УДК 004.55

СИСТЕМА СЕМАНТИЧЕСКОЙ ОПТИМИЗАЦИИ СОДЕРЖИМОГО ВЕБ-САЙТОВ НА ОСНОВЕ ПОЛЬЗОВАТЕЛЬСКИХ ПРЕДПОЧТЕНИЙ

П.И. Банокин, В.Н. Вичугов

Томский политехнический университет
E-mail: pavel805@gmail.com

Выявлена потребность в оптимизации содержимого веб-сайтов в соответствии с индивидуальными предпочтениями пользователей. Предложены способы оптимизации содержимого веб-страниц для различных сценариев использования приложения и способы хранения семантического профиля пользователя. Представлена реализация программной системы в виде набора JavaScript-компонентов и методика интеграции данной системы с существующими веб-приложениями.

Ключевые слова:
Семантическое веб-приложение, оптимизация содержимого веб-страниц, семантический профиль пользователя.

Key words:
Semantic web-application, web-page content optimization, semantic user profile.

Современные интернет-приложения содержат обновляющийся и разнообразный контент в неструктурированном виде. К таким сайтам относятся блоги, социальные сети, интернет-аукционы, фото- и видео-хостинги. В такой ситуации пользователям становится сложно ориентироваться в многообразии информации, представленной как на одной веб-странице, так и на целом веб-сайте.

Несмотря на популярность социальных сетей, поисковые системы являются основным источником новых посетителей для веб-сайтов. Существующие методы продвижения интернет-приложений в большей степени ориентированы на оптимизацию веб-страниц для поисковых систем. Довольно часто такие методы противоречат принципам удобства использования приложения. В результате создаются неестественные и сложные для понимания обычного пользователя заголовки страниц и названия ссылок, а качество интерфейса приложения и релевантность контента поисковому запросу снижаются.

В последнее время алгоритмы работы ведущих поисковых систем интернета изменились: поведение и предпочтения пользователя стали иметь большее значение при поисковом анализе веб-приложения [1]. Алгоритмы ранжирования веб-сайтов стали способны учитывать поведение пользователя на отдельной веб-странице [2]. Обычно выделяют следующие метрики для анализа поведения пользователей на веб-сайте: глубина просмотра, продолжительность визита, источник перехода, тип устройства, географическое положение пользователя. Поэтому возникла необходимость предоставления более качественного и релевантного предпочтениям пользователя контента, а также общее повышение уровня удовлетворенности от пользования приложением.

Задачей разработчиков веб-приложения становится предоставление максимально релевантной предпочтениям пользователя информации, тем самым улучшая метрики его поведения. Индивидуальная настройка внешнего вида и содержимого отдельной страницы в соответствии с интересами пользователя является одним из эффективных решений для достижения этой цели. Семантический анализатор, исполняемый на стороне клиента, и генератор семантических атрибутов, исполняемый на стороне сервера, могут производить семантический анализ информации и динамически настраивать расположение и оформление контента в соответствии с нуждами конкретного пользователя.

Предполагаемыми сферами применения подобной архитектуры могут быть приложения, работающие по принципу социальных сетей и отличающиеся большой разнородностью интересов пользователей [3].

Для создаваемой архитектуры были определены необходимые качества:

- Гибкость. Процесс интеграции должен требовать минимальной модификации логики приложения. Разработчик должен иметь возможность добавлять новые методы перестроения содержимого веб-страницы и новые алгоритмы ранжирования частей контента.
- Масштабируемость. Количество пользователей веб-приложения должно оказывать минимальное влияние на время выполнения операции семантического анализа и перестроения пользовательского интерфейса.
- Безопасность. Предпочтения пользователя должны храниться в не персонализированной форме.

Все перечисленные выше качества достигаются архитектурой семантического приложения (рис. 1). Все процессы, за исключением процесса генерации семантического HTML-кода, исполняются на стороне клиента.

