myPortal: Robust Extraction and Aggregation of Web Content

Marek Kowalkiewicz
Poznan University of Economics
Al. Niepodleglosci 10
60-967 Poznan, Poland
+48(61)8543631
marek@kowalkiewicz.net

Tomasz Kaczmarek
Poznan University of Economics
Al. Niepodleglosci 10
60-967 Poznan, Poland
+48(61)8543631
t.kaczmarek@kie.ae.poznan.pl

Witold Abramowicz
Poznan University of Economics
Al. Niepodleglosci 10
60-967 Poznan, Poland
+48(61)8543381
witold@abramowicz.pl

ABSTRACT

We demonstrate myPortal – an application for web content block extraction and aggregation. The research issues behind the tool are also explained, with an emphasis on robustness of web content extraction.

1. INTRODUCTION

The problem of personalized access to information has been researched since the information overload phenomenon had been observed. Personalized portals were expected to solve the problem in the area of Internet, and at least a few such solutions have been brought up to the public. The most common approach was to provide users with sets of sections (portlets, webparts) to construct portal pages from. One of the shortcomings of the proposed approach was that the sections had to be explicitly prepared by portal owners. Currently no major portal content providers offer functionalities to personalize views on their portals. However there is a growing interest in that area from providers of search and aggregation services (Personalized Google Home or Microsoft's Windows Live). Still, the choice in these cases is restricted to the predefined set of content sources. Recently, new approaches have been researched in order to provide personalizing possibilities without the mentioned shortcomings. The predominant, and very promising, area is web content extraction and aggregation. A natural application of web content extraction is to populate databases with content from the Web. Another interesting application area is to provide views on web content, conceptually similar to those in databases, showing only relevant fragments of web documents.

Researchers dealing with information on the Web define the so called web content. It is similar to unstructured information in information systems. The important difference is that there is an at least implicit structure (HTML) in web content. Semantically independent and distinguishable units of web documents, such as tables, content sections etc. inside web pages, are called web content blocks. Web content blocks are substructures present in the HTML structure of Web documents.

VLDB '06, September 12–15, 2006, Seoul, Korea.

Sources of information on the Internet are often dynamic. That means that there is an underlying structure of a Web document (often called template) which is populated with textual and graphical information. In most cases, the structure is static, whereas the content filling the structure changes. Structure changes are most often caused by redesign of a website. The dynamics of content sources implies that identified web content blocks may change their location within document structure, especially in cases where document structure is changed. These location changes are often subtle, for example resulting from introduction of new content blocks (for instance advertisements), but at the same time they are very hard to cope with automatically. To deal with the dynamics of Web sources, information extraction tools have to be applied.

The process of extracting content blocks from web documents is called web content extraction. Methods applied in this area may be grouped into three most common approaches: based on unique ID, based on contextual information, and based on a tree view of the document. Unique ID based extraction requires existence of IDs assigned to individual content blocks (HTML elements). The IDs may be used to locate requested content, and extract only sections marked with proper ID. Contextual information based methods treat web document as a plain text document and extract content blocks based on text patterns surrounding them. Tree view based methods view web documents as tree-like structures (HTML is a labeled ordered rooted tree) and extract nodes or subtrees of the tree. The unique ID methods are usable only in Intranets, where unique IDs may be introduced and assigned to individual HTML elements. Contextual information method is not robust to structure changes, and it does not make use of the implicit structure of web documents. The tree view based method is believed to be more robust than contextual and is most popular in web content extraction tools. The predominant language used in extracting web content based on a tree view is XPath [2, 4].

Another interesting research area contributing to reducing information overload is web content aggregation, were extracted web content blocks are put back into a web document structure. Research issues here include design of web documents, effective placement of web content blocks, filtering less relevant content blocks, and applying dynamic web document reconstruction to include only those content blocks that are most relevant to the user at the moment.

The research and implementation problem that motivated us during development of the demonstrated system – myPortal – was

to build a content extraction and aggregation system that would be characterized with two features: simplicity of use, and high robustness of the solution [6]. The most challenging goal was to improve currently used web content extraction methods to achieve higher robustness.

In this demo paper we show the applicability of myPortal in web content extraction and aggregation for Web browsing. We especially focus on aspects of robustness of content extraction and ease of use of the application. The same technique as presented in the demo can be used for populating databases with content extracted from the Web. In the following sections of this paper we relate our work to the achievements of other researchers in the field, describe the application (including an overview of research problem and results), and demonstrate the application use case.

