

ИСПОЛЬЗОВАНИЕ КЛИЕНТОВ НА ОСНОВЕ WEB-БРАУЗЕРОВ И СЕРВИС-БРАУЗЕРОВ В КРУПНОМАСШТАБНЫХ СИСТЕМАХ

Орлов В.Л., Курако Е.А.

*Институт проблем управления им. В.А. Трапезникова РАН,
Россия, г. Москва ул. Профсоюзная д.65
ovl@ipu.ru, kea@ipu.ru*

Аннотация: рассматривается вопрос использования для крупномасштабных систем тех или иных типов клиентов. В качестве основных клиентов определяются программно-аппаратные устройства, построенные на web-браузерах и сервис-браузерах. Проводится анализ использования подобных устройств и сравнение их возможностей.

Ключевые слова: клиент, web-браузер, сервис-браузер, информационная система.

Введение

Вопрос использования в крупномасштабных системах тех или иных типов клиентов стоит как никогда остро. Ранее (а это было сравнительно недавно) в процессе разработки для каждого типа рабочего места, например, для «Администратора», «Руководителя», «Менеджера» создавалось свое программное обеспечение, которое опиралось на сетевые протоколы, определяющие связи в системе. Однако, достаточно скоро стало понятно, что этот путь не является оптимальным, так как его использование приводило к созданию для каждого рабочего места дублирующих функций, в основном связанных с обеспечением надежности передачи данных и обеспечением защиты информации.

Таким образом, появлялась необходимость создания протоколов более высокого уровня и введения принципа инкапсуляции нижних уровней. Но так как для разных систем требовались разные степени надежности и защиты, то в результате разрабатывались различные клиентские системы для конкретного применения.

Вместе с тем возникала еще одна проблема, связанная с организацией сопровождения программного обеспечения. Она заключается в том, что при наличии различных клиентов резко увеличиваются трудозатраты, направленные на внесение изменений в системы в процессе эксплуатации. То есть при коррекции необходимо осуществлять множественные рассылки с учетом особенностей каждого рабочего места, или по крайней мере типов рабочих мест.

Все это создает безусловные эксплуатационные неудобства. Решением этой проблемы является концентрация данных и программного обеспечения в серверном центре с тем, чтобы клиенты могли вызывать информацию и программы с серверов, проводить отображение и коррекцию на клиенте, выполняя в том числе вызовы серверных программ и запоминание скорректированных объектов путем передачи их в центр.

Традиционно способы решения указанной задачи подразумевают использование web-браузера. Но вместе с тем возможны и другие подходы, которые рассматриваются ниже.

1 Особенности организации клиента на основе web-браузера

В простейшем случае web-браузер отображает содержимое html-страницы, полученной с сервера методом GET. Если же нужно передать данные на сервер, то для этого, в основном, предназначен метод POST. То есть web-браузер может выполнять две основные функции клиента: получение информации с сервера и передачу запросов [1].

Безусловным достоинством данного способа является его универсальность. Он основывается на протоколе http, который интерпретируется браузером и не зависит от конкретной операционной системы. Но при этом нужно помнить, что от операционной системы зависит реализация web-браузера, который выполняется на клиенте. Достоинством этого метода является то обстоятельство, что на клиенте просто нет пользовательских программ. Они все хранятся на сервере, вызываются, отображаются в виде html-страниц, причем в отображении имеются элементы, которые позволяют сформировать другие запросы к серверу. Понятно, что в таком простейшем варианте организация клиента возможна, но он будет примитивным по определению. Ведь форма отображаемой страницы может быть сложной и насыщенной, но запрос представляет собой статическую структуру. Если мы хотим обеспечить разнообразие запросов, то должны иметь в виду, что каждый новый запрос, отличающийся по форме от предыдущего, должен иметь свою форму. И эта форма должна создаваться на сервере и каждый раз передаваться клиенту.

