Использование объектов JavaScript на экранных формах SCADA-системы SIMATIC WinCC Open Architecture

А.Ю. Серов (ООО «Сименс»)

Приводится обзор возможностей, аспекты использования и особенности реализации JavaScript-интерфейса, применяемого в качестве возможного средства разработки пользовательских интерфейсов на базе SCADA-системы SIMATIC WinCC Open Architecture (WinCC OA).

Ключевые слова: SCADA-система, JavaScript, пользовательский интерфейс, экранные формы.

SCADA-система SIMATIC WinCC Open Architecture (WinCC OA) [1–3] является частью семейства продуктов компании Siemens, предназначенных для сбора, обработки, визуализации данных и диспетчерского управления. Это гибкая и адаптивная платформа для создания решений различного класса от небольших автономных систем до комплексных крупномасштабных проектов со специфическими функциональными и архитектурными требованиями, в том числе и к пользовательскому интерфейсу.

Пользовательский интерфейс (UI) — это один из функциональных компонентов WinCC OA, архитектурно построенной по модульному принципу и использующий различные технологии визуализации для отображения экранных форм, мнемосхем, пользовательских диалогов, виджетов и других графических элементов.

Одним из нововведений WinCC OA вер. 3.16 в части средств построения пользовательских интерфейсов является дальнейшее развитие JavaScript-интерфейса для реализации пользовательской логики на JavaScript и интеграции различных библиотек JavaScript с проектами WinCC OA. Отметим, что JavaScript до сих пор является современным, не утратившим актуальности языком, широко применяемым в мире для создания интерактивных Web-интерфейсов. Он поддерживает как объектно-ориентированный, так и процедурный стили программирования.

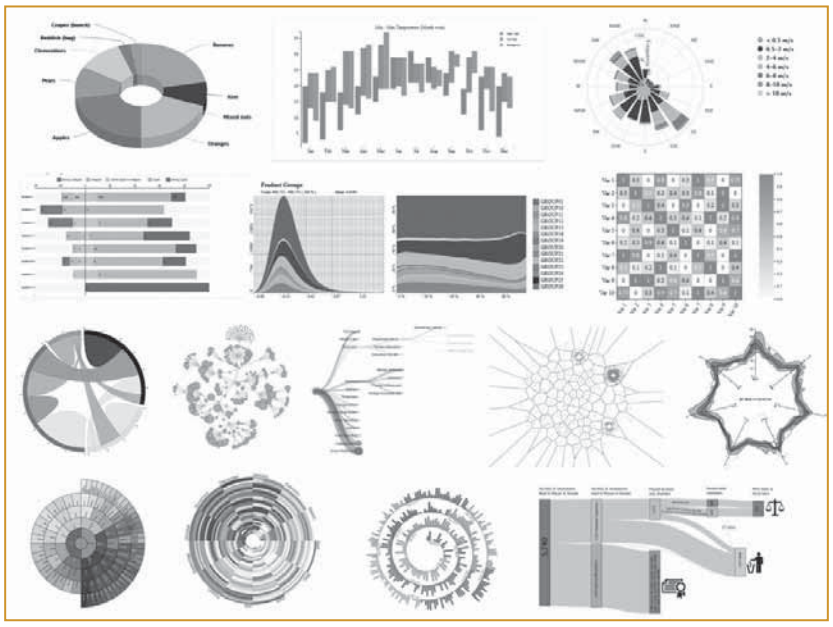

Рис. 1. Примеры объектов библиотек JavaScript

При этом сама платформа WinCC OA обладает развитыми средствами построения графических пользовательских интерфейсов (в том числе благодаря кроссплатформенному фреймворку Qt), однако добавление возможности разработки UI на JavaScript открывает для разработчиков дополнительные преимущества, например:

- применение JavaScript-сценариев непосредственно на экранных формах WinCC OA с интерфейсом взаимодействия с платформой, предоставляющим определенный набор методов непосредственного доступа из сценариев к базе данных и к значениям/атрибутам элементов экранных форм, аналогичных соответствующим функциям встроенного языка CTRL++, что дает сравнимый с последним масштаб применения;
- использование имеющихся или разработка собственных пользовательских программных реализаций на JavaScript позволяет снизить издержки на инжиниринг;
- использование готовых многочисленных библиотек и решений на JavaScript из Internet.

