Flex для побудови веб-додатків наукового шлюзу. Для використання Vine API портлет додатка має розроблятися в межах проекту Vine Toolkit (надалі будемо вживати термін «Vine-портлет» для визначення портлетів, створених у такий спосіб). Принципова відмінність Vine-портлетів від звичайних портлетів полягає не стільки в принципі їх роботи в межах сервера (серверна частина відповідає стандарту JSR 168 для коректної роботи з портал-контейнером GridSphere), скільки в принципі їх взаємодії з клієнтом. Стандартний портлет взаємодіє з користувачем через відведену йому веб-форму, яка генерується портлет-контейнером на веб-сторінці, і при відправленні/отриманні даних користувачем оновлюється не вся сторінка, а лише дана веб-форма. Vine-портлет для взаємодії з клієнтом використовує технологію Adobe Flex, згідно з якою

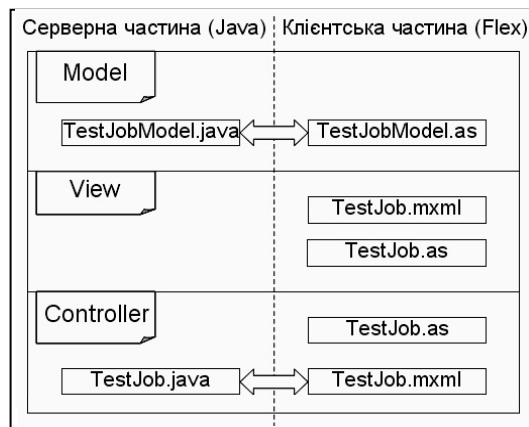


Рис. 4. Структурна діаграма взаємодії компонентів Vine-портлету в контексті MVC-моделі

взаємодія з користувачем здійснюється через спеціальні .swf файли, які відображаються портлет-контейнером у веб-сторінці за допомогою Adobe Flash Player. Файли *.swf (так звані Flex-програми) є результатом компіляції вхідного коду, написаного мовою програмування Action Script. Отже, Vine-портлет поділяється на клієнтську частину (Flex-додаток) і серверну частину (Java-класи), які поєднуються між собою серверною Java технологією BlazeDS. Структурну діаграму взаємодії

компонентів Vine-портлету показано на рис. 4.

Згідно з концепцією Model-View-Controller компоненти Vine-портлету розділено на три окремі частини: модель даних додатка, інтерфейс користувача і бізнес-логіка додатка [19] (рис. 4):

- **Model.** Модель є сполучною ланкою між клієнтською та серверною частиною. Вона містить стан компонента та надає дані (зазвичай для представлення), а також реагує на запити (зазвичай від контролера), змінюючи свій стан. Час життя Flex/BlazeDS компонента відповідає часу життя веб-додатка, тому дані зберігаються лише протягом сеансів користувачів.
- **View.** Відповідає за відображення інформації (інтерфейс користувача — Flex-додаток у вигляді кількох MXML/AS файлів).
- **Controller.** Реалізує бізнес-логіку, яка описана Java-класами, інтерпретує дані, введені користувачем та інформує модель і представлення про необхідність відповідної реакції.

Така конструкція веб-додатка отримала назву Rich Internet Application. Побудова компонентів веб-додатків у такий спосіб дозволяє створювати декілька екземплярів представлення або повторно використовувати бізнес-логіку в інших додатках. До того ж модифікація одного з компонентів має

мінімальний вплив на інші компоненти. Розглянена на рис. 4 структурна діаграма відображає конкретний набір вхідних файлів для тестового портлету (TestJob), з яких згодом і компілюються файли Vine-портлету (.swf-файли), який завантажується на комп'ютері користувача та Java-класи, які виконуються на сервері.

Розробка клієнтського додатка починається з розробки шаблону проекту в Vine Toolkit. Шаблон проекту створюється з використанням графічного інтерфейсу Vine Toolkit та являє собою визначену структуру каталогів і конфігураційних файлів. Для запуску графічного інтерфейсу Vine Toolkit необхідно виконати команду `ant gui` в інсталяційній директорії Vine Toolkit. Далі, використовуючи графічний інтерфейс, необхідно створити шаблон проекту портлету з підтримкою Flex. Після збереження налаштувань відбувається генерація повної структури каталогів проекту з файлами-прикладом для демонстраційного портлету Example. Оскільки всі інсталяційні процеси у Vine Toolkit побудовані на скриптах, які оброблюються консольною утилітою Ant, важливо дотримуватися структури каталогів проекту та розташування файлів, які створюються, у відповідних каталогах.