Недостатки здесь очевидны. И в первую очередь они связаны с созданием запросных форм. Если запрос включает достаточно много параметров, то пользователь не имеет возможности проверить правильность заполнения, пока не введет все данные и не отправит запрос на сервер. Но если заполняемых полей много, то это затрудняет диалог. Логичным выходом было бы проверять поля сразу после заполнения. Обычно для этой цели используется JavaScript [2], применение которого усложняет форму, но позволяет считывать и проверять данные по справочным значениям. Фактически таким образом мы работаем с фрагментами формы.

Отметим, что JavaScript по существу является языком программирования и таким образом на html-странице присутствует и описание формы отображения, и выполняемые программы, которые при загрузке страницы считываются с сервера.

Рассмотрим основные направления развития web-браузеров. Прежде всего это расширения за счет использования фреймворков. Наиболее известны клиентские фреймворки – Angular, Vue, Svelte, React, написанные на языке JavaScript. Мы в данном случае рассматриваем клиентскую часть, поэтому на этом можно и остановиться. Но нужно иметь в виду, что на серверах также используют свои фреймворки серверные – это Django - язык Python, Laravel, Symfony – язык PHP и другие.

То есть здесь важно то, что по мере развития web-технологий центр тяжести смещается в сторону программных компонентов. И если первые web-браузеры в большинстве своем были ориентированы на отображение графики и текста, то современные активно работают с программами.

Еще один важный момент. Очевидно, понимая, что центр тяжести на html-страницах смещается в сторону программных модулей, а программные модули обычно не модернизируются от вызова к вызову, разработчики стали помещать все скрипты на одну страницу. Обычно это обозначается как SPA (приложение одной страницы). Так возникли «одностраничные» сайты [3], в которых вся обработка происходит на одной странице путем вызова тех или иных программных фрагментов. Этим достигается следующий эффект. В таких сайтах загрузка программ происходит однократно, а выполнение – в соответствии с заданным алгоритмом. Причем одни и те же фрагменты могут выполняться многократно. Это позволяет повысить быстродействие. Нужно отметить, порталы могут содержать не один, а несколько одностраничных сайтов, объединяя их в группу. Также заметим, что возможна загрузка не всех скриптов, а «загрузка по требованию», что соответствует динамической загрузке модулей при использовании традиционных языков программирования.

Для эффективного вызова серверных модулей используются сервисы (например, web- или rest-сервисы).

Обратим внимание на тенденции развития клиентов на основе web-браузеров. Следует выделить следующее:

- современные web-браузерные клиенты наряду с элементами отображения содержат все больше программных элементов, на основе языка JavaScript и дополняющих его различных фреймворков;
- для вызова серверных модулей часто используется механизм вызова web-сервисов наряду с вызовами сервисов другого типа;
- просматривается тенденция формирования клиентских модулей в виде одностраничных сайтов, у которых появляется возможность как элементы отображения, так и программы загрузить с сервера однократно, а не погружать страницы при каждом вызове.

Последовательность работы клиента, ориентированного на web-браузер, включает следующие действия:

- пользователь на основании URL (унифицированного указателя ресурса) вызывает с сервера начальную html-страницу системы;
- обычно на уровне прикладной программы клиента проводится аутентификация пользователя;
- после аутентификации проводится обмен данными между клиентом и сервером, причем в современном исполнении организация обмена идет таким образом, что количество клиент-серверных обменов стараются минимизировать, используя JavaScript и SPA-технологии.

2 Организация клиента на основе сервис-браузеров

Вместе с тем, рассматривая тенденции развития web-браузерных клиентов, следует обратить внимание на клиентов другого типа – сервис-браузерных клиентов [4, 5].

Основное принципиальное отличие web-браузеров от сервис-браузеров состоит в том, что для web-браузеров главным является загрузка и интерпретация содержимого html – страниц (т.е. данных), а для сервис-браузеров вызовы сервисов.

Конечно, как мы видели выше, html- страницы могут включать в свой состав программы (например, в виде JavaScript-ов), а сервисы могут передавать практически только данные. Но вместе с тем важен начальный импульс.

Обратим внимание, на каких принципах строятся сервис-браузеры. Каждый сервис-браузер выполняет три основные функции:

- осуществляет связь со своими служебными сервисами;
- определяет необходимые базы данных и системы обработки информации, которыми мы хотим пользоваться;
- осуществляет загрузку, обновление и активизацию прикладных модулей выбранной системы обработки информации.

