Проблемы интеграции сетевых электронных коллекций

Н.В. Борисов, В.В. Захаркина, И.А. Мбого, П.П. Щербаков

Санкт-Петербургский государственный университет

nikborisov@gmail.com, zakharkina@gmail.com,
irina.mbogo@gmail.com, paul.tscherbakov@gmail.com

Аннотация

В настоящей статье предложен новый подход к решению задач интеграции сетевых электронных коллекций. Рассмотрены механизмы взаимодействия между электронными коллекциями, позволяющие напрямую встраивать в формируемую мультимедиа коллекцию (или научную статью с мультимедийным контентом) произвольные элементы из удаленной коллекции (научной статьи). Описаны подходы к программной реализации соответствующих интерфейсов.

Ключевые слова: электронная коллекция, онлайн коллекция, интеграция электронных коллекций, мультимедиа контент

Библиографическая ссылка: Борисов Н.В., Захаркина В.В., Мбого И.А., Щербаков П.П. Проблемы интеграции сетевых электронных коллекций // Информационное общество: образование, наука, культура и технологии будущего. Выпуск 3 (Труды XXII Международной объединенной научной конференции «Интернет и современное общество», IMS-2019, Санкт-Петербург, 19 – 22 июня 2019 г. Сборник научных трудов). — СПб: Университет ИТМО, 2019. С. 67 – 73. DOI: 10.17586/2587-8557-2019-3-67-73

Введение

Вопросы интеграции электронных коллекций волнуют научное и образовательное Интернет сообщество довольно давно. Разработано значительное количество инструментов, позволяющих передавать метаданные различным интеграторам и агрегаторам как общего назначения, так и специализированным. Ведущие IT компании, такие как Google, Facebook, Twitter разработали свои механизмы описания информации на основе метаданных. В целях сбора и обработки метаданных используются дополнительные протоколы, такие как OAI-PMH. Таким образом, реализуется «вертикальное» взаимодействие.

Взаимодействие с интеграторами и агрегаторами позволяет получить доступ к широкому кругу объектов коллекций, но не позволяет непосредственно встраивать различные элементы удаленных коллекций на свои ресурсы. При этом целый ряд веб-сервисов (социальные сети, геоинформационные системы, медиасервисы и т.д.) дают возможность интеграции своих функциональных элементов на сторонние сайты. Реализация подобной интеграции предоставляет соответствующий инструмент: WEB-API (*Application Programming Interface*, то есть программные возможности для сторонних разработчиков ресурсов World Wide Web). В случае сложного контента, внедряемого на сторонний сайт, WEB-API сервиса, как правило, предлагает программный код, реализующий отображение полнофункционального воспроизведения во встроенном блоке-«фрейме». С технологической точки зрения, это использование HTML-элемента ***iframe***. Однако, простое использование iframe, который могли бы использовать составители в своих коллекциях, не прибегая к программированию, не покрывает всего

спектра необходимых возможностей. например, вставку в один iframe не всю удаленную страницу целиком, а произвольное количество ее фрагментов. Вопросы встраивания различных объектов решаются с помощью технологий межпрограммного взаимодействия, например, программного интерфейса API или веб-сервисов. Зачастую даже развитого API бывает недостаточно для эффективного включения элементов удаленных ресурсов и нужна работа программистов. В статье рассмотрены механизмы взаимодействия, позволяющие напрямую встраивать произвольные элементы удаленной коллекции специалистам, связанным с публикацией цифрового контента и не владеющим навыками программирования.

1. Глобальные интеграторы

Отправной точкой для многих исследователей при информационном поиске являются поисковые системы общего назначения, такие как Google, Yandex, глобальный сервис Wikipedia, специализированные сервисы Google Scholar и Google Book. Ведущие социальные сети – Facebook и Twitter также имеют инструменты, позволяющие формировать более структурированную информацию об объекте. В целях интеграции ресурсов разработаны методы, связанные с облегчением индексации и поиска цифровых коллекций, основанные на метаописаниях. Анализ метаописаний может позволить браузерам, поисковым системами и другими сайтам дополнительно структурировать информацию.

1.1. Dublin Core

Первый шаг в этом направлении был реализован при разработке словаря для унификации метаданных и описания ресурсов Dublin Core, например:

```
<meta name="DC.Title" content="Заголовок страницы">
<meta name="DC.Creator" content="Имя сайта или создателя страницы">
<meta name="DC.Subject" content="Тема содержания ресурса">
<meta name="DC.Description" content="Описание страницы">
<meta name="DC.Publisher" content="Издатель">
```

Минимальный, простой словарь Дублинского ядра состоит из 15 элементов: метаданных Дублинского ядра: Title — название, Creator — создатель, Subject — тема, Description — описание, Publisher — издатель, Contributor — внёсший вклад, Date — дата, Type — тип, Format — формат документа, Identifier — идентификатор, Source — источник, Language — язык, Relation — отношения, Coverage — покрытие, Rights — авторские права [1].

