

- обе программы имеют большой функциональный набор поддержки не только технологии UML, но и других методологий, архитектурных решений и технологий, что обеспечивает многофункциональность обеих программ;

- интерфейс обеих программ довольно сложный из-за поддержки большинства технологий, что сильно его “перегружает”;

- программная документация в Enterprise Architect встроена в программу, что позволяет сразу найти необходимую информацию о возможностях некоторых функций;

- программная документация в Visual Paradigm загружается при первом обращении;

- обе программы имеют большое количество примеров работы, которые выложены на официальных сайтах;

Исходя из вышеперечисленного, Sparx Enterprise Architect можно рекомендовать специалистам, которые работают с технологиями платформы .NET из-за изначальной ориентированности на С-подобные языки. Если разработчик работает с системами, которые связаны с платформой Java, можно рекомендовать Visual Paradigm.

ЛИТЕРАТУРА

1. Ларман Крэг. Применение UML 2.0 и шаблонов проектирования. Практическое руководство. 3-е издание: Пер. с англ. – М.: ООО “И.Д. Вильямс”, 2013. – 736 с.

2. Business Process Design [Электронный ресурс]. URL:: <https://www.itcentralstation.com/categories/business-process-design#top Rated> (Дата обращения: 11.03.2017).

3. List of Unified Modeling Language tools [Электронный ресурс]. URL: https://en.wikipedia.org/wiki/List_of_Unified_Modeling_Language_tools (Дата обращения: 11.03.2017).

УДК 004.55

АНАЛІЗ ТЕХНОЛОГІЙ РОЗРОБКИ НАСИЧЕНИХ ІНТЕРНЕТ-ДОДАТКІВ НА ПЛАТФОРМІ JAVA

В.В. Герасимов¹, В.Я. Левицька²

¹асистент кафедри електронних обчислювальних машин, Дніпровський національний університет ім. О. Гончара, м. Дніпро, Україна, e-mail: gerasimov@dsu.dp.ua

²студент 4 курсу, кафедра електронних обчислювальних машин, Дніпровський національний університет ім. О. Гончара, м. Дніпро, Україна, e-mail: lera0995@gmail.com

Анотація. Проведено аналіз технології JavaFX 8, яка активно розвивається в останній час. Показано особливості цієї технології, розглянуто перспективи її розвитку. Представлені результати порівняння сучасних технологій розробки насичених інтернет-додатків на різних платформах.

Ключові слова: RIA, JavaFX, Flex/AIR, Moonlight, FXML, Java, Groovy, JRuby, Scala, JVM, AJAX, інтерактивність, платформа, технологія, інтерфейс.

ANALYSIS OF RICH INTERNET APPLICATIONS TECHNOLOGIES ON THE JAVA PLATFORM

V.V. Gerasimov¹, Valeria Levitska²

¹assistant, Department of Computer, Oles Honchar Dnipro National University, Dnipro, Ukraine, e-mail: gerasimov@dsu.dp.ua

²student, Department of Computer, Oles Honchar Dnipro National University, Dnipro, Ukraine, e-mail: lera0995@gmail.com

Abstract. The analysis of modern JavaFX 8 technology is discussed. The features of this technology and the prospects for its development are given. The results of the comparison of modern technology development of rich Internet applications on different platforms are provided.

Keywords: RIA, JavaFX, Flex/AIR, Moonlight, FXML, Java, Groovy, JRuby, Scala, JVM, AJAX, interactivity, platform, technology, interface.

Вступ. Останнім часом в області створення Інтернет-додатків намітився нахил до переходу від стандартних HTML/JavaScript/CSS технологій до платформ, які дозволяють запускати в середовищі веб-браузера програми, що зовнішнім виглядом та поведінкою не відрізняються від віконних (desktop) додатків. Цей вид програмних продуктів має назву Rich Internet Application (RIA). Можливості RIA піднімають на новий рівень надійність та зручність використання систем управління контентом, також клієнтська частина RIA може бути запущена в середовищі безпеки – «пісочниці», що надає більш високий рівень безпеки та забезпечує високу надійність передачі даних. Основним напрямком використання цієї технології є розробка клієнтської частини великих проєктів, яка зв'язується з сервером і виконує певні бізнес-функції. Також можна реалізовувати й інші окремі додатки — невеликі ігри, презентації та ін. [1].

