

ИССЛЕДОВАНИЕ МОДЕЛЕЙ АНАЛИЗА КОМПОЗИТНЫХ ВЕБ-СЕРВИСОВ НА ОСНОВЕ JSON

Скичко Анатолий Викторович

Мартынюк Александр Николаевич, к.т.н., доцент

Композитный веб-сервис (Mashup) представляет собой составное приложение, которое объединяет два или более существующих компонентов, доступных в Интернете. Этими компонентами могут быть данные, логика приложения или пользовательские интерфейсы. Стек приложения показан на уровне данных, логики и представления (пользовательского интерфейса). Созданный таким образом веб-сервис называется или mashup данных, или логикой, или интерфейсом пользователя.

Компоненты Mashup являются его строительными блоками. На практике в разработке компонентов mashup используются несколько технологий и стандартов. Простые веб-службы протокола доступа к ресурсам (SOAP), веб-службы RESTful, API Javascript, Really Simple Syndication (RSS), Comma-Separated Values (CSV) и т. д. являются одними из основных. В зависимости от их функциональности компоненты mashup были в целом разделены на три категории (рис. 1):

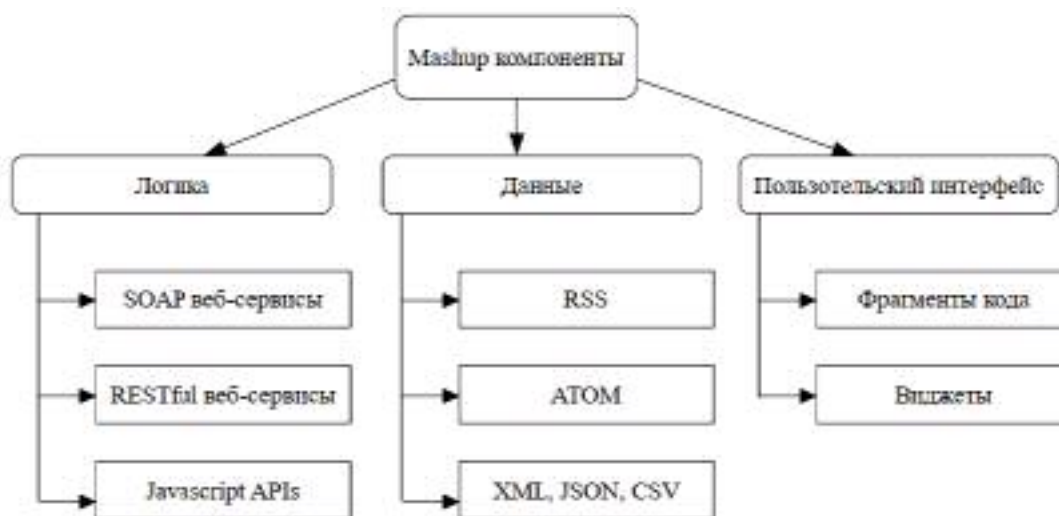


Рисунок 1 — Классификация компонентов Mashup

Для того, чтобы внутренние сервисы могли взаимодействовать между собой, синхронизироваться, используется технология AJAX — асинхронный обмен данными. С её помощью компоненты Mashup приложения быстро и, что немало важно, незаметно для пользователя выводят уже обработанную информацию на страницу без перезагрузки. На сегодняшний день данную технологию очень удобно использовать с архитектурным стилем REST. Он поддерживает несколько операций: POST, GET, PUT, DELETE, которые приемлемы для всех типов данных.

Весомой проблемой является достаточное количество разработанных Mashup с обменом данных в формате XML — он имеет избыточный синтаксис, размер документа значительно больше, нежели в JSON и, следовательно, данные передаются медленнее. В современном мире критически важна скорость обработки запросов и передаваемых данных.

Важными показателями сервиса mashup являются быстродействие и ресурсоёмкость. Быстродействие проявляется во времени отклика — время, прошедшее с момента подачи запроса до получения ответа и в пропускной способности — число выполненных запросов в единицу времени. Ресурсоёмкость можно увидеть по требуемой памяти и использованию процессора.

Целью данной статьи является построение модели анализа композитного веб-сервиса на основе JSON, позволяющего не только учесть особенности построения составных сервисов, но и связанную с веб-средой неопределенность в работе сервисов.

