Перевод статьи Mesbah A., Bozdag E., Van Deursen A. Crawling Ajax by inferring user interface state changes //Web Engineering, 2008. ICWE'08. Eighth International Conference on. – IEEE, 2008. – С. 122-134.

Переведено предисловие и первая глава. Русских знаков без пробелов = 5406.

**Краулинг AJAX (асинхронный javascript) на основе изменения пользовательского интерфейса**

**Предисловие.**

AJAX является многообещающим подходом на пути к развитию интерактивности и отзывчивости веб-приложений.

В то же время, AJAX методы нарушают метафору «веб-страница», на основе которой все современные веб-краулеры работают. Данная статья описывает новую технику краулинга веб-приложений посредством динамического анализа и реконструкции изменений состояний пользовательского интерфейса. Наш метод динамически создаёт граф состояний и переходов моделируя различные пути обхода и состояния при этом в AJAX приложении. Созданная модель может быть использована для генерации статических слинкованных страниц. Эти страницы могут использоваться для предоставления основным поисковым движкам. Более того, мы считаем, что техники краулинга, являющиеся частью нашего решения имеют и другие приложения, например, в рамках основных поисковых движков, в рамках улучшения доступности, или для автоматического перебора всех пользовательских элементов интерфейс и поддержки тестирования на основе состояний AJAX приложения. Мы представляем вам наш продукт crawljax с открытым исходным кодом, который реализует концепции, обсуждаемые в этой статье. Дополнительно, мы дадим отчёт о применении нашего подхода к набору представительных AJAX веб-приложений и обсуждаем полученные результаты.

**Введение**

Известный нам веб сейчас переживает сильные изменения. Недавно появившаяся технология в следствие распространения Web 2.0 – это AJAX (асинхронный javascript и XML) [13], в котором хитроумная комбинация javascript и манипуляций объектной модели документа (DOM) вместе с асинхронным взаимодействием с сервером, используется для достижения высокого уровня пользовательской интерактивности. Одними из самых заметных примеров являются приложения Google Maps, Google документы и последняя версия Yahoo! Mail.

C новыми изменениями в разработке веб-приложений появляются целый набор новых задач, в основном связанных с тем, что AJAX ломает метафору «веб-страница», на которой основываются многие веб-технологии. Среди этих задач есть следующие:

1. **Возможность поиска**, чтобы добиться возможности индексирования AJAX сайтов основными поисковыми движками, вместо того чтобы быть просто проигнорированными (что сейчас часто и происходит), из-за использования скриптинга на стороне клиента и динамического изменения состояний DOM.
2. **Тестебельность**, чтобы систематично проверять динамический пользовательский интерфейс (UI)и состояния AJAX, чтобы находить ошибки и аномалии.
3. **Доступность**, чтобы проверить, что все состояния AJAX страницы удовлетворяет необходимым требованиям доступности.

Единственный способ удовлетворить эти требования – это использование краулера, который может автоматически обходить различные состояния крайне динамичного AJAX сайта, создавать модель возможных путей выполнения и потенциальных состояний, и генерировать традиционную связную статическую версию сайта на основе статических страниц. Сгенерированные статические страницы могут использоваться, например, чтобы показать AJAX сайт основным поисковым движкам, или чтобы проверить доступность различных динамических состояний [2]. Подобный краулер так же может быть использован для способствования тестированию состояний асинхронных приложений [16], и автоматического тестирования всех элементов пользовательского интерфейса асинхронного сайта, чтобы найти, например, сломанные ссылки, скрытые ссылки и другие ошибки.

Сегодня не существует краулера, который способ обрабатывать сложный код на клиентской стороне асинхронного приложения. Основная причина в том, что краулить AJAX фундоментально тяжелее, по сравнению с классическими многостраничными веб-приложениями. В традиционных веб-приложениях, состояния отделены и соответствуют страницам с уникальным url. В асинхронных приложениях, тем не менее, состояние пользовательского интерфейса определяется динамически, также как и изменения в DOM, которые видны только лишь после выполнения соответствующего javascript кода.

В данной статье мы описываем подход для анализа и реконструкции состояний пользовательского интерфейса автоматически. Наш подход основан на краулере, который способен выполнять код на клиентской стороне, и способен идентифицировать кликабельные элементы (которые могут изменяться вместе с каждым кликом), которые могут изменять динамическое состояние DOM в рамках браузера. Из этих изменений мы восстанавливаем граф потока-состояний, который фиксирует состояния пользовательского интерфейса и возможные переходы между ними. Этот граф способен впоследствии быть использован для генерации много-страничной статической версии оригинального асинхронного приложения.

Идеи лежащие в основе реализованы в инструменте под названием «Crawljax».

Мы провели эксперимент по запуску нашего фреймвокра-краулера на нескольких репрезентативных асинхронных сайтах, чтобы проанализировать общую производительность нашего подхода, вычислить эффективность выбора необходимых кликабельных элементов, оценить качество и корректность детектированных состояний и сгенерированных статических страниц, и проверить способность нашего инструмента на реальных сайтах используемых на практике и проверить масштабируемость краулинга сайтов с тысячами динамических состояний и кликабельных элементов.

Статья структурирована следующим образом. Мы начнём с того, что в Секции 2 расскажем о трудностях краулинга и индексирования AJAX. В секциях 3 и 4, мы представим подробный рассказ наших новых техник краулинга, процесса генерирования и самого инструмента Crawljax. В секции 5 результаты применения наших методов к нескольких AJAX приложениях, которые будут указаны, после чего в секции 6 будет описаны открытия и найденные возможности. Секция 7 представляет различные приложения наших техник краулинга. В заключении мы рассказываем о связанных с этой тематикой работах, кратким изложением ключевых достижений и предположения о дальнейшей работе.