2. RELATED WORK

An excellent analysis of web information extraction tools and methods has been prepared by Laender et al [7]. The survey is also a thorough analysis of web content extraction research. A number of approaches towards web content extraction, especially in the area of web page segmentation, were put forward by Ma et al [3, 5, 9]. Abe and Hori [1] have focused on one specific aspect of web content extraction: robustness of XPath expressions. However, they tested their approach on a relatively small sample of webpages. A thorough analysis of other approaches to web content extraction was included in another publication of the authors [6].

Approaches to extracting and aggregating content blocks can be found in systems that offer RSS and ATOM functions. The most important shortcoming of RSS and ATOM feeds is that they have to be explicitly defined for each webpage, and that the internet users are limited to choosing only from the prepared feeds.

An important issue in our research is to give complete freedom to Internet users, so that they can choose which web pages to extract content from, and which exact web content blocks to extract. There should be no requirements on the website provider side, apart from a reasonable requirement that the web documents should be represented using HTML. In our approaches we focus on improving robustness of web content extraction methods. We believe that the currently available methods are not robust enough.

3. APPLICATION DESCRIPTION

MyPortal provides users with simple interface to extract and aggregate web content. It uses the tree based extraction method – relative XPath queries over the XHTML DOM tree.

3.1 Absolute and relative XPath extraction methods

The XPath language has been designed to extract elements' and attributes' content form XML documents. Its query capabilities are broad, however for web content extraction it has been mainly used with absolute expressions – pointing form the root of the document tree to the element to be extracted (extracted element). Following the work of Abe and Hori [1] we used the other XPath feature – relative queries. Our goal was to define the path not from the root, but from some element inside the document structure (reference element) (Figure 1).

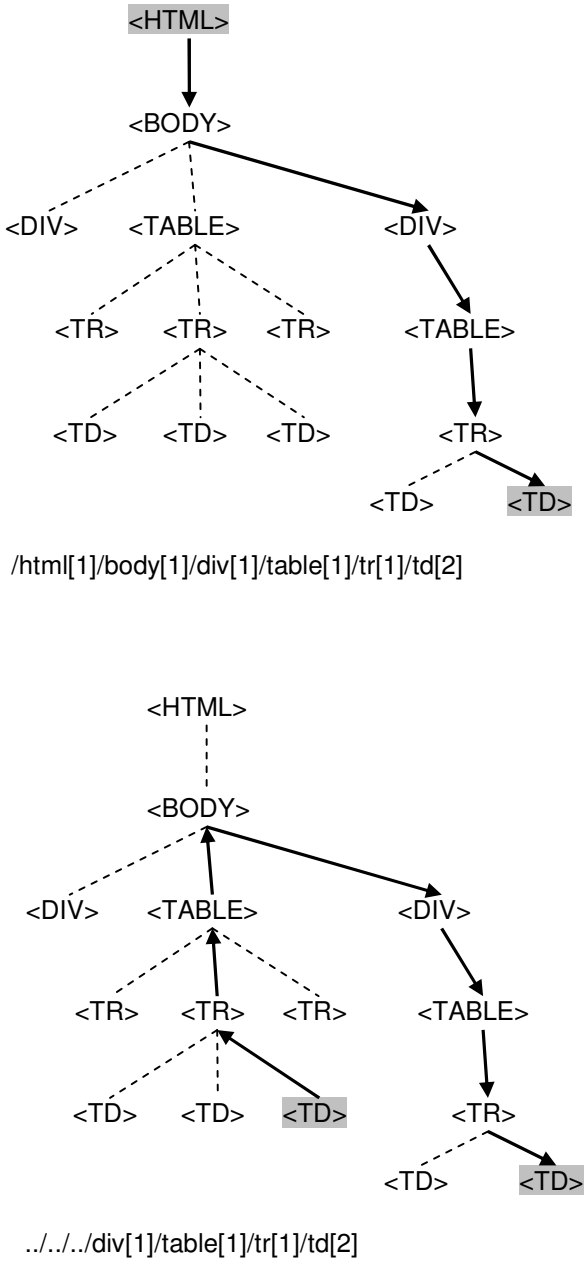

Figure 1 Absolute (top) and relative (bottom) XPath expressions. Starting (reference) and extracted nodes are highlighted. In this case relative XPath expression has a longer path, but in most cases the relative path is shorter than the absolute one.

