

Нешляева Арина Владимировна

АКТУАЛЬНОСТЬ ИНСТРУМЕНТА ПОСТРОЕНИЯ ГРАФИЧЕСКОГО ИНТЕРФЕЙСА ПОЛЬЗОВАТЕЛЯ В ОБЛАЧНЫХ СРЕДАХ РАЗРАБОТКИ

В статье проведён анализ специфических особенностей облачных сред разработки, выделены основные преимущества и недостатки их использования, сделан обзор наиболее популярных инструментов. На основе выполненного анализа, а также проведенного маркетингового исследования показаны важность и актуальность наличия инструмента для построения графического интерфейса пользователя в облачных средах разработки.

Адрес статьи: www.gramota.net/materials/1/2013/5/42.html

Статья опубликована в авторской редакции и отражает точку зрения автора(ов) по рассматриваемому вопросу.

Источник

Альманах современной науки и образования

Тамбов: Грамота, 2013. № 5 (72). С. 134-136. ISSN 1993-5552.

Адрес журнала: www.gramota.net/editions/1.html

Содержание данного номера журнала: www.gramota.net/materials/1/2013/5/

© Издательство "Грамота"

Информация о возможности публикации статей в журнале размещена на Интернет сайте издательства: www.gramota.net

Вопросы, связанные с публикациями научных материалов, редакция просит направлять на адрес: almanac@gramota.net

УДК 004.4'236

Технические науки

В статье проведён анализ специфических особенностей облачных сред разработки, выделены основные преимущества и недостатки их использования, сделан обзор наиболее популярных инструментов. На основе выполненного анализа, а также проведенного маркетингового исследования показаны важность и актуальность наличия инструмента для построения графического интерфейса пользователя в облачных средах разработки.

Ключевые слова и фразы: инструмент для построения графического интерфейса пользователя (GUI Builder); облачная среда разработки (cloud-based IDE); разработка программного обеспечения как услуга (SDaaS).

Нешляева Арина Владимировна

Новосибирский национальный исследовательский государственный университет

neshlyaeva.arina@gmail.com

АКТУАЛЬНОСТЬ ИНСТРУМЕНТА ПОСТРОЕНИЯ ГРАФИЧЕСКОГО ИНТЕРФЕЙСА ПОЛЬЗОВАТЕЛЯ В ОБЛАЧНЫХ СРЕДАХ РАЗРАБОТКИ[©]

Введение

За последние несколько лет концепция облачных вычислений и виртуализации набрала силу и стала популярной в сфере информационных технологий. Облачные вычисления обеспечивают удобный доступ к внешним вычислительным ресурсам в виде сетевого сервиса. Различают три основные модели облачных сервисов:

- инфраструктура как сервис (Infrastructure as a Service, IaaS);
- платформа как сервис (Platform as a Service, PaaS);
- программное обеспечение как сервис (Software as a service, SaaS).

Логическим развитием второй и третьей модели стала модель «Разработка программного обеспечения как услуга» (Software Development as a Service, SDaaS), объединившая возможности PaaS и SaaS для разработчиков [1]. Она позволяет создавать, тестировать и развертывать приложения в облаке. Неотъемлемой частью SDaaS является облачная среда разработки.

Облачная среда разработки

Интегрированная среда разработки (англ. *IDE, Integrated Development Environment*) – система программных средств, используемая программистами для разработки программного обеспечения (ПО) [3]. Интегрированная среда предназначена для увеличения продуктивности программиста, предоставляя соответствующие компоненты и инструменты в едином пользовательском интерфейсе.

Обычно интегрированная среда разработки состоит из:

1. текстового редактора;
2. компилятора или интерпретатора;
3. инструментов для сборки приложения;
4. инструментов для отладки.

Также обычно интегрированная среда разработки включает в себя инструмент построения графического интерфейса пользователя (*GUI Builder*, англ. *Graphical User Interface Builder*).

Облачные среды разработки размещены провайдером в облачном хранилище и, как правило, доступны разработчикам через браузер. Они обладают специфическими преимуществами и недостатками по сравнению с версиями для настольных ПК [1].