Між технологіями розробки RIA-додатків завжди існувала боротьба за першість, однак до останнього часу не було сильного конкурента для Flex/AIR від Adobe, Moonlight від Novell (вільне і відкрите втілення каркасу веб-додатків Silverlight від Microsoft), однак ситуація змінилася, адже у Java-розробників з'явився потужний конкурентоспроможний інструмент —

JavaFX 8. Всі частини одного проекту тепер можна реалізувати на одній платформі Java, що в свою чергу забезпечує швидкодію, підвищує безпеку, скорочує час розробки і економить витрати на розробку.



Рисунок 1 – Rich Internet Applications серед технологій створення програмних систем

Метою роботи є дослідження технології JavaFX 8 та проведення порівняльного аналізу її з найпоширенішими подібними платформами розробки RIA-додатків.

Основна частина. Найпершою RIA-технологією були Java-аплети, що стали доступними ще в 1995 році, але вони не набули широкого розповсюдження. Технологія JavaFX 1.x, яка була розроблена пізніше, мала ту ж саму історію. 18 березня 2014 року була випущена JavaFX 8, як частина реалізації Oracle Development Kit (JDK) 8. JavaFX 8 – базовий набір інструментів для розробки інтерфейсу користувача для Java SE 8, що представляє єдине узгоджене середовище програмування додатків як для вбудованих, так і для настільних систем [2].

На конференції JavaOne 2010 Oracle продемонстрували довгостроковий план для JavaFX (рис. 2) [3].

JavaFX — набір графічних і медіа-пакетів, що дозволяє розробникам проектувати, створювати, тестувати, налагоджувати і розгортати насичені клієнтські додатки, які працюють для різних платформ. JavaFX 8 став частиною пакетів JRE/JDK разом з версією Java 8, тому для компіляції і запуску JavaFX програм більше немає необхідності у встановленні додаткових програм. В JavaFX 8 покращена підтримка HTML 5 у WebView, робота з 3D-графікою [2].

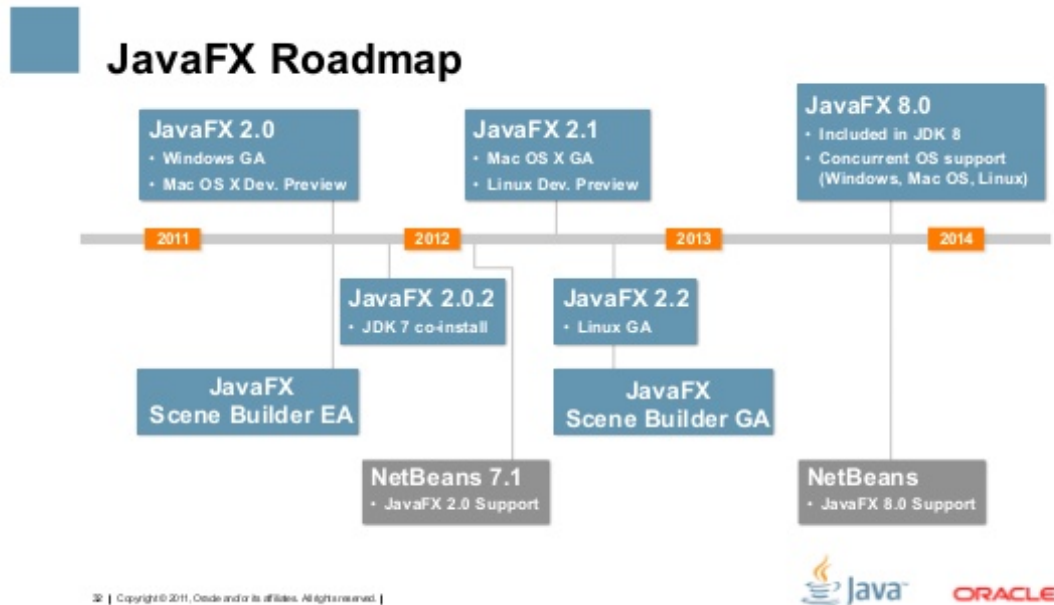


Рисунок 2 – Схема розвитку технології JavaFX

Нові функції, які були введені в JavaFX 8 [4]:

- **Java API.** API-інтерфейси призначені як альтернатива мов Java Virtual Machine (Java VM), таких як JRuby і Scala.
- **FXML і Scene Builder.** FXML — це мова розмітки для побудови інтерфейсу користувача в JavaFX-додатках. Дизайнер може кодувати в FXML або використовувати JavaFX Scene Builder для інтерактивного проектування графічного інтерфейсу користувача (GUI). Сцена Builder генерує FXML розмітку, яка може бути перенесена на IDE, де розробник може додати бізнес-логіку.
- **WebView.** Веб-компонент, який використовує технологію WebKitHTML, щоб зробити можливим вбудування веб-сторінок всередині програми JavaFX. Також була додана підтримка HTML5, в тому числі веб-сокетів, Web Workers, веб-шрифтів і можливість друку.
- **Сумісність зі Swing.** Існуючі програми Swing можуть бути оновлені з функціями JavaFX.
- Вбудовані елементи керування для інтерфейсу користувача і CSS.
- **3D-графіка.** Нові класи API для Shape3D, SubScene, Material, PickResult, LightBase і SceneAntialiasing були додані в бібліотеку 3D-графіки в JavaFX 8.
- **Canvas API** дозволяє малювати безпосередньо в районі сцени JavaFX, яка складається з одного графічного елементу (вузла).
- **API для друку.** Пакет javafx.print був доданий в Java SE 8.
- **Підтримка Rich Text.** JavaFX 8 привносить розширену підтримку тексту.

- **Підтримка Multitouch.** JavaFX забезпечує підтримку мультисенсорних операцій, ґрунтуючись на можливостях базової платформи.
- **Hi-DPI підтримка.** JavaFX 8 тепер підтримує Hi-DPI дисплеї.
- Апаратне прискорення графічного конвеєра.
- Автономна модель розгортання додатків.

Порівняємо технологію JavaFX з основними її конкурентами. По кожній властивості технології буде виставлена кількість набраних балів від 1 до 5. Властивості, за якими було порівняно технології:

- Досвід користувачів — цей критерій показує рівень інтерактивності та швидкість відгуку: human-computer interaction (HCI). На основі цього будується інтерактивна модель додатку;
- Розгортання — описує рівень комфортності при розгортанні додатку (створення релізу, інсталяція, активація, оновлення, адаптація, деінсталяція та ін.);
- Забезпечення широкої доступності — критерій описує такі речі як кількість користувачів, поширеність та доступність;
- Багатотехнологічність — критерій взаємодії або її відсутності з іншими технологіями;
- Час запуску — час, необхідний для старту роботи додатка;
- Клієнт-Сервер — порівнюються процеси на сервері або клієнті;
- Платформонезалежність;
- Підтримка розробників — цей критерій стосується інструментів розробки та мов програмування;
- SEO, доступність — можливість автоматизованого індексування та доступність для людей-інвалідів;
- Роздільна здатність екрану — критерій незалежності від розділювальної здатності екрану;
- Насиченість UI — використання стандартних компонентів, кастомізація, скіни, стилі, анімація та ін.;
- Вартість — критерій вартості розробки, ліцензії, вартість або безкоштовність використання технології.

Таблиця 1 - Порівняння технологій.

| Властивості | AJAX | Flex/AIR | Moonlight | JavaFX |
|---------------------|---|----------|-----------|--------|
| Досвід користувачів | ++ | ++++ | ++ | ++++ |
| | AJAX, відносно досвіду користувачів, революційно змінив звичайний DHTML, однак він є дуже залежним від парадигми браузера. Moonlight порівняно з іншими підтримує меншу кількість браузерів. Flex/AIR має гарну | | | |

