

БИБЛИОТЕЧНЫЕ ПОРТАЛЫ НА ОСНОВЕ XML-СЕРВЕРА ПРИЛОЖЕНИЙ

Ф. Е. ТАТАРСКИЙ, К. Е. ПАЧУЕВ

Научно-техническая библиотека

Томского политехнического университета, Россия

e-mail: fet@lib.tpu.ru, jack@lib.tpu.ru

Principal advantages of an XML Application Server for Digital Library Portal development are considered. Interactions of Z39.59 resources with XML applications are also discussed.

Введение

Семейство технологий XML завоевывает все более прочные позиции как средство построения web (XML) приложений. Построение порталов на основе XML-сервера приложений (далее XML-сервер) дает разработчикам, а следовательно, и пользователям ряд преимуществ. В то же время задача построения порталов, особенно библиотечных, требует рассмотрения вопроса интеграции XML-приложений и Z39.50 ресурсов (источников данных).

1. Преимущества использования XML-серверов приложений для построения порталов

1.1. Разделение данных, логики и представления

Это, наверное, одно из основных требований, предъявляемых к web-приложениям вообще и к порталам в частности. Оно диктуется, во-первых, необходимостью упрощать реализацию приложений. Во-вторых, как правило, в больших системах создание содержимого, web-дизайн и программная поддержка являются довольно трудоемкими задачами, требующими специальных навыков. Разделение данных, логики их обработки и представления на интерфейсах позволяет специалистам оперировать на разных уровнях отдельно.

Использование XML-сервера позволяет формулировать логику обработки данных в виде XSL-преобразований [1] максимально независимо от возможных изменений в структуре данных. Аналогично внешнее представление формулируется отдельно от логики обработки и от начальной структуры данных при помощи каскадных таблиц стилей (CSS) [2].

Особо подчеркнем роль CSS в представлении конечных документов. Спецификация CSS предусматривает ее использование не только для вывода на браузер. Существуют профили для мобильных устройств, принтеров, телевизионных приставок и т. п.

В качестве примера можно привести фрагмент XML-кода, использующегося в веб-сайте Научно-технической библиотеки Томского политехнического университета. Исходный текст раздела сайта “О библиотеке” выглядит следующим образом:

```
<webpage>
  <title>НТБ ТПУ</title>
  <sect1 id='center'>
    <title>Библиотека сегодня</title>
    <para>В настоящее время библиотека ... </para>
    .....
  </sect1>
  .....
</webpage>
```

Можно отметить, что этот документ не содержит инструкций форматирования, но только лишь структуру данных.

Исходный документ проходит ряд XSL-преобразований, результатом которых является минимально размеченный HTML-документ:

```
<div id="center">
  <span class="title">Библиотека сегодня</span>
  <p class="para">
    В настоящее время библиотека...
  </p>
  ...
</div>
```

В таком виде документ отправляется браузеру, который, в свою очередь, применяет к нему некоторую CSS для создания конечного представления:

```
.title {
  display: block;
  color: #485a7f;
  font-weight: bold;
  font-size: 10pt;
}
...
```

1.2. Тип документа

Наличие определений типов документов (DTD — Document Type Definition) для исходных XML-документов полезно по нескольким причинам. XML-процессор может выполнять автоматическую проверку XML-документов на соответствие DTD, т. е. корректность документа проверяется еще на этапе его создания.

Процесс создания DTD определяет выработку сбалансированного и ограниченного множества элементов, что значительно облегчает работу редакторам, ответственным за наполнение, и позволяет без труда выдерживать единый стиль оформления портала.

В настоящее время Интернет-сообществом развиваются различные DTD, описывающие такие типы документов, как книги, статьи (DocBook [3]), презентации (AxPoint [4])

и т. п. Документы, созданные в соответствии с этими DTD, теоретически позволяют получать библиографическое описание, online-версию и твердую копию из одного и того же исходного XML-документа.

1.3. Представление для различных устройств и в различных форматах

Необходимость представлять одни и те же данные в различных форматах и для вывода на различные типы устройств возникает довольно часто. Например, один и тот же ресурс требуется готовить для отображения в браузере, для печати на принтере, для отображения на экране мобильного устройства, или необходимо предоставлять различный дизайн одного и того же ресурса в зависимости от предпочтений пользователя.

Понятно, что поддержка отдельных версий ресурса для каждого случая не лучшее решение. Использование CSS версии 2.1 покрывает большую часть задач формирования конечного представления документа, а в случае, когда их возможностей недостаточно (может возникнуть необходимость представлять конечный документ в соответствии с различными DTD, например HTML и WML [5]), можно использовать возможность динамического выбора таблиц стилей.

Например, исходный XML-документ может иметь несколько директив `xml-stylesheet`, указывающих, какие стили применять для отображения на экране, а какие — для печати:

```
<?xml version="1.0" encoding="koi8-r" ?>
<?xml-stylesheet
  type="text/css" href="screen.css" media="screen"?>
<?xml-stylesheet
  type="text/css" href="print.css" media="print" ?>
<page>
  <title>Библиотека сегодня</title>
  <para>В настоящее время библиотека располагается в отдельном
  здании общей...
  </para>
  ...
</page>
```

В этом примере для отображения на экране браузер будет использовать стиль, находящийся в файле `screen.css`, а для печати — в файле `print.css` (пример проверялся с использованием браузеров Mozilla Firefox 9.2 и Internet Explorer 6.0).

1.4. Многоязыковая поддержка

Под многоязыковой поддержкой понимается возможность представлять одни и те же данные на различных языках (естественных) без необходимости поддерживать несколько версий одного web-приложения для каждого языка.