Преимущества:

1. Независимость от платформы (для работы в облачной среде разработки требуются только браузер и интернет-соединение. Разработчик может работать с любой платформы, даже со своего мобильного устройства).
2. Повсеместный доступ (исходный код доступен в любом месте в любое время при наличии интернет-соединения; разработчик может увидеть свой код, сделав несколько щелчков мышью в браузере).
3. Совместная разработка (облачная среда разработки дает дополнительные возможности для совместной разработки, например, может обеспечить парное программирование двух разработчиков, находящихся на разных континентах).
4. Увеличенная продуктивность (при использовании облачной среды разработчик может сосредоточиться на самой реализации, а не на технической конфигурации).
5. Уменьшенная лицензионная стоимость (облачные средства разработки позволяют организациям использовать модель «плати, пока используешь» (англ. “pay as you go”).

Недостатки:

1. Зависимость от провайдера (облачные средства разработки могут быть недоступны в течение неопределенного времени).

2. Безопасность исходного кода может быть под угрозой (при использовании облачной среды разработки исходный код, настройки окружения и многое другое хранятся на внешних серверах и управляются сторонней компанией).

3. Необходимость в постоянном интернет-соединении (при работе с облачными средствами требуется постоянное быстрое интернет-соединение).

В ходе работы были рассмотрены наиболее популярные облачные среды разработки (согласно числу ссылок на сайтах разработчиков) – *Stackoverflow* [7], *InfoQ* [12], *DZone* [11].

При этом было уделено особое внимание следующим свойствам:

- Поддержка различных языков программирования.
- Средства для упрощения написания и отладки кода.
- Возможность работать без интернет-соединения.
- Возможность интеграции и модификации готового проекта.
- Возможность работы с базами данных.
- Возможность расширения IDE с помощью плагинов.
- Наличие GUI Builder.

Свойство	CodeRun	Kodingen	Complir	Cloud9	Orion
Языки программирования	C#, PHP, HTML, JavaScript, CSS	Java, NodeJS, Perl, Python, Ruby, C, C++, Go	HTML5, JavaScript, CSS, Java, C++, C, C#, Ruby, Fortran	HTML5, JavaScript, CSS, Ruby, PHP, Python	HTML5, JavaScript, CSS
Отладчик	Да	Нет	Нет	Да	Нет
Загрузка готовых проектов	Да	Нет	Нет	Да	Да
Поддержка баз данных	Да	Да	Нет	Нет	Нет
Возможность работы без интернет-соединения	Нет	Нет	Нет	Да	Нет
Настройка с помощью плагинов	Нет	Нет	Нет	Нет	Да
Наличие <i>GUI Builder</i>	Нет	Нет	Нет	Нет	Нет

Из проведенного исследования можно сделать следующие выводы.

Облачные среды разработки пока не могут полностью заменить настольные *IDE*, такие как, например, *Visual Studio* или *Eclipse*, они не предназначены для создания настольных приложений, программ, требующих работы с базами данных. Но, несмотря на это, они являются достаточно популярными среди программистов.

В основном они ориентированы на начинающих разработчиков и программистов, которым необходимо экспериментировать с новыми технологиями и языками для создания мобильных, веб- и облачных приложений, о чем свидетельствует широкая поддержка языков *HTML5*, *CSS*, *JavaScript*. Стоит заметить, что популярность такого сорта приложений растет с каждым годом. Так, например, прирост количества веб-сайтов и веб-приложений за месяц составляет 8 млн (согласно данным отчета исследовательской компании *NetCraft* [6]), а, согласно опросу *Eclipse* 2012 года [2], количество веб-приложений, разработанных в этой среде, увеличилось на 8% по сравнению с 2011 годом, а количество мобильных приложений – на 10%.

И, безусловно, облачная среда может быть успешно использована для разработки любых приложений, если ее преимущества (см. выше) важны для конкретного проекта, команды разработчиков или организации.

Стоит заметить, что ни одна из рассмотренных в данном исследовании облачных интегрированных сред разработки не имеет встроенного инструмента для построения графического интерфейса пользователя.

Инструмент для построения графического интерфейса

Инструмент для построения графического интерфейса (англ. *GUI Builder*, также известный как *GUI Designer*) – инструмент разработки программного обеспечения, который упрощает создание графического интерфейса с помощью готовых виджетов и технологии *drag-and-drop* в *WYSIWYG* (англ. *What You See Is What You Get*, «что видишь, то и получаешь») редакторе [4]. Под виджетом далее будем понимать элемент интерфейса, имеющий стандартный внешний вид и выполняющий стандартные действия. Без *GUI Builder* графический интерфейс создается вручную, при этом параметры виджетов специфицируются в коде без визуального отображения до тех пор, пока программа не будет запущена. Пользовательские интерфейсы, как правило, создаются на основе архитектуры, управляемой событиями (англ. *event-driven architecture*), поэтому *GUI Builder* призван также упростить создание такого кода. Этот код соединяет виджеты с событиями, которые вызывают функции, обеспечивающие логику приложения.

