УДК 004.774.2 + 004.774.6

ИСПОЛЬЗОВАНИЕ МИКРОРАЗМЕТОК ДЛЯ ДОБАВЛЕНИЯ В КОНТЕНТ ВЕБ-СТРАНИЦЫ ДАННЫХ ВНЕШНИХ РЕСУРСОВ

Е. Л. Китаев[1], Р. Ю. Скорнякова [2]

Институт прикладной математики им. М.В. Келдыша Российской академии наук, г. Москва;

[1]kitaev@keldysh.ru, [2]rimmaskorn@gmail.com

Аннотация

В семантических разметках Всемирной паутины накоплено большое число данных, и их количество продолжает расти. Однако потенциал этих данных реализуется, на наш взгляд, не в полной мере. Данные, заключенные в семантических разметках, или микроразметках, широко используются поисковыми системами, отчасти социальными сетями, использование же этих данных разработчиками приложений, как правило, основано на приведении данных к стандарту RDF и выполнении SPARQL-запросов, что требует хорошего знания этого языка и умения программировать. В настоящей работе предложено использовать имеющиеся в Сети семантические разметки для автоматического включения их содержимого в контент других веб-страниц и описан инструмент для реализации такого включения, не требующий от разработчика веб-страницы владения какими-либо языками программирования помимо широко известных HTML и CSS. Инструмент не требует установки, работу выполняют подключаемые стартовые скрипты. В настоящий момент инструмент поддерживает семантические данные, заключенные в популярных типах разметок «микроданные» и JSON-LD, в тегах <meta> HTML-документов и свойствах документов Word и PDF.

***Ключевые слова**: семантическая паутина, семантические технологии, семантическая разметка, микроразметка, микроданные, JSON-LD, веб-разработка, веб-технологии*

ВВЕДЕНИЕ

Бурно начавшись, развитие Семантической паутины [1] к началу 2010-х годов затормозилось. Реализация идеи превращения Всемирной паутины в Се-

мантическую столкнулась с определенными трудностями, одной из которых стала трудность освоения веб-разработчиками языка RDFa[1] (Resourse Description Framework in attributes) – основного языка семантической разметки веб-страниц. Из-за сложности RDFa (оборотной стороны его универсальности) использование этого языка оказалось ограниченным, а имеющиеся разметки содержали большое число ошибок. Новый импульс семантическому наполнению Всемирной паутины придало появление на рубеже 2010-х годов альтернативных синтаксисов семантической разметки «микроданные»[2], RDFa Lite[3] и JSON-LD[4], обладающих лучшим соотношением гибкости и сложности. По данным аналитической компании W3Techs[5], собирающей статистику об использовании различных веб-технологий, частота использования разметок «микроданные»[6] (≈15%) и JSON-LD[7] (≈27%) на 10 миллионах наиболее популярных сайтах в настоящий момент уже выше, чем частота использования Generic RDFa[8] (≈13%). При этом, если рост использования «микроданных» приостановился, то использование JSON-LD продолжает расти. На данный момент этот тип микроразметки представляется наиболее перспективным для дальнейшего семантического наполнения Всемирной паутины и использования в различных приложениях.

Популярность разметок «микроданные» и JSON-LD помимо более простого синтаксиса объясняется также тем, что их вместе со словарем Schema.org[9] активно используют поисковые системы. Содержащиеся в разметках данные служат для представления информации о веб-станицах в виде так называемых «расширенных сниппетов» (rich snippets)[10]. В отличие от обычных сниппетов, представляющих собой выдержку из неструктурированного текста, расширенные сниппеты содержат структурированную информацию, лучше отражающую содержание веб-страницы. И хотя напрямую семантические разметки при ран-

[1] http://rdfa.info/
[2] https://html.spec.whatwg.org/multipage/microdata.html
[3] https://www.w3.org/TR/rdfa-lite/
[4] https://json-ld.org/
[5] https://w3techs.com/
[6] https://w3techs.com/technologies/details/da-microdata/all/all
[7] https://w3techs.com/technologies/details/da-jsonld/all/all
[8] https://w3techs.com/technologies/details/da-genericrdfa/all/all
[9] https://schema.org/
[10] https://developers.google.com/search/docs/guides/mark-up-content/

жировании сайтов в настоящее время поисковыми системами не используются, представление в виде расширенного сниппета увеличивает «кликабельность» (click-through rate) и тем самым косвенно сказывается на ранжировании.

Другой вариант использования данных из семантических разметок – создание массивов структурированных веб-данных для проприетарного или свободного использования. Например, компания Google использует данные семантических разметок наряду с другими источниками для наполнения своей базы знаний Google Knowledge Graph[11], информация из которой частично доступна через Google Knowledge Graph Search API[12]. Интересным представляется проект Web Data Commons[13] [2], извлекающий и сохраняющий в формате RDF N-Quads[14] структурированные данные из самого большого из общедоступных массивов веб-данных Common Crawl[15]. Упакованные с помощью GZIP файлы свободно доступны для скачивания и могут использоваться для анализа веб-данных с помощью SPARQL[16] -запросов. Примеры использования данных Web Data Commons можно найти в работах [3, 4].