Среди существующих доступных на рынке библиотек JavaScript, функциональность которых может быть востребована в проектах WinCC OA, отметим такие известные реализации, как D3 (https://d3js.org), HIGHCHARTS (http://www.highcharts.com), JS Charts (http://www.jscharts.com), Protovis (http://mbostock.github.io/protovis), Flot (http://www.flotcharts.org) и др.

Указанные библиотеки позволяют реализовать развитую визуализацию, например, различные виды графиков (линейные, круговые, лепестковые, полярные, кольцевые и др.), гистограммы, пузырьковые диаграммы, календарные представления/карты, древовидные диаграммы, дендрограммы, потоковые графики, а также такие специализированные формы представления информации, как диаграмма Вороного, диаграмма Сэнки, Гамильтонов граф, матрица совместной встречаемости и др. Кроме того, специализированные библиотеки JavaScript предоставляют возможности для создания интерактивной визуализации (анимированных графов, интерактивных диаграмм и т. п.), а также таких востребованных в условиях пере-

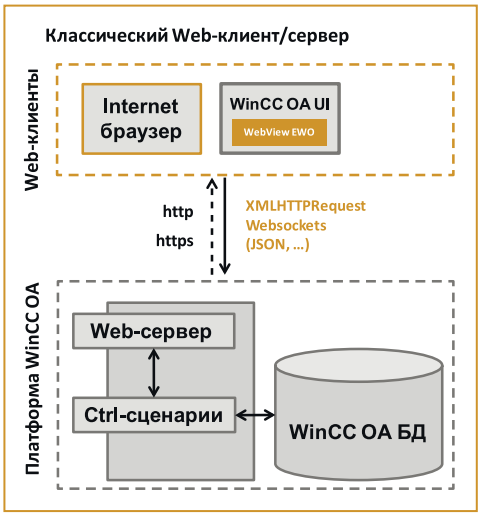

Рис. 2. Архитектура типичного Web-приложения на платформе WinCC OA

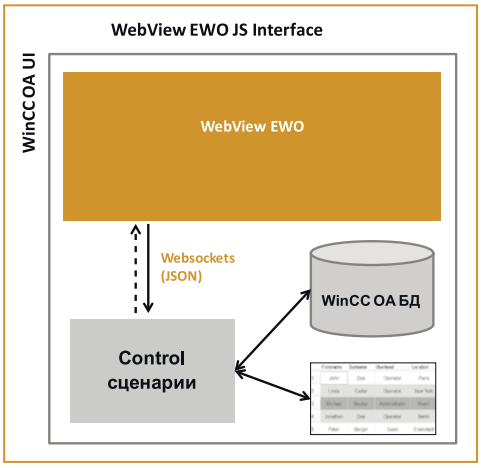

Рис.3. Архитектура JavaScript-приложения на экранной форме WinCC OA

грузки человека-оператора избыточной информацией элементов пользовательского интерфейса, как спарклайны или искрографики (sparkline) — небольшие линейные графики, дающие представление об общих трендах и занимающие минимум места на экране. Примеры внешнего вида некоторых объектов приведены на рис. 1.

JavaScript-интерфейс

Интеграция библиотек JavaScript на экранных формах WinCC OA осуществляется с помощью специального контейнера — виджета WebView EWO, что представляется вполне естественным решением, так как данный виджет применяется, в том числе и в качестве встроенного браузера для просмотра Web-страниц. Поэтому разработчикам платформы требовалось только предоставить пользователям интерфейс взаимодействия JavaScript с объектами самой системы. При этом возможно использование нескольких виджетов WebView EWO (способных обмениваться данными, в том числе и между собой) в рамках каждой экранной формы для обеспечения совершенно различной функциональности.