То есть сервис-браузер на начальной стадии обеспечивает только работу со своими служебными сервисами и управляющей базой данных. Прикладные программы и данные не включены в состав сервис-браузера. Они определяются только выбором системы с использованием управляющей базы данных.

Но если управляющая база данных содержит ссылки на конкретные прикладные системы и конкретные базы данных, то для осуществления доступа к этим ресурсам необходимо обеспечить разрешения для тех или иных пользователей на их применение. Поэтому сервис-браузер уже на начальной стадии должен иметь встроенный механизм защиты информации. При этом необходимо:

- провести аутентификацию пользователя, то есть проверку его подлинности;
- провести авторизацию, то есть проверку прав на выполнение тех или иных действий.

То есть конкретный пользователь имеет доступ к определенным системам и определенным модулям этих систем. Кроме того, в рамках использования системы пользователь работает с заданной базой данных, структура которой определяется выбранной системой.

Последовательность работы клиента, основанного на использовании сервис-браузерной технологии, определяется следующим порядком действий:

- пользователь на основании уникального имени сервера осуществляет подключение к служебным сервисам;
- проводится аутентификация пользователя;
- после проведения аутентификации определяется одна или несколько систем, которые может использовать проверенный пользователь;
- пользователь выбирает систему, которую он собирается использовать;
- на основании выбранной системы и идентификатора пользователя автоматически выбирается база данных, с которой работает данный пользователь (или группа пользователей), а также в соответствии с настройкой определяется набор модулей, которые может выполнять данный пользователь, и их роли при использовании;
- пользователю предоставляется список модулей, которые он имеет право вызывать;
- после выбора определенного модуля осуществляется вызов этого модуля и передача ему управления.

3 Сравнение клиентов

Для сравнения рассмотрим основные направления развития web-браузерного клиента, отметим задачи, решаемые в процессе этого развития, и определим, как эти же задачи решаются в клиенте, построенном на основе сервис-браузера.

3.1 Первое направление развития

Итак, первое направление развития. Оно проистекает из необходимости сложных или множественных обращений к серверу при наличии многопараметрических форм ввода, то есть форм ввода, которые имеют много полей. Действительно, при классическом подходе, использующим кнопку «Submit» мы должны передать одновременно все введенные параметры на сервер, а далее уже на сервере заниматься проверкой полученных данных, формировать множество ответов, передавать их клиенту, размещать замечания на исходной форме и удалять замечания при повторной передаче (вариант сложного обращения). Все это явно неудобно.

Вариант множественного обращения подразумевает после заполнения какого-либо поля передачу формы ввода на сервер, проверку заполненного поля и возвращения единственного ответа,

касающегося только что обработанного поля. В этом случае, чтобы корректно заполнить форму, необходимо много раз обращаться к серверу и обрабатывать ответы.

То есть становится очевидным, что без введения каких-либо программ непосредственно на html страницу – не обойтись.

Обычно эти программы проводят проверку данных, введенных в каждое поле, без обращения к серверу. Чтобы исключить такое обращение, при формировании клиентской страницы на ней размещаются перечни допустимых значений параметров и JavaScript, осуществляющий проверку. В принципе таких скриптов может быть несколько, особенно если проверки разнотипные.

Как же в этом случае действует клиент, работающий на основе сервис-браузера? Так как такой клиент по определению подразумевает наличие программного модуля, который загружается с сервера, то этот программный модуль включает необходимые программы проверки.

Более того, эти программы могут быть скорректированы при очередном обновлении и (что важно) могут не загружаться на данное клиентское место, если обновлений не было.

3.2 Второе направление развития

Второе направление развития web-браузеров связано со сложной обработкой данных на клиенте. Это направление определяется наличием фреймворков, которые в процессе развития, по существу, начинают представлять собой библиотеки сложной обработки данных разного типа. Клиентские фреймворки выполняются в браузере. Они позволяют разрабатывать оригинальные интерфейсы пользователя, выполнять анимацию и обеспечивать ряд других функций.