Для работы с данными, заключенными в семантических разметках, разработано различное программное обеспечение. Существуют инструменты для извлечения и валидации структурированных данных, заключенных в семантических разметках веб-страниц, для преобразования одних форматов структурированных данных в другие, инструменты, реализующие запросы к структурированным данным.

Для валидации структурированных можно использовать инструмент проверки структурированных данных[17], разработанный компанией Google или валидатор микроразметки[18] компании Яндекс. Для извлечения структурированных данных при просмотре веб-страниц разработаны различные плагины к браузе-

[11] https://googleblog.blogspot.com/2012/05/introducing-knowledge-graph-things-not.html
[12] https://developers.google.com/knowledge-graph
[13]http://webdatacommons.org/
[14] https://www.w3.org/TR/n-quads/
[15] https://commoncrawl.org/
[16] https://www.w3.org/TR/rdf-sparql-query/
[17] https://search.google.com/structured-data/testing-tool/u/0/
[18] https://webmaster.yandex.ru/tools/microtest/

рам, например, плагины к браузеру Google Chrome Microdata.reveal[19], Semantic inspector[20], JSON-LD Tester[21], Structured Data Testing Tool[22]. Они позволяют нажатием на иконку выделить и представить в отдельном окне структурированные данные текущей страницы.

Среди инструментов, предназначенных для программистов и позволяющих извлекать и преобразовывать структурированные данные из одного формата в другой, одним из наиболее популярных является свободно распространяемый фреймворк Apache Any23[23] (Anything To Triples), включающий библиотеку на языке Java, инструмент командной строки и веб-службу. Поддерживаются формат RDF[24] в различных сериализациях, RDFa, микроформаты[25], JSON-LD, микроданные, CSV, YAML[26]. Для преобразования «микроданных» в RDF используют также библиотеку на языке Python pyMicrodata[27]. Для извлечения структурированных данных из веб-страниц может служить API Валидатора микроразметки[28] компании Яндекс.

Для работы со структурированными данными формата RDF широко используется фреймворк с открытым кодом Apache Jena[29], написанный на языке Java. Он включает набор различных API и инструментов командной строки. Пример использования Apache Jena для обработки данных о товарах одного из интернет-магазинов приведен в блоге [5] сотрудника компании Commonwealth Computer Research, Inc, специалиста в области Semantic Web Боба ДюШарма. Данные о товарах, заключенные в разметках формата JSON-LD, преобразуются в

[19] https://chrome.google.com/webstore/detail/microdatareveal/olapakiakkblfdaajcifgldandnikpdh?hl=ru
[20] https://chrome.google.com/webstore/detail/semantic-inspector/jobakbebljifplmcapcooffdbdmfdbjh
[21] https://chrome.google.com/webstore/detail/json-ld-tester/aohmciehgjboidolkmoaofcbnejmokan?hl=de
[22] https://chrome.google.com/webstore/detail/structured-data-testing-t/kfdjeigpgagildmolfanniafmplnplpl
[23] https://any23.apache.org/
[24] https://www.w3.org/RDF/
[25] http://microformats.org/
[26] https://learn.getgrav.org/16/advanced/yaml
[27]https://github.com/RDFLib/pymicrodata
[28] https://yandex.ru/dev/validator/
[29] https://jena.apache.org/

формат RDF, а затем с помощью процессора SPARQL-запросов ARQ, входящего в состав Apache Jena, делаются выборки товаров с определенными характеристиками.

Более подробное изложение стандартов, концепций, приложений и инструментов для разработчиков, относящихся к области Semantic Web, имеется в книге [6]. В этой книге, как и во множестве других работ (см., например, [7, 8]), основной подход к разработке приложений, использующих структурированные данные, связан с использованием стандарта RDF и языка запросов SPARQL. Такой подход безусловно является мощным и универсальным, позволяющим создавать разнообразные приложения, однако для его реализации разработчик должен был специалистом в этой области, что ограничивает, на наш взгляд, возможности использования размещенных в Сети семантических данных. Эта область нуждается в инструментах, которые могли бы использовать не только специалисты в области Semantic Web, но и другие разработчики.

В настоящей работе мы предлагаем одно из возможных использований семантических данных при разработке веб-страниц и инструмент для его реализации, не требующий каких-либо дополнительных знаний и умений, помимо владения минимумом, необходимым веб-разработчику: языком разметки HTML и языком стилей CSS, а также знания основ синтаксиса микроразметок. Инструмент позволяет «на лету», в момент загрузки веб-страницы, включать в контент веб-страницы данные, заключенные в семантических разметках внешних ресурсов, и при этом не требует программирования.

1. МИКРОРАЗМЕТКИ И ДИНАМИЧЕСКАЯ АГРЕГАЦИЯ ВЕБ-ДАННЫХ БЕЗ ПРОГРАММИРОВАНИЯ

Как правило, использование семантических данных, размещенных в Сети, предполагает их предварительное извлечение и промежуточное хранение для дальнейшей обработки. Однако, если не стоит задача отбора веб-ресурсов по определенным условиям, гиперссылки на ресурсы известны, и данные необходимы только для показа пользователю, организация промежуточного хранения не является необходимой. Примером может служить составление для размещения в Сети разного рода списков: списков организаций с контактными данными, библиографических списков, подборок кулинарных рецептов, таблиц с ценами

на один и тот же товар в разных интернет-магазинах и т. п. В этих случаях данные можно динамически извлекать непосредственно из веб-ресурсов, где они размещены, и это особенно актуально, если данные не постоянны, как, например, контактные данные, цены, даты последней редакции в ссылках на живые публикации [9, 10] и т. п.