Аспекты использования JavaScript в WinCC OA. Сравнение архитектур

Традиционно типичные Web-приложения создавались с помощью встроенного Web-сервера WinCC OA. Сегодня появилась возможность использовать новую функциональность, доступную в виджете браузера — WebView EWO, которая позволяет использовать JavaScript на экранных формах, создавая, по сути, Web-приложения непосредственно в рамках UI. Различия обеих архитектур проиллюстрированы на рис. 2 и. 3.

Неотъемлемая часть архитектуры типичного Web-приложения на платформе WinCC OA (рис. 2) — это Web-сервер WinCC OA, способный генерировать по запросу динамические html-страницы с помощью пользовательских CTRL-сценариев (имеющих доступ к БД). Передача данных может осуществляться как обычным способом (HTTPRequest/HTTPResponse), так и с использованием различных подходов и идеологий, например, технологии обращения к серверу без перезагрузки страницы (AJAX) на основе метода XMLHTTPRequest, технологии Websockets, дополнительной сериализации данных в формате JSON и пр. В качестве клиентов могут выступать отдельный Web-браузер или встроенный виджет WebView.

Особенность архитектуры JavaScript-приложения на экранной форме WinCC OA (рис. 3) — отсутствие Web-сервера и взаимодействие ПО-компонентов исключительно в рамках экранной формы. Специальные JavaScript-методы транслируются в соответствующие CTRL-методы с помощью Websockets, а CTRL-методы штатно исполняются. Кроме того, собственные пользовательские JavaScript-методы также могут вызываться из CTRL-методов. Таким образом, обеспечивается практически полная интеграция JavaScript как языка программирования UI с WinCC OA.

Реализация JavaScript-интерфейса, принципы обмена данными

JavaScript-интерфейс виджета WebView EWO — точка взаимодействия пользовательских JavaScript-сценариев с компонентами WinCC OA, и наоборот. Интерфейс реализован с помощью библиотеки OAJsApi, которая содержит набор функций для подключения и работы с объектами WinCC OA.

JavaScript-методы, оттранслированные в методы CTRL-скрипта (при передаче сериализуются в JSON), получают доступ к нужным объектам WinCC OA. Активируется интерфейс с помощью метода WebView.loadSnippet (url), который загружает HTML-сниппет, представляющий собой пользовательскую html-страничку (содержащую JavaScript и прочие html-объекты при необходимости), которую нужно отображать на экранной форме. Пользовательские сценарии на html-страничке имеют доступ к функциям из библиотеки OAJsApi.

На самой панели может размещаться несколько таких виджетов с разной функциональностью, причем между этими виджетами также возможно взаимодействие (JavaScript-метод одного виджета может вызвать

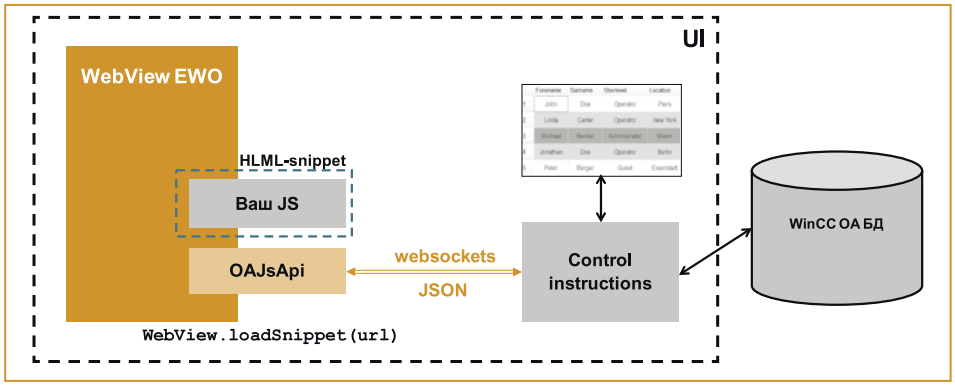

Рис. 4. Реализация JavaScript-интерфейса

JavaScript-метод другого виджета). На рис. 4 для простоты изображен только один такой виджет. По сравнению с Web-клиент/Web-сервер-архитектурой здесь обеспечивается высокая скорость взаимодействия между компонентами за счет использования технологии WebSockets и того факта, что эти компоненты находятся всегда в рамках одной машины.

