



Рис. 1. Управление качеством проекта

Увеличение сложности и размера современных мобильных приложений и одновременный рост ответственности и важности выполняемых ими функций отразились на предъявляемых требованиях в области качества и безопасности их использования со стороны заказчиков и пользователей. Одной из основных проблем обеспечения качества программных средств является формализация характеристик качества и методология их оценки. Существует ряд современных инструментов и технологий, позволяющих добиться высокого уровня организации процесса разработки приложений, однако для обеспечения высокой эффективности и качества функционирования программного обеспечения, адекватности процесса оценки качества, создания возможностей для дальнейшего совершенствования и развития приложения необходимо применение действующих международных стандартов.

Оптимальный выбор модели качества, следование актуальным международным стандартам, постоянный контроль и обеспечение качества разработки лежат в основе успешной реализации мобильного приложения, в полной мере отвечающего предъявляемым к нему современным требованиям.

СПОСОБ ОРГАНИЗАЦИИ АСИНХРОННОЙ КОММУНИКАЦИИ МЕЖДУ JAVASCRIPT И ACTIONSCRIPT

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Ясюкевич П. П.

Лукьянец В. Г. – канд. техн. наук, доцент

Несмотря на то, что технология Flash постепенно теряет былую популярность, уступая новейшим технологиям HTML5, она по-прежнему остаётся незаменимой в некоторых областях web-разработки, таких как 2D- и 3D-анимация, воспроизведение потокового видео и др. Поэтому при её использовании необходимо хорошо понимать, как происходит взаимодействие между страницей (вычислительной средой браузера) и самой средой выполнения Flash.

Использование коммуникации между Flash (с помощью языка ActionScript) и браузером (с помощью языка JavaScript) подразумевает учёт того, что передача данных может быть как синхронной, так и асинхронной, т.е. осуществиться спустя некоторый интервал времени. Примером такого поведения может служить запуск трансляции видеопотока на внутри плагина, где невозможно сказать сразу, удалось ли начать воспроизведение или нет, из-за того, что Flash нужно некоторое время для подключения к видеопотоку и распознавания его формата. Так может возникнуть ошибка из-за неверного адреса потока или из-за отсутствия поддержки кодека, используемого в видео, о чём можно уведомить страницу только через некоторое время. Именно с учётом возможной задержки и разрабатывался данный способ взаимодействия вычислительных сред.

Также на начальном этапе разработки были замечены особенности работы плагина Flash в различных вычислительных средах, именно поэтому было исследовано поведение плагина Flash версий 10, 11 и 12 в наиболее популярных браузерах, таких как Mozilla Firefox, Google Chrome, Microsoft Internet Explorer, а также Opera (Presto).

Результаты исследования показали различия способов функционирования Flash в различных средах выполнения, а также различия в поведении браузера, связанного с подключением среды функционирования Flash, в частности это касается браузера Microsoft Internet Explorer, где плагин реализован в виде расширения ActiveX, что накладывает некоторые ограничения на использование коммуникаций между JavaScript и ActionScript.

В общем случае взаимодействие между ActionScript и JavaScript происходит по следующей схеме:

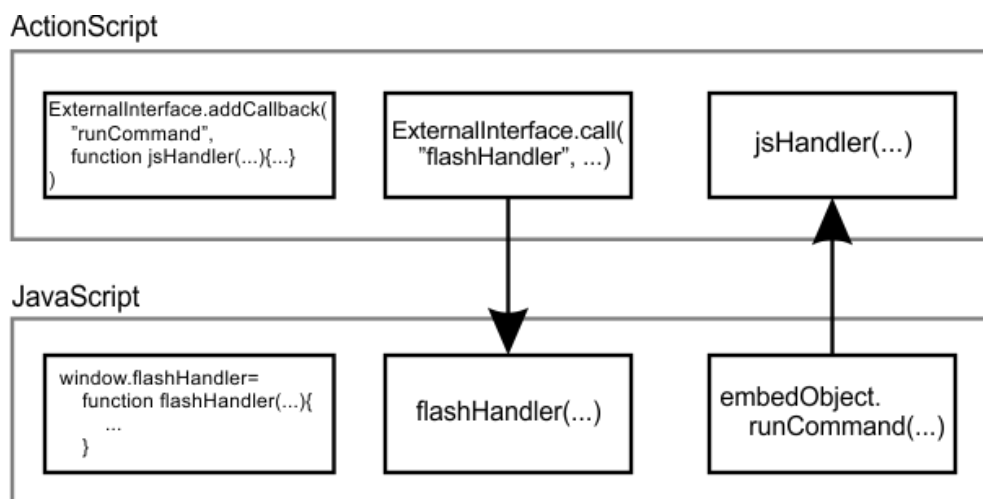


Рис. 1 – Структурная схема вызова внешних функций между ActionScript и JavaScript

Первым блоком в каждой группе идёт блок инициализации, определяющий функцию обратного вызова для данного языка. Стрелками на схеме обозначены направления вызовов функций. Стоит отметить, что такой вызов синхронный, т.е. вызывающая функция получает значение от выполненной функции. Передаваемое значение может иметь как простой формат (числа, строки, булевы значения), так и комплексный в виде JSON-объекта (JSON — JavaScript Object Notation, способ записи объектов JavaScript).

Также в результате исследования было установлено, что во всех браузерах Flash можно подключить как с помощью классического тега `<object>`, так и с помощью тега `<embed>`, несмотря на то, что данный способ не указан в официальной документации как доступный для всех браузеров. Выявленной особенностью стал тот факт, что Internet Explorer не инициализирует функции обратного вызова внешнего интерфейса ActionScript, если используется тег `<embed>`, однако их можно вызвать с помощью ActiveX-функции `CallFunction`.

К тому же, Internet Explorer не позволяет использовать произвольное имя для функции внешнего вызова ActionScript, запрещая такие как `play`, `stop`, `isPlaying` и др.

Последней обнаруженной особенностью взаимодействия Flash и браузеров стала повторная инициализация Flash в случае перемещения тега встраивания в дереве элементов страницы, а в Internet Explorer происходил полный отказ внешнего интерфейса ActionScript. Поэтому в случае использования механизма внешних функций стоит отказаться от перемещения объекта встраивания.

Для организации асинхронного взаимодействия было решено использовать механизм очереди, в котором должны храниться объекты, ещё не прошедшие инициализацию из-за асинхронности выполняемого действия. Впоследствии проинициализированные объекты должны передаваться назад во внешнюю функцию, где предполагается их асинхронный приём и обработка.

Для возможности создания коммуникации с использованием функций с произвольным именем было принято решение организовывать взаимодействие между средами с помощью единственной функции обратного вызова с каждой стороны, причём имя этой функции не должно быть зарезервировано в Internet Explorer. Как видно на рис. 1, для внешней функции ActionScript было выбрано имя `runCommand`, а для JavaScript — `flashHandler`. Таким способом достигается одинаковое поведение кода в различных браузерах.

Таким образом, было исследовано поведение плагина Flash различных версий в различных браузерах и был разработан способ организации взаимодействия браузера с данным плагином, учитывающий не только особенности поведения Flash в различных условиях, но также предоставляющий возможность избежать потенциальных проблем, которые могли бы возникнуть при организации коммуникации в случае, когда данный способ не используется. Также составлены рекомендации по разработке программного обеспечения, использующего данный механизм организации коммуникации.

Список использованных источников:

1. Мук, К. ActionScript 3.0 для Flash CS3 / Колин Мук — СПб.: Питер, 2009. — 987 с.
2. Флэнаган, Д.. JavaScript. Подробное руководство, 5-е издание. — Пер. с англ. / Дэвид Флэнаган — СПб.: Символ-Плюс, 2008. — 992 с.