В общем случае задача извлечения информации из Сети весьма сложна, поскольку данные слабо структурированы и предназначены в первую очередь для прочтения человеком. В этом случае необходимо кодирование алгоритма извлечения данных, и для разных веб-ресурсов это алгоритм может быть разным. И хотя извлечение данных из Сети (веб-скрейпинг) является весьма популярной задачей и существует множество инструментов, помогающих в ее решении[30], создание на их основе инструмента, который позволил бы веб-разработчику без программирования динамически включать в контент данные из разных веб-ресурсов, не представляется возможным.

При наличии семантической разметки задача существенно упрощается, поскольку алгоритм извлечения данных зависит только от типа разметки. Ее успешно решают, например, упоминавшиеся выше фреймворк Apache Any23[31] и API Валидатора микроразметки[32] компании Яндекс. Однако мы не стали их использовать в качестве основы для нашего инструмента из-за ряда недостатков при одновременном извлечении данных из нескольких источников: для получения данных надо делать отдельный запрос для каждого из источников, из-за чего увеличивается общее время получения данных; в запросе нельзя указать тип данных – результат включает все, имеющиеся в разметке – и т. п.

Созданный нами инструмент StructScraper [11] позволяет динамически извлекать и включать в контент веб-страницы семантические данные, заключенные в разметках «микроданные» и JSON-LD, в тегах <meta> HTML-документов и свойствах документов Word и PDF.

Стандарт разметки «микроданные» был предложен в 2008 году сотрудником Google и на тот момент участником консорциума W3C Яном Хиксоном как

[30] http://scraping.pro/software-for-web-scraping/, https://www.garethjames.net/a-guide-to-Web-scrapping-tools/
[31] https://any23.apache.org/
[32] https://yandex.ru/dev/validator/

часть стандарта HTML5 с целью добавления семантики к имеющимся html-элементам. Для этого в HTML были введены специальные глобальные атрибуты itemscope, itemtype, itemprop, itemid, itemref. Атрибут itemscope помечает html-элемент как узел микроданных, атрибут itemtype задает тип данных (как правило, тип выбирается из какого-нибудь словаря), атрибут itemprop задает имя свойства, атрибут itemid предназначен для глобальных идентификаторов, атрибут itemref предназначен для ссылки на свойство, не содержащееся внутри узла микроданных. В листинге 1 приведен пример разметки микроданными контактных данных организации с использованием словаря Schema.org.

```
<div itemscope itemtype="http://schema.org/Organization">
    <span itemprop="name">ИПМ им. М.В.Келдыша РАН</span>
    <div>
        Контакты
        <div itemprop="address"
             itemscope
             itemtype="http://schema.org/PostalAddress">
            Адрес:
            <span itemprop="postalCode">125047</span>,
            <span itemprop="addressLocality">Москва</span>,
            <span itemprop="streetAddress">
                Миусская пл.,д.4
            </span>
        </div>
        Телефон:
        <span itemprop="telephone">+7 499 978-13-14</span>,
        Факс:
        <span itemprop="faxNumber">+7 499 972-07-37</span>,
    </div>
</div>
```

Листинг 1. Микроразметка с использованием микроданных

Стандарт разметки JSON-LD (JSON for Linking Data) был предложен в 2010 году как альтернатива формату RDFa программистами, которые в своем проекте использовали для внутреннего хранения данных широко распространенный в веб-приложениях формат JSON. При этом данные, которые они извлекали и об-

рабатывали, хранились в Сети в формате RDFa. Идея состояла в том, чтобы данные в веб-документах, которые предназначаются для программной обработки, тоже хранились в формате JSON. В этом случае отпадает необходимость преобразования одного формата в другой. В 2014 году JSON-LD стал официальным стандартом, поддерживаемым консорциумом W3C. JSON-LD совместим с RDF, он является еще одним способом сериализации RDF.

В html-документе данные формата JSON-LD помещаются в тег <script> с атрибутом type="application/ld+json". Ключевые слова, входящие в синтаксис JSON-LD, начинаются с символа "@". В листинге 2 приведен пример разметки JSON-LD для научной публикации.

```
<script type="application/ld+json">
{
  "@context": "http://schema.org",
  "@type": "ScholarlyArticle",
  "name": "Живая публикация",
  "author": {
       "@type": "Person",
       "name": "Горбунов-Посадов, Михаил Михайлович"
            },
  "datePublished": "2011",
  "dateModified": "2019-05-01",
  "url": "https://keldysh.ru/gorbunov/live.htm"
}
</script>
```

Листинг 2. Микроразметка с использованием JSON-LD и словаря Schema.org

Для использования предлагаемого нами инструмента автору веб-страницы достаточно соответствующим образом подготовить разметку HTML-страницы (вставив нужные атрибуты) и подключить стартовые скрипты (вставив в страницу фрагмент заранее подготовленного кода) – вся остальная работа по включению данных из семантических разметок внешних ресурсов выполняется автоматиче-

ски в процессе загрузки страницы. Инструкция по оформлению веб-страницы размещена на посвященном инструменту сайте[33].