Почему используется WebSockets, а не привычный AJAX? Цель применения WebSockets — обеспечить низкозатратное, двунаправленное, полнодуплексное и долговременное соединение между браузером и сервером. Это особенно необходимо при использовании, например, функции подписки на изменения значений тегов — dpConnect ().

Кроме того, в отличие от HTTP-взаимодействия Web-сокеты не имеют ограничений на время жизни в неактивном состоянии, то есть нет необходимости периодически обновлять соединение. Кроме того, поддерживаются кросс-доменные приложения, что может быть важно, так как Web-сервер, первоначально загружающий HTML-сниппет, может располагаться на удаленной машине.

При работе WebSockets ни HTTP, ни HTTPS не используются, они нужны только при первоначальном подключении (клиент-серверном «рукопожатии»). WebSockets, как протокол связи поверх TCP-соединения, предназначенный для обмена сообщениями между веб-клиентом и веб-сервером в режиме реального времени – это неплохой пример клиент-серверного взаимодействия, лишенного недостатков типичного веб-приложения, для которого характерно сравнительно медленное взаимодействие с HTTPRequest/HTTPResponse по http/https протоколу.

Сравнение архитектур. Обобщение

С учетом вышесказанного сделаем некоторые обобщения и выводы.

Решение с использованием архитектуры типичного Web-приложения на платформе WinCC OA обычно используется для реализации тонких клиентов и может вполне заменить собой штатного ULC UX-клиента. К дополнительным преимуществам решений данного типа относится возможность создания собственных пользовательских Web-приложений совершенно различных, в том числе и распределенных, архитектур и с применением различных подходов и технологий (например, AJAX со своей идеологией обмена данными в фоновом режиме).

Целевым применением решения с использованием JavaScript-интерфейса является расширение функциональности толстого клиента за счет эффективной интеграции JavaScript и нативного языка WinCC OA CTRL++. К дополнительным преимуществам подобных решений относится возможность разработки экранных форм практически целиком на Javascript, без необходимости думать, как именно организовать взаимодействие html-части с БД и графическими объектами WinCC OA, так как в JavaScript-интерфейсе уже реализованы все нужные методы доступа.

Таким образом, с появлением функциональности, позволяющей размещать объекты JavaScript на экранных формах WinCC OA совместно с нативными графическими объектами WinCC OA, разработчики получили в свое распоряжение новый актуальный инструментарий и средство визуализации для удовлетворения различных технических требований к пользовательскому интерфейсу WinCC OA как в части графического представления, отвечающего современным тенденциям и стандартам в области HMI, так и в части обеспечения производительности.

Список литературы

1. *Серов А.Ю., Соловьев С.Ю.* Интеллектуальные системы управления транспортной инфраструктурой на базе SIMATIC WinCC Open Architecture: возможности видео- и интерактивной картографии // Автоматизация в промышленности. 2018. №4.
2. *Мельников А.С., Соловьев С.Ю.* Обеспечение информационной безопасности при применении SCADA-системы SIMATIC WinCC Open Architecture // Автоматизация в промышленности. 2017. №7.
3. *Серов А.Ю., Соловьев С.Ю.* Применение SCADA-системы WinCC Open Architecture в нефтегазовой отрасли // Автоматизация в промышленности. 2017. №3.

Серов Андрей Юрьевич *– ведущий инженер по интеграции проектов, ООО «Сименс».*
Контактный телефон +7 (495) 737-1-737.
E-mail: winccoa.ru@siemens.com
icc.ru@siemens.com