# Применение протокола SRW/U на примере стандартных задач

# Applying SRW/U Protocol to Standard Problems

# Застосування протоколу SRW/U на прикладі стандартних задач

*Колобов О. С.*
*Институт сильноточной электроники (ИСЭ) СО РАН, Томск, Россия*

*Oleg S. Kolobov*
*Institute for High-current Electronics, Russian Academy of Sciences Siberian Branch, Tomsk, Russia*

*Колобов О. С.*
*Інститут високоструменевої електроніки (ІВЕ) СО РАН, Томськ, Росія*

Дано описание подхода к реализации двух стандартных библиотечных приложений на основе протокола SRW/U. Это система удаленного доступа к электронному каталогу и единая точка доступа к распределенному электронному каталогу. Показано как приложения, базирующиеся на протоколе SRW/U, могут использовать XML ориентированные технологии и сосуществовать в окружении Z39.50-приложений.

The approach towards the realization of two standard library applications on the basis of SRW/U Protocol is described. These are the system of remote access to electronic catalog and interconnect access point for distributed electronic catalog. The author demonstrates how the applications based on SRW/U Protocol may employ XML-oriented technologies and co-exist with Z39.50 environment.

Розглянуто підхід до реалізації двох стандартних бібліотечних додатків на основі протоколу SRW/U. Це система віддаленого доступу до електронного каталогу і єдина точка доступу до розподіленого електронного каталогу. Показано як додатки, що базуються на протоколі SRW/U, можуть використовувати XLM орієнтовані технології та співіснувати в оточенні Z39.50-додатків.

## Введение

Протокол SRW/U[1] - один из проектов группы ZING[2]. Первая спецификация протокола была опубликована в 2003 году. Сегодня протокол SRW/U позиционируется как частная вариация протокола Z39.50[3] применительно для *web-среды*, как обобщенный протокол поиска и извлечения федеральной информации[4] и одобрен к регистрации в NISO .[5].

SRW/U протокол использует легкодоступные современные информационные технологии и применение протокола доступно двумя способами - либо через протокол SOAP, либо через URI. Это в значительной степени делает дружественными *шлюзы доступа* к базам данных на основе протокола SRW/U. Например, при использовании протокола SRW/U через URI, что для краткости называют SRU (Search Retrieval by URL), мы можем построить запрос к различным базам данных основываясь только на методе GET протокола HTTP:

- http://library.tsc.ru:9000/hcei?operation=searchRetrieve&version=1.1&query=perl
- http://z3950.loc.gov:7090/voyager?operation=searchRetrieve&version=1.1&query=perl
- http://www.google.com/search?operation=searchRetrieve&version=1.1&query=perl

Для начального ознакомления с протоколом SRW/U рекомендуем обратиться к документу *"SRW: Search/Retrieve Webservice* [6].

## Он-лайн доступ к электронному каталогу

*On-line Public Access Catalogue (OPAC)* – система удаленного доступа к электронному каталогу для выполнения задач поиска и извлечения библиографических описаний документов с последующей возможностью выполнения задачи удаленного заказа документа.

Одним из применений протокола SRW/U является решение задачи по созданию шлюзов доступа к базам данных. Таким приложением является CUBA-OPAC[7] (далее просто OPAC), которое использует протокол SRW/U (версии 1.1) для поиска и извлечения записей в библиографических базах данных.

## XML-сервер и конвейерная обработка XML-документа

Приложение OPAC построено на основе XML-сервера, и все публикуемые документы в OPAC изначально имеют XML-форму. Такой подход позволяет применить конвейерную обработку XML-документа различными XML-процессорами.

Конвейерная обработка XML-документа привлекательная тем, что на каждом этапе можно задействовать различные XML-процессоры. Один XML-процессор, к примеру, может выполнять трансформирование XML-документа из одной формы в другую, другой может выполнить программный код, который ассоцииро-

ван с XML-фрагментом документа. Данная возможность используется в OPAC и достигается путем применения техники множественной трансформации XML-документа как показано на рис.1.

```
(XML)=>[XSP]=>(XML)=>[XSLT]=>(HTML)

() - файл, [] - процессор
```

**Рис. 1**

В приложении OPAC используются два типа процессоров - процессор с языка XSP (Extensible Server Pages) (см. проекты *Apache Cocoon*[8], *Apache AxKit* [9]) и процессор с языка *XSLT* [10].