StructScraper могут использовать как профессиональные разработчики, так и непрофессиональные авторы, создающие собственные страницы, поскольку для его использования необходимо знать только основы HTML и CSS. Он может быть полезен блогерам, авторам страниц с кулинарными рецептами, научным работникам для создания персональных страниц и списков публикаций, его можно использовать для сравнения цен на товары, рейтингов сайтов и т.п.

2. ПРИМЕРЫ ИПОЛЬЗОВАНИЯ

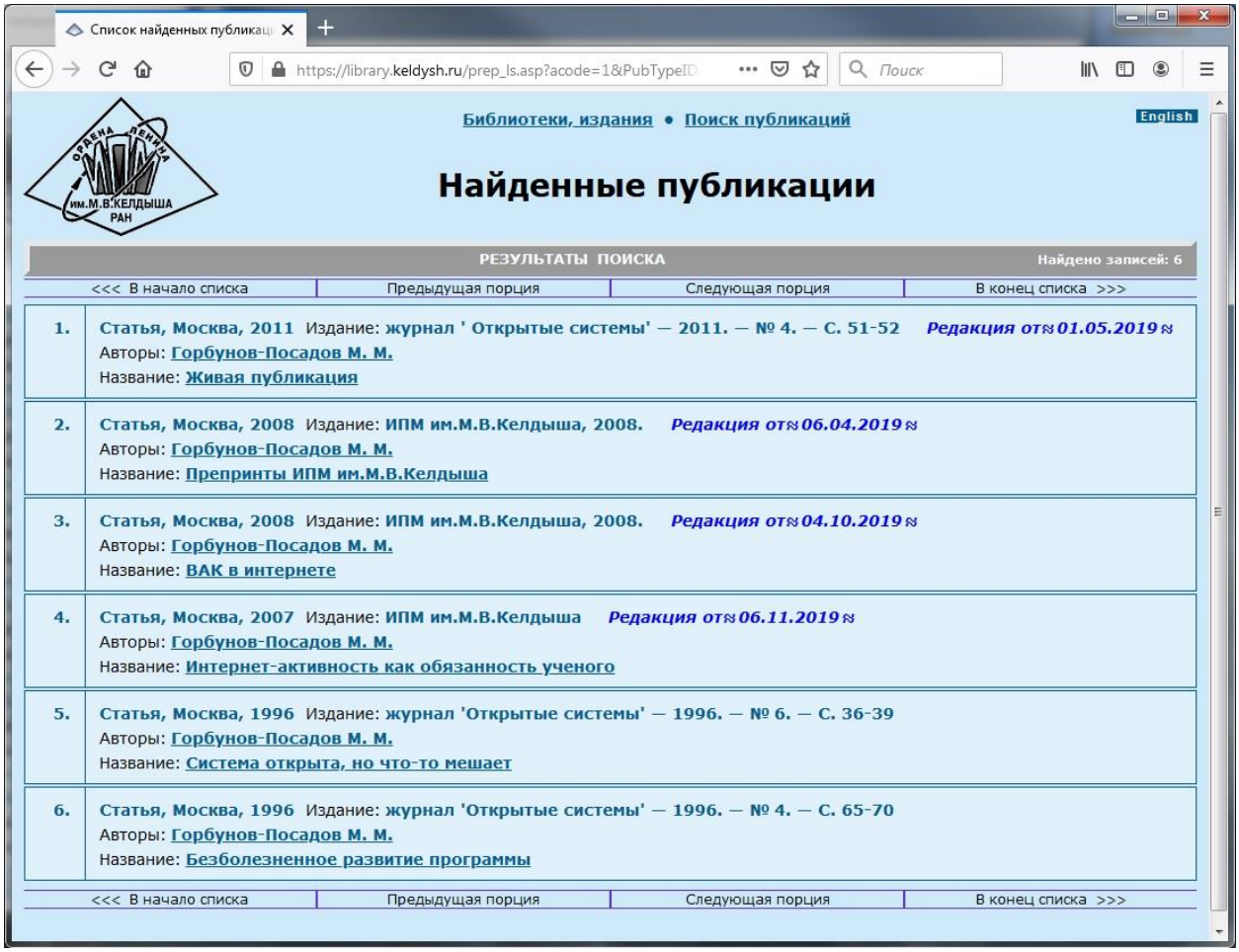

Рис. 1. Пример использования инструмента: дата последней редакции в ссылке на «живую» публикацию

С необходимостью автоматического включения в контент веб-страницы данных внешних веб-ресурсов мы впервые столкнулись в работе над инструмен-

[33] http://struct-scraper.keldysh.ru/doc-page.html

тами поддержки «живых» публикаций [9]. «Живая» публикация – это размещенная в свободном доступе в интернете научная работа, которая постоянно совершенствуется и развивается ее автором. Дата последней редакции является важным атрибутом такой публикации, показывающим, насколько актуальна работа. Эта дата должна храниться в самой публикации и автоматически обновляться в онлайн-ссылках на нее. Ручное обновление смысла не имеет, поскольку может происходить с запаздыванием. Ссылки на работу могут размещаться на веб-страницах разных сайтов, и не все авторы этих страниц могут быть профессиональными фронтенд-разработчиками, поэтому необходим был простой в использовании инструмент [10]. Инструмент был реализован и внедрен в ИПМ им. М.В. Келдыша в 2017 году (рис. 1).

