UDC 004.62

WEB-RESOURCES PARSING IMPLEMENTATION FOR ACQUISITION OF MEDIA CONTENT

DENIS TSVIRKO, SERGEI SURTO
Polotsk State University, Belarus

There is great amount of media content in the world, which constantly grows after release of new films, serials, etc. In order to collect all this information together, it is necessary to customize web-resources parsing process, which is constantly able to update the single database of media content. The necessity of writing is based on the fact that there is no single parser which allows to obtain all necessary information from web-sources.

Site parsing can be divided into three phases: obtainment of the content in the original view, extraction and transformation of data, report generation.

In the first phase a web page is downloaded for its further analysis and extraction of necessary information. Different libraries are provided for almost every language which allow to get the information from the site.

In the second phase parser extracts necessary information from the code of web page thereby separating necessary information from the programming code of the page. For each resource its own data collection rule is created. Before these programmers analyze the page with data and work out a solution where the necessary information is displayed, where it is hidden, which way it is displayed in. All this process, in fact, means the customization of parser.

The last phase is report generation. Parser transforms obtained data in the type of information that was set. It might be text files, or data record directly in the database of the site, or another type of final information obtainment [1]. Data received during the parsing is necessary to enter into temporary storage. After preview and settings, they can be used in the main storage. Also it is necessary to enter the logging of the parser's work, in order to be able to respond on the errors which appeared in the work.

The are some approaches to data parsing: analysis of DOM tree, Xpath, Regex.

The first approach is based on analysis of DOM tree. Using this approach, data may be obtained by ID directly: name or other attribute of the tree element (such elements can serve as paragraph, table, block, etc.). Also if element is not specified by any ID, that you can get to it by some unique way such as getting down the DOMs tree.

You should use a DOM parser when:

– you need to know a lot about the structure of a document;

– you need to move parts of the document around (you might want to sort certain elements out, for example);

– you need to use the information in the document more than once.

When you parse an XML document with a DOM parser, you get back a tree structure that contains all of the elements of your document. The DOM provides a variety of functions you can use to examine the contents and structure of the document.

Advantages of this approach are the following:

– you can get data of all types and difficulty level;

– if you know the placement you can get its value by writing the path to it.

Disadvantages of this approach are the following:

– different HTML / JavaScript tools generate DOM tree in various ways, so it is necessary to bind to a specific tool;

– the path of the element can be changed, so, as a rule, such parsers are designed on short-term period of data collection;

– DOM-path can be difficult and now always definite [2].

The next evolutionary step of DOM tree analysis is the use of XPath — it means the paths, which is widely used for parsing XML data. The nature of this approach is to describe, with help of some simple syntax, the path to the element without necessity of gradual movement down along the DOM tree.

Xpath provides a simple way to get an enumeration of necessary elements of HTML page, for example, the list of new series, which will present a container, that stores text and graphic information.

HtmlAgilityPack tool is necessary for the adequate work of Xpath. Xpath can work only with valid XML. HTML page in most cases is not valid, because some browsers do not demand the valid XML – just one unopened tag and standard components for work with XPath and it simply cannot download the document for further processing [3].

Following are the key components of XPath are required:

– Structure Definitions - XPath defines the parts of an XML document such as elements, attributes, texts, namespace, processing-instructions, comments, and document nodes.

– Path Expressions XPath provides powerful path expressions and select nodes or list of nodes in XML documents.

– Standard FunctionsXPath provides a rich library of standard functions for manipulation of string values, numeric values, date and time comparison, node and QName manipulation, sequence manipulation, Boolean values etc.

– Major part of XSLTXPath is one of the major element in XSLT standard and it must have data in order to work with XSLT documents.

– W3C recommendation XPath is official recommendation of World Wide Web Consortium (W3C).

XPath uses a path expression to select node or list of nodes from an xml document. XPath parser is used to navigate XML Document only. It is better to use DOM parser for creating XML.

One of the approaches is based on the parsing with the use of regular expressions. Regular expression (regexp) – is a mechanism, which allows to assign the sample for a string and to implement the data search, which is appropriate to this sample in assigned text. In addition, extra function for work with regexp allow to get found data in the form of string array, produce replacement in the text according the sample, split the string according the sample, etc. However, the main function, which all others are based on, is exactly the function of data search in text appropriate to sample described in syntax of regular expressions. Regular expressions are necessary only for extraction of data, which has strict forms.

Regular Expressions, most often known as "RegEx" is one of the most popular and widely accepted technological means used for parsing the specific data contents from large texts. A regular expression is a specific pattern or a specific sequence of some special characters (known as "meta characters") that gives you an ability to concisely and flexibly "match" or "capture" (specify and recognize) strings of text, such as sequence of particular characters, words, or patterns of characters.

The technique of extracting only the required data and neglecting all the other unnecessery content from the given large text with the help of Regular Expressions is nothing but "RegEx Parsing".

There are many tools that are used for RegEx parsing. Some of them are RegexBuddy, RegexPal, RegexMagic, RegexPlanet and Rubular.

The main thing about regular expressions that makes it so simple and useful for all is its syntax. The regular expression syntax is declarative. The pattern "looks like" what you want to match. Another most important thing that makes regular expression to spread their magic very quickly is its vast and powerful set of meta characters. Regular expressions are blessed with very rich and powerful set of meta characters. Each of these meta characters has its unique meaning in itself and plays an important role independently or dependently in making the regular expression more powerful.

The most appropriate approach for obtainment media content is the approach based on the analysis DOM tree. It provides to get necessary information regardless the validity of html-page. During the analysis of DOM tree, you can get data of all types and difficulty levels. The example of parser which implements the obtainment of information about the serials is described on listing 1.

Listing 1. Example of parser implementation

```
// create a DOM from the URL
$html = file_get_html('http://www.lostfilm.tv/series/');
// path for saving pictures
$path = './images/serials/';
// select all the div tags with the class row on the page
// this block contains information about the show
foreach ($html->find('div[class=row]') as $element) {
// search for a link to image
    $imgUrl = $element->find('img[class=thumb]')->src;
    // retrieve image by link
```

```
        $img = file_get_contents($imgUrl);
        // get the series name
        $nameSerials = $element->find('div[class=name-ru]')->text;
        // save information about the sereal in the database
        (new Serials)
            ->setName($nameSerials)
            ->setImagPath($path . $nameSerials)
            ->save();
         // save the picture to the server in the specified location
         file_put_contents($path . $nameSerials, $img);
    }
```

REFERENCES

1. Что такое парсиг сайтов [Электронный ресурс] // Форум продвижения сайтов. – Режим доступа: https://ruseo.net/chto-takoe-parsing-saytov-t16056.html. – Дата доступа: 11.02.2018.
2. Подходы к извлечению данных из веб-ресурсов [Электронный ресурс] // Форум хабрахабра. – Режим доступа: https://habrahabr.ru/post/99918/. – Дата доступа: 11.02.2018.
3. Преимущества Xpath для парсинга HTML [Электронный ресурс] // Infostart. – Режим доступа: https://infostart.ru/public/295334/. – Дата доступа: 11.02.2018.