В соответствии с основными процессами, представленными выше на диаграмме потоков данных (рис. 1), можно выделить четыре основных компонента создаваемой системы семантического анализа:

1) генератор семантического HTML-кода;

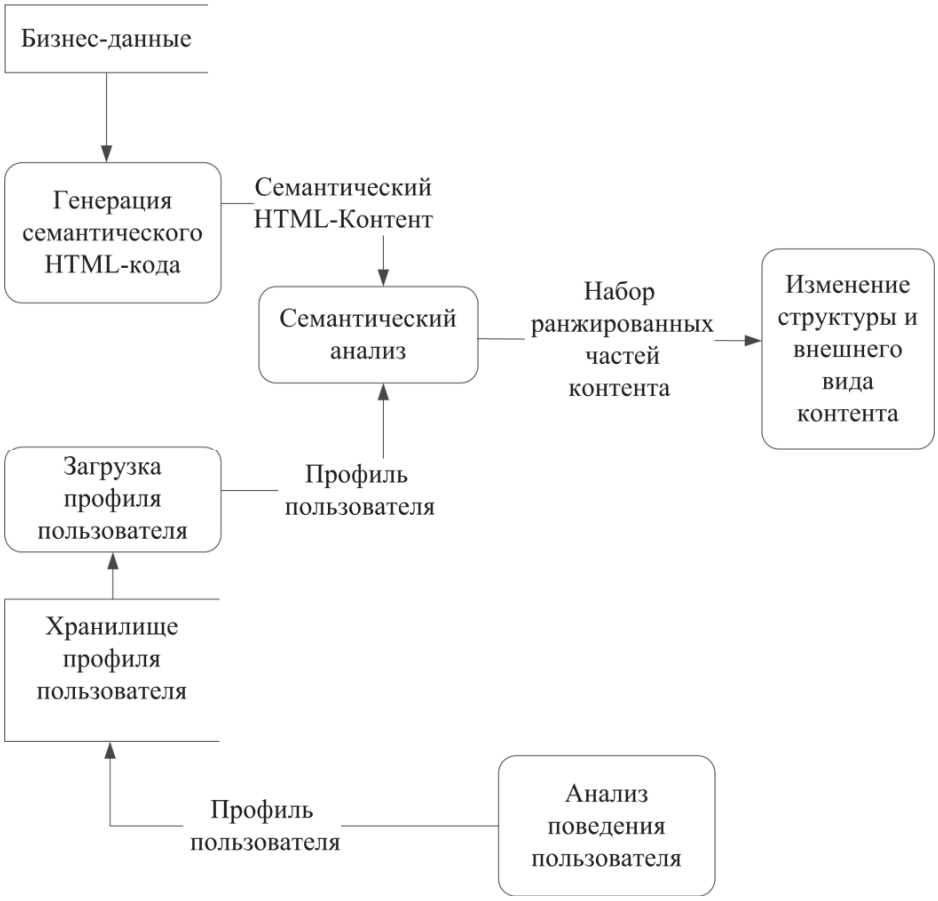

Рис. 1. *Диаграмма потоков данных системы семантической оптимизации веб-сайтов*

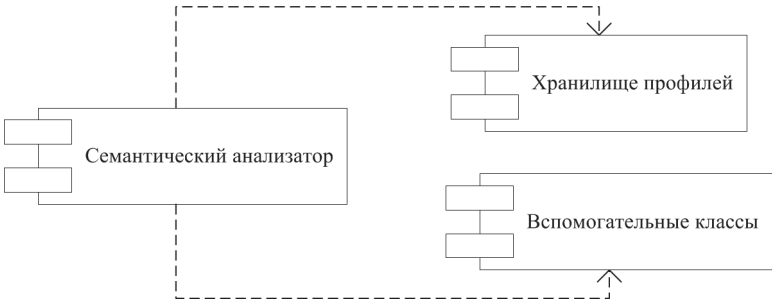

Рис. 2. *Диаграмма клиентских компонентов*

2) хранилище семантических профилей пользователя;
3) семантический анализатор;
4) библиотека сервисных функций.

Архитектура рассматриваемой системы является распределенной и состоит из двух физических узлов – веб-сервера (среды исполнения веб-приложения) и клиента. Компоненты программной системы, за исключением генератора семантического HTML-кода, исполняются интернет-браузером пользователя (рис. 2).

Генератор семантического HTML-кода – компонент архитектуры, создающий входные данные для семантического анализатора. Генератор семантического HTML-кода добавляет семантическую информацию к элементам контента, которые будут в дальнейшем проанализированы семантическим анализатором.