В терминах XML эта задача решается элементарно — простейшее XSL-преобразование может из начального ресурса, содержащего текст на всех поддерживаемых языках, выбрать только часть, интересующую конкретного пользователя.

В качестве примера такого преобразования можно привести следующий XSL-документ:

```

<xsl:stylesheet
  version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:param name='lang'>ru</xsl:param>
<xsl:template match='*[@lang!=\${lang}]' />
<xsl:template match='*[@lang=\${lang}] | *[string-length(@lang)=0]''>
  <xsl:copy>
    <xsl:copy-of select="@*" />
    <xsl:apply-templates />
  </xsl:copy>
</xsl:template>
</xsl:stylesheet>

```

Выполнив это преобразование над документом

```

<page>
  <title lang='ru'>Историческая справка</title>
  <title lang='en'>History</title>
  <para lang='ru'>Научно-техническая библиотека Томского
    политехнического университета (бывшего технологического,
    а затем политехнического института) - первая вузовская
    библиотека Сибири технического профиля. Она была открыта
    одновременно с институтом в октябре 1900 г.
  </para>
  <para lang='en'>The Scientific Library of Tomsk Polytechnic
    University (former Technological Institute and later
    Polytechnic Institute) is the first technical university
    library in Siberia. Its opening coincided with the
    beginning of studies of the institute in October, 1900,
  </para>
</page>,

```

мы получим документ

```

<page>
  <title lang='ru'>Историческая справка</title>
  <para lang='ru'>Научно-техническая библиотека Томского
    политехнического университета (бывшего технологического,
    а затем политехнического института) - первая вузовская
    библиотека Сибири технического профиля. Она была открыта
    одновременно с институтом в октябре 1900 г.
  </para>
</page>

```

1.5. Автоматическое формирование метаданных

Сам процесс формирования документа (разметка) подразумевает его индексирование. Например, несложным XSL-преобразованием можно извлекать из документа части, помеченные определенными тегами, и включать их содержимое в теги META заголовка конечного

HTML-документа. Таким образом, метаданные для ресурсов будут всегда поддерживаться в актуальном состоянии.

1.6. Построение интерактивных ресурсов

Сложно представить себе web-приложение без возможности интерактивного взаимодействия с пользователем. Традиционно для реализации таких возможностей используется технология CGI. В случае с XML-приложением CGI-параметры могут быть обработаны на этапе XSL-преобразований либо возможно использование расширений XML-сервера.

Расширения XML-сервера (иначе TagLib) — это произвольные функции, реализованные на некотором языке программирования и связанные с определенным тегом в XML-документе. Появление такого тега в XML-документе инициирует вызов соответствующей ему функции.

2. Взаимодействие с Z39.50

При всех своих достоинствах XML-сервер был бы мало полезен при построении библиотечных порталов, если бы в нем было затруднено взаимодействие с Z39.50 ресурсами. Относительно недавно опубликованный протокол SRW/SRU 1.1 (Search and Retrieve Web-service/Search and Retrieve by URL service) [6] (далее SRW) значительно облегчает решение этой задачи.

Протокол SRW — это XML-ориентированный протокол. Структура протокола была создана в результате 20-летнего опыта разработки протокола Z39.50. Протокол SRW является одновременно надежным и простым для понимания, сохраняя при этом все сильные стороны своего предшественника.

Конструкции протокола могут передаваться двумя способами: как SOAP (SRW)-сообщения или как параметры URL (SRU).

В настоящее время большинство Z39.50 серверов еще не поддерживают протокол SRW (хотя в YAZ Toolkit [7] поддержка SRW уже включена). Однако если использовать простейший сервис-посредник, транслирующий SRW-запросы в Z39.50 запросы и обратно, преимуществами этой технологии можно воспользоваться уже сейчас. В этом случае взаимодействие с Z39.50 сводится к тривиальному включению в XML-документ результатов запросов как XML-структур. Поскольку транспортом для SRW является протокол HTTP, такое включение осуществляется стандартными средствами XSL-процессора.

Демонстрацией системы, реализующей рассмотренные выше положения, может служить web-сайт Научно-технической библиотеки Томского политехнического университета [8], а также OPAC-система Научно-медицинской библиотеки Сибирского государственного медицинского университета [9], которая демонстрирует возможности взаимодействия XML-приложения с Z39.50 серверами через сервис-посредник SRU.

Список литературы

- [1] THE Extensible Stylesheet Language Family (XSL) / XSL Working Group, 2004. <http://www.w3.org/Style/XSL/>
- [2] CASCADING Style Sheets / Bert Bos, 2004. <http://www.w3.org/Style/CSS/>

- [3] DocBook / Oasis, O'Reily, Norman Walsh, 2004. <http://www.docbook.org/>
- [4] AxPoint — XML Based Presentations / Matt Sergant, 2001. <http://axpoint.axkit.org/>
- [5] Open Mobile Alliance / Open Mobile Alliance Ltd., 2002. <http://www.openmobilealliance.org/>
- [6] SRW — Search/Retrieve Web Service / Z39.50 International: Next Generation, 2004.
<http://www.loc.gov/z3950/agency/zing/srw/>
- [7] YAZ / IndexData ApS, 2004. <http://www.indexdata.dk/yaz/>
- [8] НАУЧНО-ТЕХНИЧЕСКАЯ библиотека Томского политехнического университета.
<http://www.lib.tpu.ru/>
- [9] НАУЧНО-МЕДИЦИНСКАЯ библиотека Сибирского государственного университета.
<http://medlib.tomsk.ru/>

Поступила в редакцию 18 марта 2005 г.