Эта задача не единственная, где автоматическое включение в контент веб-страницы данных внешних ресурсов может быть полезным. Поэтому возникла идея разработать более общий инструмент, пригодный для разных случаев использования.

Примером такого использования может служить мониторинг цен. Сейчас многие интернет-магазины добавляют микроразметку к описаниям товаров, представленных в каталогах, включающую цену на товар. Веб-страница, доступная по адресу https://struct-scraper.keldysh.ru/test_pages/product.html, содержит таблицу с ценами на одну и ту же модель смартфона (рис. 2), подгружаемыми «на лету», что гарантирует их актуальность на момент загрузки. Изначальная разметка для каждой строки таблицы имеет вид, как в листинге 3. Она содержит только гиперссылку на модель смартфона. В html-код добавлен также вызов готового плагина jQuery, которому в качестве параметров передаются адрес REST API и тип Product из словаря Schema.org. При такой разметке подгружаются все свойства Product и записываются в теги . Какие из них должны быть видны пользователю, определяется в CSS.

Смартфон Samsung Galaxy Note 10

Интернет-магазин	Модель	Цена
1Click	Samsung Galaxy Note 10	47690
Болтун	Samsung Galaxy Note 10 8/256GB N970 Черный РСТ	56489
Комус	Смартфон Samsung Galaxy Note 10 256 ГБ черный (SM-N970FZKDSER)	76990.00
М-Видео	Смартфон Samsung Galaxy Note10 Black (SM-N970F)	76990
Мегафон	Смартфон Samsung Galaxy Note10 Чёрный	76 990
МТС	Смартфон Samsung N970 Galaxy Note 10 8/256Gb Черный	76990.00
Плеер.ру	Сотовый телефон Samsung SM-N970F Galaxy Note 10 8Gb/256Gb Black	61599
Самсунг	Samsung Galaxy Note 10, 256 Гб, Чёрный	76990
Технопарк	Смартфон Samsung Galaxy Note10 черный	76990.00
ТехноСити	Смартфон Samsung SM-N970 Galaxy Note 10, 256 Gb, чёрный (SM-N970FZKDSER)	76990.00
Allo.market	Samsung Galaxy Note 10 8/256GB (EXINOS)	55890
Ant-Shop	Смартфон Samsung Galaxy Note 10 8/256GB Черный	69990
Appleavenue	Samsung Galaxy Note 10 8/256Gb (SM-N970F) (Aura White)	52550
Elecity	Смартфон Samsung Galaxy Note 10 8/256GB черный	63130
FLASH	Смартфон Samsung Galaxy Note 10 (2019) SM-N970 8/256GB черный, SM-N970FZKDSER	54 380.–
Galaxystore	Galaxy Note10 256 ГБ, черный	76990
GPod	iMac	45912
HiFi Zona	Смартфон Samsung Galaxy Note 10 Black (SM-N970F) черный	58970
Likemobmarket	Samsung Galaxy Note 10 256GB Aura Glow	47679
lite-mobile	Смартфон Samsung Galaxy Note 10 SM-N970F/DS 256Gb (Цвет: Aura Black)	56750

Рис. 2. Пример использования инструмента: мониторинг цен

```
<tr class="import-struct">
  <td>Мегафон</td>
  <td>
    <a class="struct-url"
       href="https://moscow.shop.megafon.ru/mobile/117195.html">
    </a>
   </td>
</tr>
```