Добавление семантической информации к HTML-содержимому реализуется путем использования специальных атрибутов. При данном способе каждому семантически значимому HTML-тегу присваиваются специальные атрибуты, описывающие контент (рис. 3). Значение атрибута представляет собой строку, состоящую из набора слов (тегов), разделенных пробелом: «тег1 тег2 тег3 тег4». При данном способе форматирования можно задавать произвольное число тегов (строковых констант), характеризующих предпочтения пользователя. Данный подход может использоваться в соче-

```
<div semantic="moscow economics foodmarket">...</div>
```

Рис. 3. *Пример семантического атрибута*

тании с обычным разделением веб-страницы на семантические области, такие как область навигации и основная функциональность область [4].

В настоящее время большинство веб-приложений используют различные вариации MVC-архитектуры. Логика по форматированию и формированию значений семантических атрибутов должна реализовываться в слое представления веб-приложения.

При анализе HTML-документа компонент «Семантический анализатор» использует данные, созданные генератором семантического HTML-кода. Поэтому генератор семантического HTML-кода не имеет прямых отношений использования или ассоциации с другими компонентами архитектуры.

Семантический профиль пользователя – объект для хранения информации о предпочтениях пользователя приложения. Семантический профиль состоит из набора полей с уникальными названиями. Каждое поле имеет количественное значение, именуемое в дальнейшем оценкой поля, определяющее его релевантность к предпочтениям пользователя. В хранилище семантический профиль представляется в виде строки, состоящей из пар «поле: значение», разделенных пробелами. Примером такой строки может быть строка «*moscow*:16 *foodmarket*:7 *aircrafts*:10 *russia*:11».

Операции, которые могут выполняться над семантическим профилем:

а) Дополнение. К семантическому профилю пользователя добавляются новые поля (предпочтения). Если поле уже присутствует в профиле пользователя, то происходит увеличение оценки поля на требуемое значение.

б) Вычитание. Оценка полей профиля, перечисленных во входной строке, уменьшается на требуемое значение. В случае, если оценка поля стала отрицательной или равной нулю, такое поле исключается из профиля пользователя.

в) Удаление. Из профиля удаляются теги (предпочтения), перечисленные во входной строке.

Поставщик семантических профилей ответственен за формирование набора характеристик предпочтений пользователя. Хранилище семантического профиля является децентрализованным. Профиль пользователя хранится в объекте *HTML5 LocalStorage* [5]. Данный подход делает архитектуру приложения менее зависимой от конкретной исполняющей среды и СУБД.

Сценарии, при которых целесообразно использовать клиентское распределенное хранилище:

- Большинство пользователей используют персональные устройства для доступа к веб-приложению, такими как интернет-планшеты или коммуникаторы.
- Страницы веб-приложения просматриваются анонимно без использования регистрационных учетных записей.
- Посетители совершают множество действий во время посещения веб приложения. В результате семантический профиль, если он был утрачен в процессе очистки временных файлов или смены веб-браузера, быстро восстанавливается в процессе самообучения.

Следующие источники могут быть использованы для получения данных о пользователе: социальные сети, опросы (анкетирование) на веб-сайте, статистика поведения пользователя, данные о географическом положении пользователя.

Требования к хранилищу семантических данных:

а) Производительность. Время, затраченное на процесс получения семантического профиля пользователя из хранилища, может существенно увеличить время выполнения индивидуальной настройки веб-страниц.

б) Наличие механизмов разграничения доступа между несколькими веб-приложениями.

Созданное решение для хранения семантических профилей состоит из двух составных частей (рис. 4):

1. Веб-страница на удаленном сервере (серверная страница). Основное назначение этой страницы перенаправлять запросы к объекту *LocalStorage* после проверки прав доступа клиента на осуществление операции.
2. *JavaScript*-библиотека (клиент), представляющая из себя скрипт с программным интерфейсом для доступа хранилищу. Данная библиотека должна быть включена во все веб-страницы, которые подлежат семантической оптимизации. При первом обращении к программному интерфейсу на странице создается объект *iframe*, который ссылается на серверную веб-страницу. В дальнейшем запросы на получение или запись данных в объект *LocalStorage* направляются через объект *iframe*. В объект *iframe* загружается страница с главного домена. Объект *iframe* выступает в роли сервера. На странице, загруженной в объект *iframe*, содержится набор функций для записи и чтения данных из объекта *LocalStorage*.