Задачей является моделирование с помощью сетей Петри картографического композитного веб-сервиса. Для наглядного примера mashup на рисунке 2 показана модель такого веб сервиса:

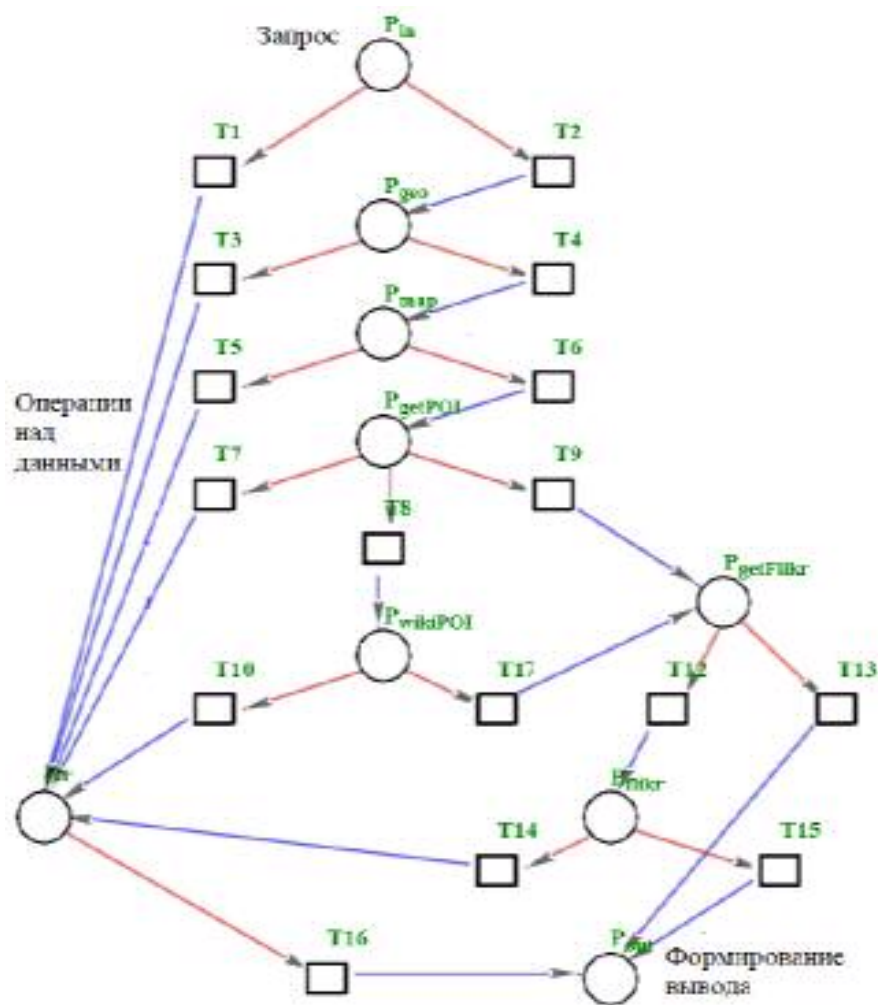


Рисунок 2 — Общая модель mashup в виде сети Петри-Маркова

В сети Петри (рис.2) запрос — это метка, находящаяся в позиции P_{in} . При попадании метки в позицию возбуждаются два перехода $t1$ и $t2$, то есть может сработать любой из них.

В представленной выше модели выбор операции происходит после срабатывания перехода $t2$, и попадания метки (метку также можно назвать фишкой) в позицию P_{geo} , после чего идёт запрос к сервису Google Map для получения геоданных местоположения (долгота и широта) заданного физического адреса.

Далее срабатывает переход $t4$ и происходит загрузка карты с помощью сервиса OpenStreetMap.

После срабатывания перехода $t6$, в случае если пользователь

отключил компонент POI (Points of Interest) запрашивается разрешение на получение адресов фотографий с веб сервиса Flickr. После чего идёт запрос уже на их получение и результат выводится пользователю.

Преимущества данного mashup в том, что пользователь сам может управлять компонентами, которые ему требуются. Так же все данные между компонентами передаются в формате JSON, что значительно ускоряет вывод результата и не заставит ждать пользователя.

Таким образом была построена гибкая модель композитного веб-сервиса быстрой передачей данных. Этот mashup не зависит от платформы, так как выполняется в браузере пользователя. Он удобен в использовании при поездках на отдых, если местность не знакома. Так же данное приложение можно довольно быстро масштабировать, подключив необходимые компоненты.