### Три этапа обработки XML-документа в OPAC

Типичная обработка XML-документа в OPAC выполняется в три этапа, как это показано на рис. 2

```
(XML)=>[1:XSP]>(XML)=>[2:XSLT]=>(XML)=>[3:XSLT]=>(HTML)

() - файл, [] - процессор
```

**Рис. 2**

Этап 1
Обработка XML-документа XSP-процессором.
На этом этапе происходит обращение к базе данных через протокол SRW, если в теле XML-документа определено пространство имен `srw` и указан XML-фрагмент с элементами SRW запроса, как это показано на рис. 3.

```
<?xml version="1.0" encoding="koi8-r"?>
<?xml-stylesheet href="NULL" type="application/x-xsp"?>
<?xml-stylesheet href="stage_2.xsl" type="text/xsl"?>
<?xml-stylesheet href="stage_3.xsl" type="text/xsl"?>
<xsp:document
 xmlns:xsp="http://apache.org/xsp/core/v1"
 xmlns:srw="http://www.loc.gov/zing/srw/"
>
<webpage>
 <srw:searchRetrieveRequest>
  <srw:url>http://library.tsc.ru:9000/hcei</srw:url>
  <srw:version>1.1</srw:version>
  <srw:query>dc.title=история</srw:query>
  <srw:startRecord>1</srw:startRecord>
  <srw:maximumRecords>5</srw:maximumRecords>
  <srw:recordSchema>marcxml</srw:recordSchema>
  <srw:recordPacking>xml</srw:recordPacking>
 </srw:searchRetrieveRequest>
</webpage>
</xsp:document>
```

**Рис. 3**

Для взаимодействия с базами данных по протоколу SRW/U в OPAC используется *библиотека тегов XSP* - `AxKit::XSP::SRW` [7], которая задействуется всякий раз, когда встречается XML-тег принадлежащий пространству имен `srw`.

Ответ на SRW запрос представляется как XML-фрагмент, который в последующие этапы будет обработан с помощью шаблонов (templates) языка XSLT.

**Этап 2**

Обработка XML-документа с использованием правил-шаблонов языка XSLT в файле `stage_2.xsl`.

На этом этапе обрабатывается XML-документ, который содержат ответ на SRW запрос (XML-фрагмент). После обработки генерируется новый XML-документ.

**Этап 3**

Окончательная обработка XML-документа также на основе правил-шаблонов XSLT в файле `stage_3.xsl`.

Задача этого уровня собрать все компоненты в единый HTML-документ.

Как это видно, на 1-ом этапе идет обращение к базе данных через SRW протокол, на 2-ом этапе обрабатывается ответ из базы данных и на заключительном - 3 этапе формируется HTML-документ, который и будет отправлен клиенту.

Каждый из этапов является независимым, что позволяет разделить обработку XML-документа на несколько логических уровней.

### MARC записи в OPAC

Традиционно библиографические записи представляются в форме MARC ISO 2709, но в среде SRW/U все данные представляются в XML-синтаксисе и для преобразования MARC-записей используется транспортная XML-схема MARC21XML/Slim [11].

Для идентификации XML-схемы записи как в запросе так и в ответе на запрос используется стандартный идентификатор. Это позволяет автоматически определить в какой XML-схеме запись и подключить соответствующую обработку для записи. На сегодняшний день протоколом SRW/U определены несколько стандартных XML-схем для записей, но *обязательной* для поддержания в SRW/U-приложениях является лишь DC (Dublin Core) [12].

Приложение OPAC использует транспортную схему MARC21XML/Slim как для MARC21, так и для RUSMARC записей. Недостатком такого подхода является то что невозможно применить формальный способ для идентификации диалекта MARC-записи, и на сегодняшний день в OPAC применяются неформальные способы идентификации диалекта MARC-записи.

### Единая точка доступа к распределенным Z39.50 базам данных

Совокупность распределенных Z39.50 баз данных будем называть *распределенным электронным каталогом*. Такие структуры сегодня не редкость в сети и их появление было инициированно процессом создания библиотечных консорциумов. Как правило в консорциум входят организации с разной ведомственной принадлежностью, и являются правообладателями баз данных. В таких условиях консорциум организует централизованный доступ к базами данных партнеров.

Создание единой *точки доступа* к базам данных для консорциума является привлекательной необходимостью по нескольким причинам: возможность осуществлять контроль; собирать статистические данные; экономия ресурсов и т.п.

Рассмотрим построение единой точки доступа к распределенному электронному каталогу на основе приложение YAZPROXY[13]. YAZPROXY это Z39.50 прокси-сервер с помощью которого возможно организовать электронный сервис SRW/U для Z39.50 баз данных.

### Описание идеи

Идея состоит в том, чтобы создать в сети единую точку доступа ко всем базам данных распределенного электронного каталога по протоколу SRW/U. Под *точкой доступа* мы понимает адрес прокси-сервера, через который работает клиент протокола SRW/U с распределенным электронным каталогом.

Пусть в распределенный электронный каталог входят следующие базы данных, доступные через протокол Z39.50:

```
1. z39.50s://library.tsc.ru:2100/hcei;rs=rusmarc;esn=f
2. z39.50s://z3950.lib.tpu.ru:9999/book+prd+tpu;rs=rusmarc;esn=f
3. z39.50s://z3950.loc.gov:7090/voyager;rs=usmarc;esn=f
```

**Рис. 4**

Пусть единая точка доступа по протоколу SRW/U к этим базам данных есть:

```
http://libraray.tsc.ru:9000
```

**Рис. 5**

Адрес каждой базы данных в SRW/U представляется в виде URL и в нашем случае необходимо поставить в соответствие каждой Z39.50 базе данных уникальный адрес в SRW/U на основе URL единой точки доступа:

```
1. http://library.tsc.ru:9000/hcei
2. http://library.tsc.ru:9000/ucat
3. http://library.tsc.ru:9000/voyager
```

**Рис. 6**

Заметим, что база данных `ucat` логически объединяет несколько Z39.50 баз данных (`book, prd, tpu`), а база данных `voyager` содержит записи в формате MARC21, а не в RUSMARC как все остальные базы данных.