Перед выполнением любого запроса происходит сопоставление URL-адреса веб-приложения со списком разрешенных доменов. Список разрешенных доменов хранится в JavaScript-библиотеке, размещенной на серверной странице. Список разрешенных доменов представляет собой строковую константу, содержащую список доменов любого уровня, разделенных символом пробела. В случае несоответствия домена клиента выполнение

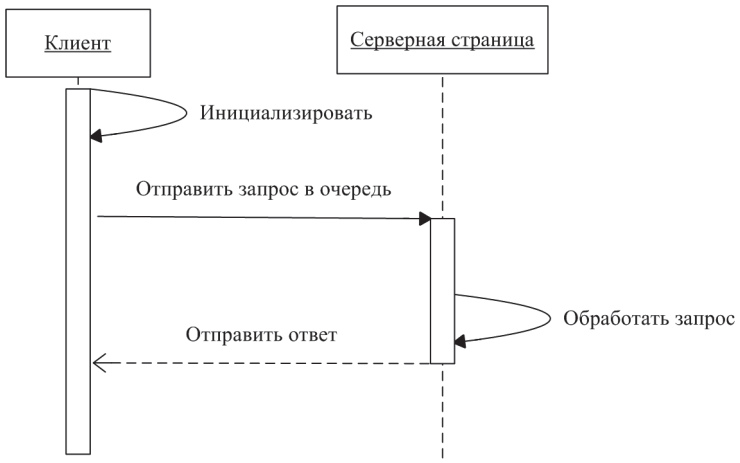

Рис. 4. *Процесс обращения к семантическому хранилищу.*

скрипта приостанавливается. Данный процесс выполняется на стороне клиента, но оригинальный *JavaScript*-код, загруженный вместе с серверной страницей, не может быть изменен сторонними объектами. Следовательно, исполнение данного процесса на стороне клиента не может негативно повлиять на безопасность приложения.

Разработанная архитектура предполагает, что группа веб-приложений, имеющих достаточно количество общих посетителей, может использовать общий семантический профиль. Данные о поведении пользователя и его предпочтениях могут стать доступными заранее определенному списку веб-приложений. Идентификатором приложения в таком списке является его доменное имя. Подобный подход позволяет получить более обширный набор данных о предпочтениях пользователя для более точного перестроения содержания страницы.

Компонент «Семантический анализатор» ответственен за процесс выборки всех элементов HTML-контента и дальнейшее выявление частичных или полных совпадений значений их семантических атрибутов с набором атрибутов семантического профиля пользователя. Процесс семантического анализа состоит из следующих процессов: получение набора элементов для анализа, установление рейтинга для каждого элемента и выполнение *callback*-функции для каждого из семантических элементов.

Релевантность того или иного элемента содержимого веб-страницы рассчитывается следующим образом:

- Поля семантического профиля располагаются по убыванию их оценки.
- Каждый тег семантического элемента сопоставляется полям профиля. В случае, если тег соответствует полю семантического профиля, происходит начисление рейтинга в зависимости от расположения поля в отсортированном семантическом профиле пользователя. При настройке семантического анализатора указывается шаг уменьшения рейтинга в зависимости от позиции поля семантического профиля. Оценки за каждое совпадение суммируются. В случае совпадения оценка начисляется по следующей формуле:

$$ratio_i = ratio_0 \times incr^i,$$

где $ratio_i$ – рейтинг, начисляемый за совпадение поля; i – позиция поля в отсортированном профиле пользователя; $incr$ – коэффициент убывания значимости.

По завершении работы семантического анализатора происходит настройка блоков контента в соответствии с предпочтениями пользователя следующими образами:

а) Изменение уровня прозрачности блоков контента, которые по результатам семантического анализа могут быть нерелевантными профилю пользователя.
б) Изменение порядка следования HTML-блоков, находящихся на одном уровне иерархии в DOM. Применение данного подхода более целесообразно для пользователей мобильной версии веб-сайта. Мобильные устройства имеют небольшой размер экрана, содержимое страницы мобильной версии интернет-приложения располагается в одной колонке, и пользователь уделяет большее внимание контенту, находящемуся в начале страницы [6].
в) Изменение оформления. При данном типе перестроения страницы возможно изменения размера, типа и цвета шрифта, добавление специальных графических символов или изменение цвета блока, содержащего семантические атрибуты.
г) Полное сокрытие нерелевантных блоков. Данный метод применим к узкому кругу веб-приложений. Наиболее целесообразно применять данный способ перестроения страницы, если пользователь явно отмечает свои предпочтения, а возможности самообучения отключены.