| Властивості | AJAX | Flex/AIR | Moonlight | JavaFX |
|----------------------------------|--|----------|-----------|--------|
| | підтримку в браузерах. Під час запуску Java-додатків дещо погіршується швидкість відгуку додатку. | | | |
| Розгортання | ++++ | ++++ | +++ | +++ |
| | Центральне розгортання серверу пропонують всі технології. AJAX не потрібно оновлювати, тому що він поставляється з браузерами. Тому він має найменший бар'єр для використання користувачами веб-додатку. Flex і JavaFX пропонують автоматичний оновлювач. Але Java при оновленні перериває на довший час потік робіт (workflow) користувача. | | | |
| Забезпечення широкої доступності | +++++ | ++++ | +++ | ++++ |
| | AJAX вбудований в усі браузери. Flex/AIR є доступним у 97 % браузерів, іноді інсталюється з операційною системою. Java також в більшості випадках інсталюється. Moonlight менш поширений. | | | |
| Багатотехнологічність | + | +++ | ++++ | ++++ |
| | AJAX не вистачає взаємодії між різними стандартами та платформами і різними мовами програмування. Flex/AIR, JavaFX та Moonlight — поодинокі технології, які пропонують єдине середовище виконання, яке забезпечує однакову поведінку на багатьох платформах. Silverlight дозволяє обирати мову для реалізації. В JavaFX також це доступно через використання таких мов JVM, як Groovy і Scala. | | | |
| Час запуску | +++++ | +++++ | ++++ | +++ |
| | Головною рисою AJAX є те, що він одразу підтримується браузерами і запуск додатків на ньому миттєвий. Flex/AIR також дуже швидкий. Moonlight трохи повільніший. Час запуску Java потребує декількох кроків і тому він більший за інших в цьому списку. | | | |
| Клієнт – Сервер | ++ | ++++ | +++ | +++++ |
| | AJAX пропонує асинхронну комунікацію і зменшення навантаження на сервер, але не має сокет-з'єднання. Flex/AIR має сокет-з'єднання з AIR. Moonlight та Flex/AIR забезпечують швидку взаємодію з користувачем, але ця технологія націлена на виконання на боці клієнта. Java технологія добре підходить для серверної сторони та пропонує однорідну комунікацію між сервером та RIA-клієнтом. | | | |

| Властивості | AJAX | Flex/AIR | Moonlight | JavaFX |
|----------------------------|---|----------|-----------|--------|
| Платформо-незалежність | +++++ | ++++ | +++ | ++++ |
| | Серед всіх Moonlight підтримує меншу кількість платформ, Flex/AIR та JavaFX є кращими за нього, але в той же час AJAX має ширші можливості майже на усіх мобільних пристроях. | | | |
| Підтримка розробників | +++ | ++++ | ++++ | +++++ |
| | Для AJAX розробки існують допоміжні інструменти, але їх небагато і вони не є кросплатформними. Допоміжні інструменти розробки Flex/AIR є платними, а інструменти розробки JavaFX є безкоштовними. | | | |
| SEO, доступність | +++ | + | + | + |
| | AJAX пропонує найкращі показники SEO та доступності. JavaFX, Moonlight та Flex/AIR потребують використання HTML-альтернатив. | | | |
| Роздільна здатність екрану | + | ++++ | ++++ | ++++ |
| | AJAX може використовувати тільки браузерну піксельну технологію. JavaFX, Moonlight та Flex/AIR використовують векторну графіку. | | | |
| Насиченість UI | + | +++++ | ++++ | ++++ |
| | AJAX впроваджує стилювання за допомогою CSS (каскадні таблиці стилів), але не має детального стилювання компонент та недостатньо реалізує концепцію «Look and Feel» (вигляд і поведінку). Flex/AIR порівняно з JavaFX та Moonlight в цьому плані пропонує більше. | | | |
| Вартість | +++ | ++ | +++ | ++++ |
| | Для деяких браузерів потрібна інша реалізація додатку на AJAX і це потребує додаткових витрат. Flex/AIR не є безкоштовною технологією, як і засоби для їх розробки. JavaFX і Moonlight та інструменти для розробки таких додатків є безкоштовними, з відкритим кодом. | | | |
| Результат | 35 | 44 | 39 | 45 |

Таким чином, не дивлячись на відносно невелику швидкість запуску додатків, написаних за допомогою JavaFX, ця технологія одержує найбільшу кількість балів завдяки безкоштовності технології та інструментів розробки, великої кількості Java-спільноти та підтримки різних мов програмування. Не дивлячись на невелику кількість набраних балів AJAX є дуже популярною, але її переважно використовують для підвищення інтерактивності та комфо-

рності при роботі з веб-ресурсами (соціальні мережі, сайти, інтернет-магазини), а не для розробки RIA додатків. Flex — набір утиліт для розробки RIA на платформі Adobe Flash, що надає програмний спосіб розробки RIA. Створення інтерфейсу користувача відбувається за допомогою мови MXML — простої та інтуїтивної, що дає можливість швидко її освоїти досвідченим розробникам. Оскільки Flex має значно менше сторонніх бібліотек та основне середовище розробки Flex Builder є платним, то технологія JavaFX знаходить своїх прихильників і кількість розроблених додатків за допомогою цієї технології нестримно росте.