В сервис-браузере программные модули разрабатываются с использованием языков программирования. Современные языки программирования всегда содержат обширную базу встраиваемых функций. Следует отметить, что помимо функциональных баз здесь также могут использоваться дополнительные функциональные библиотеки различных разработчиков. Таким образом, активное использование фреймворков для сервис-браузера предусматривается с самого начала.

3.3 Третье направление развития

Третье направление развития web-технологий на клиенте связано с созданием одностраничных клиентов. Это важное направление. Оно позволяет разместить все скрипты, имеющие отношение к определенному направлению обработки информации, на одной html- странице. А это означает, что нет необходимости загружать поочередно множество страниц, достаточно произвести однократную загрузку данной страницы и вызывать те или иные скрипты с этой страницы по мере необходимости.

То есть эта работа, по существу, соответствует работе обычного программного модуля. Программный модуль обычно содержит также набор функций, которые также могут вызываться произвольным образом в соответствии с внутренним алгоритмом. Программный модуль содержит данные, которые представляют собой базовые фрагменты для отображения различных форм, текстов и графики. Вместе с тем все эти данные могут динамически перестраиваться, меняя текущее изображение. Итогом реализации этого направления развития является повышения быстродействия работы клиента.

В итоге web-браузер начинает работать по алгоритму, близкому к алгоритму работы сервис-браузера. Разница в том, что web-браузер загружает и направляет на исполнение программы, в основном оформленные в виде JavaScript-ов, а сервис-браузер – программы, написанные на различных языках программирования.

3.4 Мультиязычность сервис-браузера

Следует напомнить, что мы работаем с клиентом. Клиент, реализуемый на основе web-браузера, имеет определенные языковые ограничения.

Все программы, исполняемые web-браузером, должны быть написаны на одном языке – JavaScript. Эти программы поступают на выполнение в виде интерпретируемого кода.

Программы сервис-браузера могут быть написаны на разных языках. По существу, к этим языкам может быть предъявлено одно требование: они должны обеспечивать работу с различными сервисами и в первую очередь с SOAP и REST-сервисами.

Что важно, клиентская и серверная части могут быть написаны на одном языке. Это обстоятельство является важным для определенных команд программистов. Кроме того, этот язык в принципе может предусматривать трансляцию, и таким образом, обеспечивать большее быстродействие для выполняющихся программ.

Заключение

Для крупномасштабных систем очень важна организация клиентов, использующих определенные технологические принципы. Наиболее эффективным в настоящее время является использование web-технологий. Среди них можно выделить технологии, обеспечиваемые web-браузерами и технологии, обеспечиваемые сервис-браузерами.

Web-браузерные клиенты в настоящее время широко распространены. Вместе с тем они не лишены недостатков и постоянно развиваются. Если проследить основную линию развития, то можно обнаружить интересную тенденцию, в соответствии с которой происходит сближение технологий. Причем сближение это идет путем постепенного преобразования web-систем в направлении перехода к увеличению доли программного кода в html-страницах. А так как сервис браузеры ориентированы на использование в основном программного кода, то они получают определенное преимущество, что дает им возможность опередить по ряду показателей web-браузеры.

Литература

1. Кришнамурти Б., Рексфорд Дж. Web-протоколы. Теория и практика. – М.: ЗАО «Издательство Бином», 2002. – 592с.
2. Крокфорд Д. Как устроен JavaScript. – СПб.: Питер, 2019 - 304с.
3. Миковски М.С., Пауэлл Дж.К. Разработка одностраничных веб-приложений. – М.: // ДМК Пресс, 2018. – 512с.
4. Курако Е.А., Орлов В.Л. Сервис-браузеры для информационных систем // Программная инженерия. – М. 2017. – Т. 8, № 9. – С. 413-421.
5. Kurako E.A. Orlov V.L. Service-browser Architecture and Large-scale Information Systems / Proceedings of the 11th International Conference "Management of Large-Scale System Development" (MLSD). Moscow: IEEE, 2018. С. <https://ieeexplore.ieee.org/document/8551831>.