Способы изменения внешнего вида страницы, не меняющие порядок следования частей контента, удобны тем, что они не требует блокировки интерфейса в процессе анализа. Пользователь может

продолжать работать с содержимым страницы и по прошествии нескольких секунд наблюдать результаты анализа.

Созданное решение предлагает разработчику определять собственные *callback*-функции, выполняющие настройку контента путем изменения CSS-атрибутов HTML-элементов или иным способом. Входными параметрами для этой функции являются отсортированный по убыванию значения атрибута *rating* массив DOM-элементов, находящихся на одном уровне иерархии.

Пользователь веб-сайта с регулярно обновляющимся контентом посещает ресурс с некоторой периодичностью. Пользователь открывает для себя новые области знаний или сферы общественной жизни, что может означать изменение или коррекцию его предпочтений. При обычном сценарии использования веб-приложения пользователь не будет каждый раз редактировать свой профиль вручную в соответствии изменившимися настройками. Оптимальным поведением семантического анализатора является наблюдение за поведением пользователя и внесение изменений в его профиль, основанных на анализе поведения. Семантический анализатор отслеживает переходы пользователя по ссылкам, которые находятся внутри HTML-элементов с семантическими атрибутами.

Разработанная программная система может быть интегрирована с существующими веб-приложениями согласно следующей последовательности действий:

1. Определение сущностей, подлежащих семантическому анализу, и их семантических атрибутов. Примерами таких сущностей является статья, новость. Как правило, для семантического анализа выбираются сущности, которые отображаются на веб-странице в виде списка.
2. Создание логики для вывода семантических атрибутов. Данный этап может в себя включать изменение схемы базы данных. Изменение схемы базы данных не потребуется, если анализируемые сущности имеют иерархические отношения с другими сущностями, которые способны поставлять семантические атрибуты.
3. Подключение *JavaScript*-библиотек. Указание списка доверенных доменов. Задание начальной конфигурации анализатора и хранилища семантических профилей.

Процесс интеграции был подтвержден на примере фотоблога и агрегатора новостей.

Выводы

Разработанная архитектура семантического интернет-приложения может быть применена к широкому кругу существующих веб-приложений. Решение позволит разработчикам одновременно улучшить ранжирование интернет-приложения поисковыми системами и повысить удобство использования для посетителей. Произведенные работы по интеграции на примере двух веб-приложений подтвердили, что процесс интеграции не является трудоемким и не требует глубокой модификации уже существующих приложений.

Созданное решение отличается от существующих использованием интернет-браузера в качестве среды исполнения семантического анализатора и распределенным по конечным пользовательским устройствам хранилищем данных. Помимо этого, решение предлагает разделяемые наборы пользовательских профилей между несколькими веб-приложениями.

Созданный набор *JavaScript*-библиотек был опубликован на хостинге проектов с открытым исходным кодом *github.com* в репозитории *semanticOpt* и доступен широкому кругу разработчиков.

СПИСОК ЛИТЕРАТУРЫ

1. Enge E., Spencer S., Stricchiola J., Fishkin R. The art of SEO. Second edition. – Sebastopol, CA: O'Reily, 2012. – 714 p.
2. Agichtein E., Brill E., Dumais S. Improving Web Search Ranking by Incorporating User Behavior Information // SIGIR '06. The 29th Annual International SIGIR Conference Seattle. – WA, USA, 2006. – P. 19–26.
3. Porter J. Designing for the social web. – Berkeley, CA: New Riders, 2008. – 201 p.
4. Semantic HTML // MSDN. 2012. URL: http://msdn.microsoft.com/en-us/library/windows/desktop/gg671917.aspx (дата обращения: 10.03.2011).
5. Web Storage W3C Candidate Recommendation 08 December 2011 – W3C. Дата обновления: 08.12.2011. URL: http://www.w3.org/TR/webstorage/ (дата обращения: 11.05.2011).
6. Lubbers P., Albers B., Salim F. Pro HTML5 Programming: Powerful APIs for Richer Internet Application Development. – Berkeley, CA: Apress, 2010. – 304 p.

Поступила 13.07.2012 г.