Листинг 3. HTML-разметка для строки таблицы с ценами

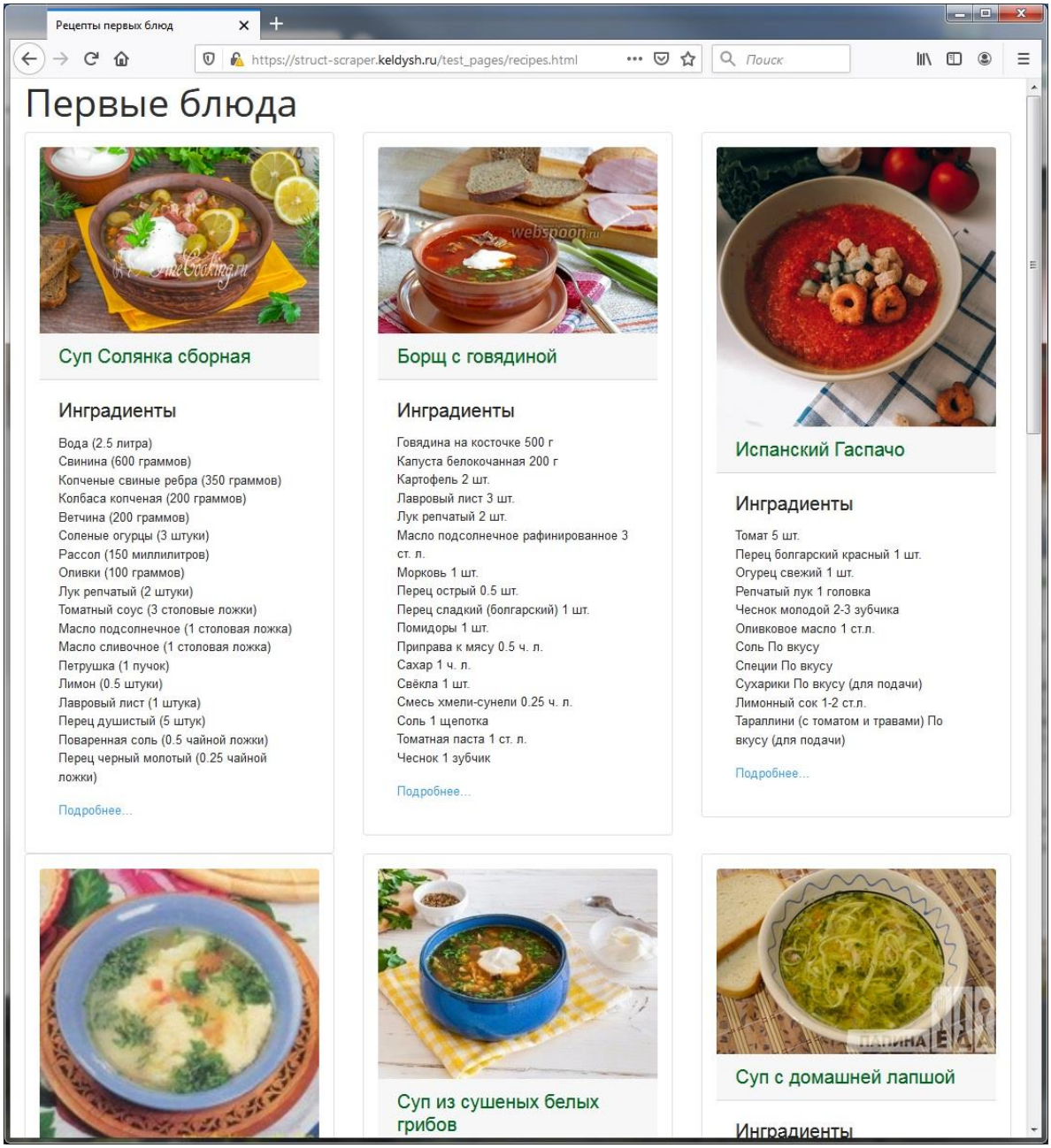

Рис. 3. Пример использования инструмента: страница с кулинарными рецептами

```
<div class="card import-struct">
    <a href="https://papinaeda.ru/244">
        <img class="card-img-top"
             itemprop="resultPhoto"
             alt="Фото"
             data-unique="true">
    </a>
    <a href="https://papinaeda.ru/244">
        <img class="card-img-top"
             itemprop="image"
             alt="Фото"
             data-unique="true">
    </a>
    <h5 class="card-header">
        <a class="struct-url"
           href="https://papinaeda.ru/244" itemprop="name">
           Суп с домашней лапшой
        </a>
    </h5>
    <div class="card-body">
        <h5 class="card-title">Ингрaдиенты</h5>
        <ul class="list-unstyled">
            <li itemprop="recipeIngredient"></li>
            <li itemprop="ingredients"></li>
        </ul>
        <a href="https://papinaeda.ru/244"
           class="card-link">
           Подробнее...
        </a>
    </div>
</div>
```

Листинг 4. HTML-разметка для кулинарного рецепта

Еще один пример – страница с подборкой кулинарных рецептов (рис. 3). Рецепты могут располагаться на разных сайтах. При использовании предлагаемого инструмента составителю подборки нет необходимости копировать их на свою страницу. Достаточно только вставить гиперссылки на рецепты в шаблон html-разметки, и рецепты автоматически загрузятся на страницу. Пример доступен по адресу https://struct-scraper.keldysh.ru/test_pages/recipes.html. Поскольку в этом случае на страницу добавляются из внешних сайтов не только тексты, но и изображения, здесь используется тип разметки, отличный от предыдущего примера (листинг 4).

На сайте инструмента [11] и его странице[34] на веб-сервисе GitHub можно найти дополнительные примеры использования.

3. РЕАЛИЗАЦИЯ ИНСТРУМЕНТА

Реализация предлагаемого нами инструмента StructScraper включает серверную и клиентскую части. Серверная часть представляет собой REST API для извлечения данных – ее можно использовать как вместе с клиентской частью, так и самостоятельно. Клиентская включает jQuery плагины, вызов которых при загрузке веб страницы выполняет работу по добавлению данных в контент.

REST API StructScraper реализован на языке C#, имеющем встроенную поддержку асинхронного программирования, с использованием технологии Microsoft ASP.NET Web API.

Обращения к REST API производятся методом POST с передачей параметров в формате JSON. Параметры включают адреса веб-ресурсов, из которых необходимо извлечь данные, и сведения о том, какие именно данные должны быть извлечены. Для метаданных, извлекаемых из тегов и свойств документов, передаются названия, для микроданных и разметки JSON-LD передается список типов из словаря Schema.org.

Обработка нескольких URL на сервере происходит асинхронно, время ответа равно максимальному времени ответа от одного ресурса. Поэтому время ответа на клиентский запрос не больше, чем если бы запросы с клиентской стороны для каждого URL производились отдельно, а за счет сокращения времени на установку отдельных соединений к REST API, а также из-за отсутствия ограничений на число одновременно выполняемых асинхронных запросов, оно становится меньше.