1.2. Google Scholar

Специализированная поисковая система по научной литературе Google Scholar, включающая научные журналы, статьи, тезисы, рефераты, разработала мета-тэги описания библиографической информации. Google Scholar требует, чтобы веб-страница любой статьи имела мета-теги для названия этой статьи (citation_title), автора (авторов) (citation_author) и даты публикации (citation_publication_date). Веб-страницы трудов конференций и журналов требуют метаданных для их идентификации при цитировании в другой статье. Они включают теги для тома и номера выпуска (citation_volume и citation_issue), а также номера первой и последней страниц статьи (citation_firstpage и citation_lastpage).

```
<meta name="citation_journal_title" content="Journal Name">
<meta name="citation_authors" content="Last Name1, First Name1; Last Name2, First Name2">
<meta name="citation_title" content="Article Title">
```

```
<meta name="citation_date" content="01/01/2019">
```

К мета-тэгам Google Scholar относятся: citation_title, citation_author, citation_publication_date, citation_journal_title, citation_issn, citation_isbn, citation_volume, citation_issue, citation_firstpage, citation_lastpage, citation_dissertation_institution, citation_technical_report_institution, citation_technical_report_number, citation_pdf_url [2].

1.3. Facebook

С целью управления способом отображения при публикации ссылки в социальных сетях и передачи информации другим интернет-сервисам специалисты Facebook разработали протокол OpenGraph. Facebook обрабатывает эти мета-тэги и, вместо ссылки, указанной пользователем вставляет целый фрагмент с сайта на который указывает ссылка. В превью можно добавлять не только текстовые, но и мультимедиа элементы.

```
<meta property="og:title" content="The Rock" />
<meta property="og:type" content="video.movie" />
<meta property="og:url" content="//www.imdb.com/title/tt0117500/" />
<meta property="og:image" content="//ia.media-imdb.com/images/rock.jpg" />[3]
```

К мета-тэга OpenGraph относятся: og:title, og:type, og:image, og:url, og:audio, og:description, og:determiner, og:locale, og:locale:alternate, og:site_name, og:video.

1.4. Twitter

Социальная сеть Tvitter по аналогии с OpenGraph предлагает использование Twitter Card –инструмента, позволяющего управлять выводом заголовков, изображений, описаний в твитах.

```
<meta name="twitter:card" content="summary" />
<meta name="twitter:site" content="@nytimesbits" />
<meta name="twitter:creator" content="@nickbilton" />
<meta property="og:url" content="http://bits.blogs.nytimes.com/2011/12/08/a-twitter-for-
my-sister/" />
<meta property="og:title" content="A Twitter for My Sister" />[4]
```

К мета-тэгам Twitter относятся: twitter:card, twitter:title, twitter:description, twitter:image, twitter:domain, twitter:url, twitter:data1, twitter:label1, twitter:data2, twitter:label2.

2. Агрераторы контента

Наряду с глобальными интеграторами существует довольно много как универсальных научных агрегаторов, так и агрегаторов, специализирующихся на более узкой тематике. Для пополнения базы агрегаторами используются различные методы, начиная с индексирования страниц по аналогии с поисковыми системами и заканчивая материалами, обрабатываемыми вручную. Самым крупным интегратором в России является система Соционет [5], выполненная с использованием результатов и рекомендаций международных инициатив RePEc, Open Archives Initiative, CRIS-CERIF, Web Annotation и др.

3. Возможности горизонтальной интеграции

Возможность вставить в свою коллекцию только некоторую часть данных из другой коллекции, либо иного сервиса, непосредственно, минуя агрегаторов, требует реализации «горизонтальной» интеграции сайт – сайт.

Так, на сайте конференции на странице "Программа IMS-2019" приводится фрагмент скриншота веб-страницы Яндекс:Карты, объясняющий участникам путь к месту проведения мероприятия. Таким образом, организационный комитет конференции, для

решения навигационной задачи, использовал возможности, предоставляемые популярным российским картографическим сервисом. Однако сервис Яндекс:Карты в действительности дает возможность получить гораздо большую функциональность при использовании WEB-API.