Обычно *GUI Builder* состоит из 3-х основных частей:

1. Панель инструментов, на которой располагаются доступные виджеты.
2. Основная секция для создания дизайна, куда перетаскиваются виджеты из панели инструментов.

3. Панель свойств, в которой можно изменить характеристики выбранного виджета и добавить к нему события.

Актуальность инструмента для построения графического интерфейса в облачной среде разработки

Отношение разработчиков к использованию инструмента для построения графического интерфейса пользователя весьма поляризовано – кто-то создает пользовательский интерфейс всегда вручную, кто-то наоборот считает, что для создания визуального интерфейса нужно пользоваться визуальными средствами. Так, например, *Josh Marinacci*, эксперт в области графических интерфейсов, ведущий разработчик в компаниях *JavaFX*, *Swing*, *NetBeans* в своей статье «A Response to GUI Building: Tool vs Hand Coded» пишет: «Создание графического инструмента – это визуальная задача. Используйте визуальный инструмент!» [9].

Ben Galbraith, Java-архитектор, участник сотни технических презентаций и симпозиумов в своей статье «Successful GUI Building... Today» утверждает, что у разработчика среднего звена производительность работы с использованием визуальных инструментов увеличивается в 2-10 раз (в зависимости от разрабатываемого приложения) [8].

Были проанализированы обсуждения на эту тему на сайтах *Stackoverflow* [7], *Iphonedevsdk* [5], *Coderanch* [10]. Анализ показал, что при аргументации своего выбора разработчики опираются на следующие факторы:

- опыт разработчика;
- настольное или веб-приложение;
- статические или динамические элементы пользовательского интерфейса;
- специализированный или стандартный интерфейс;
- требуемая скорость разработки;
- дальнейшая поддержка и внесение модификаций в сгенерированный код.

В таблице ниже отражены предпочтения разработчиков в отношении подходов к разработке *GUI* в зависимости от приведённых факторов:

Свойство	Создание интерфейса с помощью <i>GUI Builder</i>	Создание интерфейса вручную
Опыт разработчика	Начинающий	Опытный
Тип приложения	Настольное или мобильное	Веб-приложение
Интерфейс	Стандартный	Специализированный
Скорость разработки	Быстрая	Средняя/Медленная
Элементы	Статические	Динамические

На основании представленных данных можно сделать вывод, что *GUI Builder* является незаменимым инструментом для начинающего разработчика, или если требуются быстрая разработка или прототипирование интерфейса для настольного или мобильного приложения. Данное заключение приводит нас к тому, что *GUI Builder* может быть очень востребованной частью облачной среды разработки (которая, как мы выяснили раньше, предназначена для быстрой разработки и начинающих программистов), особенно в случае разработки мобильных приложений. Графический интерфейс является важной и неотъемлемой частью мобильных приложений, однако из-за новизны мобильных технологий и скорости развития этой отрасли интерфейс программ для мобильных устройств является на текущий день очень разнородным и нестандартизованным. Использование *GUI Builder* при разработке мобильных приложений может помочь стандартизовать пользовательский интерфейс и сделать его интуитивно понятным для конечного пользователя.

Список литературы

1. **Coding in the Cloud: White Paper from Parity Solutions, 2012** [Электронный ресурс]. URL: <http://www.parity.net/SiteCollectionDocuments/Coding%20in%20the%20Cloud.pdf>
2. **Eclipse Open Source Developer Report, 2012** [Электронный ресурс]. URL: http://www.eclipse.org/org/foundation/boardminutes/2012_06_exhibits/ExhibitH.pdf
3. http://en.wikipedia.org/wiki/Integrated_development_environment
4. http://en.wikipedia.org/wiki/Gui_builder
5. <http://iphonedevsdk.com/>
6. <http://news.netcraft.com/archives/2012/12/04/december-2012-web-server-survey.html>
7. <http://stackoverflow.com/>
8. http://weblogs.java.net/blog/javaben/archive/2005/08/successful_gui.html
9. http://weblogs.java.net/blog/joshy/archive/2007/06/a_response_to_g.html
10. <http://www.coderanch.com>
11. <http://www.dzone.com>
12. <http://www.infoq.com/>