REST API StructScraper допускает CORS, запросы к нему могут производиться из клиентских частей веб-приложений любых доменов. CORS[35] (Cross Origin Resource Sharing) – это технология, реализованная в современных браузерах, частично снимающая ограничения правила одного источника, введенного из соображений безопасности с целью дать возможность коду из веб-страницы одного сайта свободно взаимодействовать с ресурсами этого же сайта и максимально

[34] https://github.com/RimmaSkorn/struct-scraper
[35] https://fetch.spec.whatwg.org/#http-cors-protocol

ограничить такое взаимодействие с ресурсами других сайтов. Без такого ограничения скрипт из страницы, загруженной с какого-нибудь сайта, мог бы обратиться, например, к почтовому серверу пользователя через сессию, открытую в другом окне браузера, получить его почту или отправить от его имени письмо, что непременно бы использовали злоумышленники. В соответствии с правилом одного источника браузеры запрещают производить ajax-запросы к стороннему серверу. До появления технологии CORS такие запросы были запрещены полностью. Технология CORS, реализованная как надстройка над HTTP протоколом, позволяет их осуществлять, если сторонний сервер дает на это явное разрешение. При этом сервер также контролирует детали кросс-доменных запросов: разрешенные методы, возможности передачи авторизующих заголовков и т.п.

Работу по загрузке клиентом данных на веб-страницу осуществляет JavaScript код, оформленный в виде плагинов jQuery. Для того чтобы плагин загрузил данные на веб-страницу, в нее должна быть добавлена специальная разметка, по которой код плагина определяет, из каких веб-ресурсов должны быть загружены данные и какие именно данные необходимо загрузить. Имеется несколько типов разметок с использованием только атрибутов class и с использованием атрибутов class и микроданных.

StructScraper является инструментом с открытым кодом, доступном[36] на веб-сервисе GitHub.

ЗАКЛЮЧЕНИЕ

Сегодня во Всемирной паутине еще редко встречаются страницы, предъявляющие посетителю информацию из нескольких активно обновляемых сайтов, собранную в момент обращения к этой странице. В настоящей работе

— предложено использовать для включения в контент веб-станицы данные, заключенные в получивших широкое распространение семантических разметках;

— описан созданный авторами инструмент для реализации такого включения, не требующий установки и написания программного кода;

— приведены примеры использования созданного инструмента.

[36] https://github.com/RimmaSkorn/struct-scraper

В дальнейшем предполагается расширить возможности предлагаемого инструмента за счет добавления поддержки разметки RDFa Lite, реализации выбора пользователем словаря семантической разметки, одновременного включения в контент нескольких объектов из одного внешнего ресурса и др. Предполагается также добавить шаблоны разметок для других случаев использования, в частности, для автоматического формирования библиографических ссылок по гиперссылкам на размещенные в Сети научные публикации.

СПИСОК ЛИТЕРАТУРЫ

1. *Bizer C., Heath T., Berners-Lee T.* Linked Data – The Story so far // International Journal on Semantic Web and Information Systems. 2009. V. 5. No. 3. P. 1–22.

2. *Meusel R., Petrovski P., Bizer C.* The WebDataCommons Microdata, RDFa and Microformat Dataset Series // Proceedings of the 13th International Semantic Web Conference: «The Semantic Web – ISWC 2014», Part I, Riva del Garda, Italy, October 19–23, 2014. Lecture Notes in Computer Science, vol. 8796. Springer: 2014. P 277–292.

3. *Lehmberg O., Ritze D., Ristoski P., Meusel R., Paulheim H., Bizer C.* The Mannheim Search Join Engine // Journal of Web Semantics. 2015. V. 35. Part. 3. P. 159–166.

4. *Lohvynenko C., Nedbal D.* Usage of Semantic Web in Austrian Regional Tourism Organizations // Proceedings of the 15th International Conference on Semantic Systems: «SEMANTiCS 2019», Karlsruhe, Germany, September 9–12, 2019. Lecture Notes in Computer Science, vol. 11702. Springer: 2019. P. 3–18.

5. *DuCharme B.* Exploring JSON-LD. URL: http://www.bobdc.com/blog/json-ld/

6. *Yu Liyang.* A Developer's Guide to the Semantic Web. Second Edition. Heidelberg: Springer, 2014. 829 p. DOI:10.1007/978-3-662-43796-4.

7. *Апанович З.В.* Ресурсы и инструменты для преподавания методов и средств Semantic Web // Системная информатика. 2017. № 11. С. 1–20.