Сервис Яндекс:Карты предоставляет инструмент, который позволяет выбрать требуемый фрагмент карты, задать нужные виджеты управления и отображаемые информационные слои, добавить собственные информационные элементы и получить фрагмент HTML-кода, который можно внедрить на WEB-страницу. Пример кода для вставки на сайт пользователя:

```
<iframe src="https://yandex.ru/map-widget/v1/-/CCCZRRK-" width="560" height="400" frameborder="1" allowfullscreen="true"></iframe>.
```

Близкую функциональность можно встретить на видеохостинге YouTube – найдя нужный видеоклип и выбрав опцию «Поделиться» пользователь, выбрав необходимые параметры видеоклипа и доступных элементов управления, получает фрагмент HTML-кода, подобного приведенному ниже:

```
<iframe width="560" height="315" frameborder="0"
src="https://www.youtube.com/embed/V9jeErVgEzQ"
allow="accelerometer; autoplay; encrypted-media; gyroscope; picture-in-picture"
allowfullscreen></iframe>
```

Межсайтовое взаимодействие можно реализовывать, развивая WEB-API и пользовательские визуальные инструменты работы с ним. Наиболее комплексным решением является создание веб-приложения, которое из набора полей страницы коллекции-донора формирует код, который далее может быть вставлен на страницу коллекции реципиента.

3.1. Кросс-доменные запросы к источнику данных

Первый вариант возможной реализации горизонтальной интеграции заключается в формировании, например, JavaScript кода, который бы с помощью кросс-доменных запросов скачивал необходимые поля и вставлял их на страницу. Все браузеры ограничивают использование межсайтового скриптинга в целях безопасности. Однако, новая спецификация асинхронных запросов XMLHttpRequest позволяет использовать кросс-доменные запросы, определяя каким доменам разрешено соединение, какие методы протокола HTTP доступны для взаимодействия и другие параметры. Главная проблема использования кросс-доменных запросов – это отсутствие их поддержки во всех браузерах и необходимость специальной настройки веб-сервера донора, которую не все хостинг-провайдеры разрешают.

3.2. API сервера коллекции

Другой подход основан на использовании API сервера-донора. Обычно сервер на любой вид запроса по протоколу HTTP возвращает машиночитаемый результат в формате XML или JSON. Следует отметить, что для использования в обмене только некоторых фрагментов данных, необходимо указывать уникальный идентификатор для каждого фрагмента и создавать методы API, возвращающие отдельные поля. То есть, если необходимой степенью детализации является абзац, то необходимо указывать id для каждого абзаца в большом тексте. Обращение к полю, например, к изображению, дате, автору или ISBN можно реализовывать обращением к серверу через API. Таким образом, используя несколько возможностей идентификации элемента коллекции можно сформировать сериализованные данные.

Передача текстовых данных через указанные форматы достаточно стандартна и не вызывает сомнений. Сложнее обстоит ситуация с передачей мультимедиа данных:

изображений, аудио, видео и др. Возможно как минимум два варианта решения этой задачи:

— передача ссылки на объект;
— кодирование с помощью base64 видео, изображения и др., таким образом превращая бинарный файл в строку.

В дальнейшем можно встроить эти сериализованные данные в ответ сервера и внедрить их на сайт клиента используя схемы dataÛRL. Однако использование base64 увеличивает объем данных примерно на 30%. Кроме этого данные не кэшируются браузером и будут загружаться при каждом просмотре страницы. Такой вариант передачи данных оправдан только в случае наличия жестких требований на обеспечение сохранности авторских прав на контент.

3.3. Обработка данных на сайте-клиенте

Встроить полученные через JSON или XML данные представляется отдельной задачей и может развиваться как минимум в двух направлениях:

— без поддержки исходного форматирования, с использованием данных как есть;
— с использованием форматирования на сервере.

В первом случае требуется только разобрать данные и вставить их на страницу в соответствии с некоторым своим шаблоном. Вариант с поддержкой исходного форматирования, такого как размер изображений, галереи, слайдеры, видео плеер, требует вместе с данными передавать с сервера CSS описание и JavaScript код.

Встроить данные на веб-страницу клиента можно, как известно, несколькими способами – встроить непосредственно в DOM, нарисовать на canvas или использовать iframe. Очевидно, наиболее гибким и эффективным способом встраивания на страницу информации из другого источника с поддержкой форматирования является использование iframe. При этом в качестве атрибута src у iframe должен быть указан адрес удаленного файла: это может быть HTML-документ, изображение или имя серверной программы, возвращающей данные. В нашем случае предполагается иметь возможность использования произвольного набора элементов отображения. Эти элементы могут идти не подряд, что не позволит использовать адрес файла, который будет загружаться во фрейм через атрибут src. На наш взгляд, содержимое фрейма необходимо передавать непосредственно в виде HTML документа и вставлять в атрибуте srcdoc.