Висновки. Корпорація Oracle зробила великий крок вперед в напрямку RIA, що є дуже доброю новиною для Java-розробників. Технологія має багато різних реалізованих елементів інтерфейсу та класів, які дуже прискорюють розробку додатків. Цю технологію також можуть використовувати розробники Ruby, Scala та інших мов, оснований на JVM. Проєкт JavaFX з відкритим вихідним кодом і є частиною проєкту OpenJDK. Великою перевагою JavaFX є те, що додаток, написаний за допомогою цієї технології, може мати декілька режимів розгортання і працювати на різних типах пристроїв, що робить цей додаток дуже універсальним і при цьому заради цієї універсальності не потрібно нічого робити додатково.

JavaFX 8 була повністю інтегрована в платформу Java 8 і перестала вимагати окремого встановлення. Також була доповнена програмним інтерфейсом CSS API, інтеграцією з бібліотекою SWT, 3D-графікою, покращено інструмент JavaFX Scene Builder. Зараз технологія JavaFX забезпечує створення потужного графічного інтерфейсу користувача (GUI), 2D- та 3D-графіка для великомасштабних додатків, орієнтованих на обробку даних, насичених медіа-додатків, що поставляють різноманітний контент користувачам [5].

Удосконалення JavaFX продовжується й зараз. Розробники зосереджені на вилученні деяких корисних частин API із коду JavaFX, щоб зробити їх загальнодоступними.

ЛІТЕРАТУРА

1. Carl-David Granbäck. Rich Internet Applications (RIAs): A Comparison Between Adobe Flex, JavaFX and Microsoft Silverlight. — Chalmers University of Technology — Göteborg, 2009 — P. 1-70.
2. Kishori Sharan. Learn JavaFX 8 (The Expert's Voice in Java): Building User Experience and Interfaces with Java 8. — New York, 2015 — P. 1-1173.
3. SlideShare. Simon Ritter. Introduction to JavaFX — Режим доступу: <https://www.slideshare.net/lukaszromanowski1981/03-java-fxintroductionssitter>
4. JavaFX Documentation Home. Getting Started with JavaFX — Режим доступу: <https://docs.oracle.com/javase/8/javafx/get-started-tutorial/jfx-overview.htm>

5. Revolvу. JavaFX – Режим доступа:
https://www.revolvу.com/topic/JavaFX&item_type=topic

УДК 004.42

ФУНКЦИОНАЛЬНАЯ СТРУКТУРА АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ РАЗРАБОТКИ РАСПИСАНИЯ ДВИЖЕНИЯ ПРИГОРОДНЫМ ВОДНЫМ ТРАНСПОРТОМ

А.Р. Гибадуллин

студент 1-го курса магистратуры кафедры автоматизированной системы обработки информации и управления, Казанский национальный исследовательский технический университет им. А.Н. Туполева – КАИ, г. Казань, Российская Федерация, e-mail: gibadullin.albert@gmail.com

Аннотация. В данной работе представлена функциональная структура АИС: логическая модель, модель поведения и графическая нотация, предназначенная для формирования и описания автоматизированной информационной системы разработки расписания движения пригородного водного транспорта.

Ключевые слова: автоматизированная информационная система, пригородный водный транспорт, расписание движения.

FUNCTIONAL STRUCTURE OF AUTOMATED SYSTEM OF DEVELOPMENT OF SCHEDULE OF MOVEMENT BY COUNTRY WATER TRANSPORT

Albert Gibadullin

1-year student of magistracy of the department of automated systems of information processing and management, Kazan National Research Technical University A. Tupolev - KAI, Kazan, Russian Federation, e-mail: gibadullin.albert@gmail.com

Abstract. In this paper, the functional structure of the AIS is presented: a logical model, a behavior model, and a graphical notation intended for the formation and description of an automated information system for the development of a timetable for suburban water transport.

Keywords: automated information system, the suburban water transport timetables.

Введение. Сегодня пассажирские суда преимущественно используются для перевозки туристов и пассажиров в труднодоступные районы, что особенно подчеркивает значение речного транспорта в транспортной системе.