### Клиент

Клиент протокола SRW/U может выполняет запросы к базам данных через единую точку доступа, не акцентируя внимания на том что работает с распределенными по сети Z39.50 базами данных. Например, в приложение OPAC, для получения описаний базы данных распределенного каталога необходимо выполнить *Explain Request* к каждой из баз данных:

```
<?xml version="1.0" encoding="koi8-r"?>
<?xml-stylesheet href="NULL" type="application/x-xsp"?>
<?xml-stylesheet href="stage_2.xsl" type="text/xsl"?>
<?xml-stylesheet href="stage_3.xsl" type="text/xsl"?>
<xsp:document
 xmlns:xsp="http://apache.org/xsp/core/v1"
 xmlns:srw="http://www.loc.gov/zing/srw/"
>
<webpage>
 <srw:explainRequest>
  <srw:url>http://library.tsc.ru:9000/hcei</srw:url>
  <srw:version>1.1</srw:version>
  <srw:recordPacking>xml</srw:recordPacking>
 </srw:explainRequest>
 <srw:explainRequest>
  <srw:url>http://library.tsc.ru:9000/ucat</srw:url>
  <srw:version>1.1</srw:version>
  <srw:recordPacking>xml</srw:recordPacking>
 </srw:explainRequest>
 <srw:explainRequest>
  <srw:url>http://library.tsc.ru:9000/voyager</srw:url>
  <srw:version>1.1</srw:version>
  <srw:recordPacking>xml</srw:recordPacking>
 </srw:explainRequest>
</webpage>
</xsp:document>
```

**Рис. 7**

В результате обработки XML-документа в приложении OPAC, на HTML-страничке будут представлены ZeeRex[14] описания все баз данных распределенного каталога в том виде, который определен с помощью правил шаблонов XSLT.

Альтернативный способ получения ZeeRex описаний баз данных, но уже через HTTP GET запрос:

```
1. http://library.tsc.ru:9000/hcei?operation=explain&version=1.1
2. http://library.tsc.ru:9000/ucat?operation=explain&version=1.1
3. http://library.tsc.ru:9000/voyager?operation=explain&version=1.1
```

**Рис. 8**

Результатом HTTP GET запроса, в этом случае, будет XML-документ, который содержит в себе запись в XML-схеме ZeeRex.

### Заключение

Протокол SRW/U не является стандартом сегодня, но библиотечные приложения построенные на основе этого протокола, как показано в работе, могут выполнять поставленные задачи уже сегодня.

Приложение OPAC это пример реализации пользовательского интерфейса для работы с базами данных на основе протокола SRW/U, который может быть расширен для реализации более крупных проектов — интернет-порталов, систем управления контентом.

Программа YAZPROXY позволяет воплотить идею создания единой точки доступа через SRW/U протокол в реальность. С помощью YAZPROXY можно предложить SRW/U сервис для уже существующих в сети Z39.50 баз данных. А также разработчики добавили в программу ряд интересных и полезных свойств:

- *распределение нагрузки на Z39.50 сервер*

Если одна и та же Z39.50 база данных расположена на нескольких хостах YAZPROXY позволит равномерно распределить нагрузку, чередуя адреса хостов;

- *кеширование*

YAZPROXY позволяет выполнять кеширование: транспортных соединений, запросов, и более того, результирующих множеств, если к Z39.50 базам данных осуществляется *открытый доступ*;

- *конвертирование записей*

Коммуникативные форматы MARC21, UNIMARC, RUSMARC и др. являются стандартными для библиотечного сообщества. И задача конвертирования записей в другую форму является регулярной задачей. Используя предложенный подход можно дополнить единую точку доступа таблицами стилей на языке XSL, которые позволят преобразовывать "на лету" записи из одной XML-схемы в другую. Например из MARC21XML/Slim в DC или MODS[15].

### Ссылки


```
1.  http://www.loc.gov/z3950/agency/zing/srw/
2.  http://www.loc.gov/z3950/agency/zing/
3.  http://www.loc.gov/z3950/agency/
4.  http://www.search.gov/interop/requirements.html
5.  http://www.niso.org/registration/registration_approved.html
6.  http://srw.cheshire3.org/SRW-1.1.pdf
7.  http://sibtechcenter.ru/~cuba/
8.  http://cocoon.apache.org/
9.  http://www.axkit.org/
10. http://www.w3.org/Style/XSL/
11. http://www.loc.gov/standards/marcxml/
12. http://www.dublincore.org/
13. http://www.indexdata.dk/yazproxy/
14. http://explain.z3950.org/
15. http://www.loc.gov/standards/mods/
```