Vine Toolkit не надає інструментів для створення клієнтської частини портлету, так званих flex-додатків. Вони зазвичай розробляються за допомогою мов програмування ActionScript та MXML. Для компіляції файлів flex-додатків використовується набір класів Flex SDK [20], які потрібно встановити додатково. Бізнес-логіка та графічні елементи представляють собою програмні модулі, які розробник кодує безпосередньо в процесі створення додатка. Функціональність проекту розподіляється в таких модулях (рис. 4):

- TestJobModel.java — модель класів Java взаємодії Flex і Java-технології;
- TestJobModel.as — класи моделі даних клієнтської частини;
- TestJob.mxml — опис візуальних графічних елементів додатка;
- TestJob.as — опис подій взаємодії користувача з візуальними елементами;
- TestJob-config.xml — налаштування компілятора FlexSDK для проекту;
- TestJob.java — Java-класи реалізації бізнес-логіки портлету.

Після завершення кодування вище перелічених модулів необхідно виконати компіляцію вихідних файлів Action Script у файл *.swf за допомогою команди `ant compile-flex - DflexComponent = TestJob`. Інсталяція створеного проекту в середовище Vine Toolkit виконується за допомогою утиліти Apache Ant. Для підключення створеного портлету в Грід-потал використовується стандартний інструментарій для управління відображенням GridSphere.

На сьогодні ведуться розробки зі створення портлетів для організації виконання програмного пакета Gromacs у Грід-середовищі gLite.

ВИСНОВКИ

У цій роботі було проведено огляд особливостей використання наукових інженерних додатків у Грід-інфраструктурі. Зокрема, було висвітлено проблему використання спеціального ПЗ для обчислень науковою спільнотою через відсутність зручних інтерфейсів та засобів для виконання завдань у Грід-середовищі.

Авторами було подано розширену концепцію наукового шлюзу як спосіб вирішення визначених проблем. Згідно з нею, науковий шлюз має надавати зручні графічні інтерфейси користувачеві та реалізовувати процес виконання наукових додатків шляхом взаємодії з наявними ППЗ для можливості використання максимуму наявних Грід-ресурсів. У роботі розглянуто особливості використання технології Flex для розробки користувацьких інтерфейсів веб-додатків для наукового шлюзу.

Застосування зазначеного підходу для створення наукового шлюзу дозволить реалізувати процес виконання додатка в Грід-інфраструктурі в максимально простій та зручній формі.

ЛІТЕРАТУРА

1. *Romaniuk O., Svistunov S., Karpenko D., Marchenko O.* Complex Science Gateway: use of different grid infrastructures to perform scientific applications // Proc. of the 4-th International Conference ACSN-2009. — Lviv. — 2009. — P. 81–84.
2. *Cactus Project: An open source problem solving environment.* — http://www.cactuscode.org/Presentations/GridPortals_2000.ppt.
3. *Grid Compute Environment (GCE) Research Group-Global Grid Forum.* — <http://www.computingportals.org>.
4. *The Astrophysics Simulation Collaboratory (ASC).* — <https://www.ascportal.org>.
5. *NPACI HotPage Grid Computing Portal.* — <http://hotpage.paci.org>.
6. *Globus Alliance.* — <http://www.globus.org>.
7. *Boisseau J., Mary P.* Building Grid Computing Portals // The NPACI Grid Portal Toolkit — Grid Computing. — 2003. — P. 675–700. — <http://www.globus.org/toolkit/cog.html>.
8. *DOE Science grid: Grid Portal Development Kit (GPDK).* — <http://doesciencegrid.org/projects/GPDK/>.
9. *OGCE: Open Grid Computing Environment.* — <http://www.ogce.org/index.php>.
10. *The MyGrid Project.* — <http://www.ebi.ac.uk/mygrid/>.
11. *GridSpeed.* — <http://www.gridspeed.org>.
12. *Suzumura T., Nakada H., Matsuoka S.* GridSpeed: A Web-based Grid Portal Generation Server // Seventh International Conference on High Performance Computing and Grid in Asia Pacific Region. — Tokyo, Japan. — 2004. — P. 26–33.
13. *GPE4GTK project.* — <http://gpe4gtk.sourceforge.net/>.
14. *Novotny J., Russell M., Wehrens O.* GridSphere: An Advanced Portal Framework // Proceedings of the 30-th EUROMICRO Conference. — 2004. — P. 412–419.
15. *The GridSphere Project.* — <http://www.gridsphere.org/gridsphere/gridsphere>.
16. *The Vine Toolkit Project.* — <http://vinetoolkit.org/>.
17. *Russell M., Dziubecki P., Grabowski P.* The Vine Toolkit: A Java Framework for Developing Grid Applications // Parallel Processing and Applied Mathematics. — 2008. — **4967**. — P. 331–340.
18. *Dziubecki P., Grabowski P., Krysiński M., Kuczyński T.* How to build Scientific Gateways with Vine Toolkit and Liferay/GridSphere framework // Presentations and Session Materials for OGF28, 15–18 March. — Munich Germany. — 2010. — http://gforge.man.poznan.pl/files/vine_docs/OGF28_Vine_Toolkit.pdf.
19. *The Vine Toolkit documentation page.* — <http://vinetoolkit.org/content/docs>.
20. *Adobe Open Source Flex 3 SDK downloads page.* — <http://opensource.adobe.com/wiki/display/flexsdk/Download+Flex+3>.

Надійшла 02.06.2010