8. *Апанович З.В.* Преподавание методов Semantic Web разработчикам программного обеспечения // Труды XIX Всероссийской научной конференции «Научный сервис в сети Интернет», г. Новороссийск, 18–23 сентября 2017 г. М.: ИПМ им. М.В. Келдыша: 2017. С. 9–20. URL: http://keldysh.ru/abrau/2017/37.pdf. DOI:10.20948/abrau-2017-37

9. *Горбунов-Посадов М.М.* Живая публикация // Открытые системы. 2011. № 4. С. 51–52. URL: http://keldysh.ru/gorbunov/live.htm

10. *Горбунов-Посадов М.М., Скорнякова Р.Ю.* Обновляемая дата последней редакции в ссылке на живую публикацию // Препринты ИПМ им. М.В. Келдыша. 2017. № 82. 14 с. DOI:10.20948/prepr-2017-82, URL: http://library.keldysh.ru/ preprint.asp?id=2017-82

11. *StructScraper.* URL: https://struct-scraper.keldysh.ru/

LEVERAGING SEMANTIC MARKUPS FOR INCORPORATING EXTERNAL RESOURCES DATA TO THE CONTENT OF A WEB PAGE

E. L. Kitaev[1], R. Y. Skornyakova [2]

Keldysh Institute of Applied Mathematics (Russian Academy of Sciences), Moscow;

[1]kitaev@keldysh.ru, [2]rimmaskorn@gmail.com

Abstract

The semantic markups of the World Wide Web have accumulated a large amount of data and their number continues to grow. However, the potential of these data is, in our opinion, not fully utilized. The semantic markups contents are widely used by search systems, partly by social networks, but the usual approach to using that data by application developers is based on converting data to RDF standard and executing SPARQL queries, which requires good knowledge of this language and programming skills. In this paper, we propose to leverage the semantic markups available on the Web to automatically incorporate their contents to the content of other web pages. We also present a software tool for implementing such incorporation that does not require a web page developer to have knowledge of any programming languages other than HTML and CSS. The developed tool does not require installation, the work is performed by JavaScript plugins. Currently, the tool supports semantic data contained in the popular types of semantic markups “microdata” and JSON-LD, in the <meta> tags of HTML documents and the properties of Word and PDF documents.

Keywords*: semantic web, semantic technologies, semantic markup, microdata, JSON-LD, web development, web technologies*

REFERENCES

1. *Bizer C., Heath T., Berners-Lee T.* Linked Data – The Story so far // International Journal on Semantic Web and Information Systems. 2009. V. 5. No. 3. P. 1–22.

2. *Meusel R., Petrovski P., Bizer C.* The WebDataCommons Microdata, RDFa and Microformat Dataset Series // Proceedings of the 13th International Semantic Web Conference: «The Semantic Web – ISWC 2014», Part I, Riva del Garda, Italy, Oc-

tober 19–23, 2014. Lecture Notes in Computer Science, vol 8796. Springer: 2014, P 277–292.

3. *Lehmberg O., Ritze D., Ristoski P., Meusel R., Paulheim H., Bizer C.* The Mannheim Search Join Engine // J. of Web Semantics. 2015. V. 35. Part. 3. P. 159–166.

4. *Lohvynenko C., Nedbal D.* Usage of Semantic Web in Austrian Regional Tourism Organizations // Proceedings of the 15th International Conference on Semantic Systems: «SEMANTiCS 2019», Karlsruhe, Germany, September 9–12, 2019. Lecture Notes in Computer Science, vol 11702. Springer: 2019. P. 3–18.

5. *DuCharme B.* Exploring JSON-LD. URL: http://www.bobdc.com/blog/json-ld/

6. *Yu Liyang.* A Developer's Guide to the Semantic Web. Second Edition. Heidelberg: Springer, 2014. 829 p. DOI:10.1007/978-3-662-43796-4.

7. *Apanovich Z.V.* Resursy i instrumenty dlia prepodavaniia metodov i sredstv Semantic Web // Sistemnaia informatika. 2017. No 11. S. 1–20.

8. *Apanovich Z.V.* Prepodavanie metodov Semantic Web razrabotchikam programmnogo obespecheniia // Trudy XIX Vserossiiskoi nauchnoi konferentsii «Nauchnyi servis v seti Internet», g. Novorossiisk, 18–23 sentiabria 2017 g. M.: IPM im. M.V. Keldysha: 2017. S. 9–20. URL: http://keldysh.ru/abrau/2017/37.pdf. DOI:10.20948/abrau-2017-37

9. *Gorbunov Posadov M.M.* Zhivaia publikatsiia // Otkrytye sistemy. 2011. No 4. S. 51–52. URL: http://keldysh.ru/gorbunov/live.htm

10. *Gorbunov Posadov M.M., Skorniakova R.Iu.* Obnovliaemaia data poslednei redaktsii v ssylke na zhivuiu publikatsiiu // Preprinty IPM im. M.V. Keldysha. 2017. № 82. 14 s. DOI:10.20948/prepr-2017-82 URL: http://library.keldysh.ru/preprint.asp?id=2017-82.

11. *StructScraper.* URL: https://struct-scraper.keldysh.ru/

СВЕДЕНИЯ ОБ АВТОРАХ

КИТАЕВ Евгений Львович – инженер-исследователь Института прикладной математики им. М.В. Келдыша РАН, специалист в области веб-технологий и информационных систем.

Evgeny L'vovich KITAEV – Research Engineer at the Keldysh Institute of Applied Mathematics RAS, specialist in web technologies and information systems.

email: kitaev@keldysh.ru

СКОРНЯКОВА Римма Юрьевна – научный сотрудник Института прикладной математики им. М.В. Келдыша РАН, специалист в области разработки информационных систем.

Rimma Yuryevna SKORNYAKOVA – Researcher at the Keldysh Institute of Applied Mathematics RAS, specialist in the development of information systems.

email: rimmaskorn@gmail.com

Материал поступил в редакцию 15 ноября 2019 года