3.4. Разработка инструментария

Наиболее перспективным вариантом создания платформы для интеграции мультимедиа электронных коллекций, с нашей точки зрения, является параллельная разработка API сервера и, на его базе, онлайн инструментария динамического формирования кода для обеспечения встраивания его в клиентские ресурсы. Основными требованиями при разработке инструментария являются:

— наличие уникальных идентификаторов (id) для всех элементов страницы коллекции, которые могли бы быть переданы в API;
— формирование полного HTML кода страницы вместе с CSS описанием и JS кодом для передачи форматирования и встраиваться на клиенте в iframe в виде srcdoc;
— передача, при реализации API сервера, текстовых данных непосредственно, а информации о мультимедиа объектах - в виде ссылок.

4. Пользовательские коллекции на уровне текущего сайта

Возможность включения избранных элементов в новую коллекцию может быть актуальна не только на уровне межсайтового взаимодействия. Как правило, достаточно

обширные коллекции предполагают внутреннее структурирование в рамках некоторых классификаторов. Для целого ряда предметных областей существуют формальные общепринятые системы таксономии. Так, например, в области петрографии для России актуальны как иерархическая классификация горных пород по Петрографическому кодексу, так и европейская классификация на основе диаграммы QAPF. Реализация веб-представления коллекции предполагает наличие отдельных полей, обеспечивающих связь с классификаторами и позволяющих фильтровать результаты вывода. Таким образом, пользователь просматривает коллекцию, последовательно выбирая очередной уровень иерархии, и не имеет возможности вывести на одну веб-страницу элементы разных ветвей таксономии.

Реализация пользовательских коллекций на базе исходной весьма востребована как в научных, так и образовательных целях. В этом случае инструментарий не предполагает разработки WEB-API вышеописанного типа, но требует определённых программных решений и создания соответствующих интерфейсов для зарегистрированного пользователя. Авторизовавшись, такой пользователь сможет отобрать необходимые элементы коллекции, разместить их на странице своей виртуальной коллекции и снабдить комментариями в произвольной форме.

Работа выполнена при поддержке Российского фонда фундаментальных исследований, проект 19-07-01012.

Литература

[1] Дублинское ядро. Материал из Википедии — свободной энциклопедии. URL: https://ru.wikipedia.org/wiki/%D0%94%D1%83%D0%B1%D0%BB%D0%B8%D0%BD%D1%81%D0%BA%D0%BE%D0%B5_%D1%8F%D0%B4%D1%80%D0%BE (дата обращения:15.06.2019).
[2] Google Scholar. URL: https://scholar.google.com/intl/ru/scholar/inclusion.html#indexing (дата обращения:15.06.2019).
[3] The Open Graph protocol. URL: http://ogp.me/ (дата обращения:15.06.2019).
[4] Optimize Tweets with Cards. URL: https://developer.twitter.com/en/docs/tweets/optimize-with-cards/guides/getting-started.html (дата обращения:15.06.2019).
[5] Соционет. URL: https://socionet.ru/ (дата обращения:15.06.2019).

Problems of Integration of Online Digital Collections

N.V. Borisov[1], V.V. Zakharkina[1], I.A. Mbogo[1], P.P. Shcherbakov[1]

[1] *St Petersburg University*

This article proposes a new approach to solving the problems of integration of online digital collections. The mechanisms of interaction between electronic collections, allowing directly embed in the formed multimedia collection (or scientific article with multimedia content) arbitrary elements from a remote collection (scientific article). Approaches to software implementation of the corresponding interfaces are described.

Keywords: online digital collection, integration of digital collections, multimedia content

Reference for citation: Borisov N.V., Zakharkina V.V., Mbogo I.A., Shcherbakov P.P. Problems of Integration of Online Digital Collections // Information Society: Education, Science, Culture and Technologies of the Future. Vol. 3 (Proceedings of the XXII International Joint

Scientific Conference «Internet and Modern Society», IMS-2019, St. Petersburg, June 19-22, 2019). - St. Petersburg: ITMO University, 2019. P. 67 – 73. DOI: 10.17586/2587-8557-2019-3-67-73

Reference

[1] Dublin Core. URL: https://ru.wikipedia.org/wiki/%D0%94%D1%83%D0%B1%D0%BB%D0%B8%D0%BD%D1%81%D0%BA%D0%BE%D0%B5_%D1%8F%D0%B4%D1%80%D0%BE (date of access:15.06.2019).
[2] Google Scholar. URL: https://scholar.google.com/intl/ru/scholar/inclusion.html#indexing (date of access:15.06.2019).
[3] The Open Graph protocol. URL: http://ogp.me/ (date of access:15.06.2019).
[4] Optimize Tweets with Cards. URL: https://developer.twitter.com/en/docs/tweets/optimize-with-cards/guides/getting-started.html (date of access:15.06.2019).
[5] Socionet. URL: https://socionet.ru/ (date of access:15.06.2019).