This approach has two appealing features. First it mimics human behavior. When looking at web pages, users typically seek distinguishing elements like headers, bolded text or graphically emphasized blocks. If the page structure changes, we are able to find the information thanks to the fact that it usually stays close (or in some constant relation) to its header. Secondly, relative queries alleviate one of the vulnerabilities that all content

extraction systems are subject to: structure changes of dynamic web content sources. If the document structure changes, it would likely influence the performance of the absolute XPath expression. With relative ones we are only vulnerable to the changes within the smallest sub-tree of the DOM tree that includes our reference and extracted elements.

The proposed approach may be also applied in the area of information and data integration from the Web. It may be considered as an improvement of information extraction methods.

3.2 Application modes

The application works in two modes: query definition and portal construction. The former consists of three steps. First, the content source context has to be identified. It comprises not only from source URL, but also HTTP variables that are send via GET or POST methods, HTTP headers and cookies – all the information needed to ensure that the requested page will be properly returned by web server. After collecting source location information, myPortal requests users to choose elements (web content blocks) to be extracted. Then users choose the reference element with a simple pointing and click interface on the web page in the browser. Relevant content blocks (defined by tables, paragraphs, divisions and other structural elements) are highlighted dynamically with bounding boxes as they are selected by users (Figure 2). As soon as myPortal identifies both extracted and reference element locations in the document tree, it is able to construct a query which consists of textual content of a reference element and relative XPath expression from the reference to the extracted element.

User is able to define multiple queries on different content sources and store them in query library. They can be later used when defining a personalized portal. There is a simple interface that allows dividing a portal page into multiple parts and binding defined queries to each of them. As soon as the portal has been defined, the next mode is applied – portal construction.

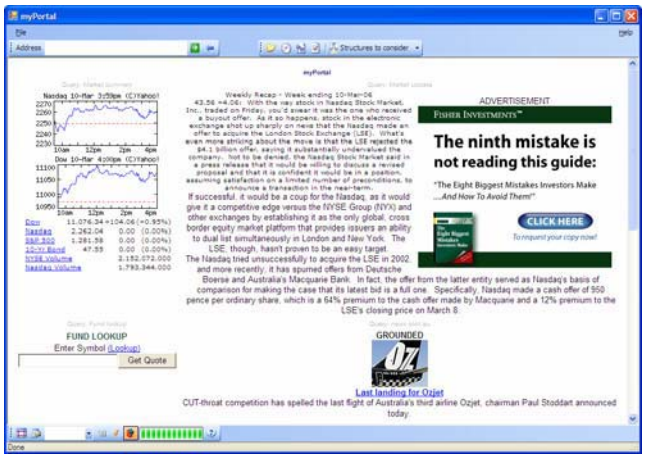

Figure 2 myPortal: a result of XPath query evaluation – aggregated view of extracted Web content blocks.

During the portal construction process – the second mode – each of the bound queries is evaluated. First, based on source context, the request is made against appropriate Web server. Then the textual content of the reference element (from the query) is searched for within a downloaded document. For each of the found references, relative XPath expression is evaluated. The outcome of this evaluation is placed in the proper part of personalized portal, which concludes content aggregation step, and results in a ready to view document (Figure 2).

3.3 The problems encountered

The problem of erroneous web page coding is a well known issue which usually aggravates the information extraction problem. We applied HTML Tidy [8] for each of the processed pages to correct badly formed HTML documents. One of the reasons was to convert the page to XML, which was required to use XPath.

The main problem introduced with relative XPath addressing is that multiple elements containing the same reference text can be found. Interestingly, during our study, which involved testing around four thousand XPath queries, we did not encounter even a single occurrence of this problem. It is probably due to the fact that even if multiple elements containing the same reference text were found, they were placed in completely different parts of DOM tree, and therefore evaluating moderately complex XPath expression relative to them brought empty result sets. The only results obtained were ones for the "right" reference element. Therefore the problem of multiple instances of relative element can be neglected.

One of the main issues of information extraction is the robustness of selected method (its ability to resist to changes in structure of web documents). Our experiments show that the relative XPath expressions are on average significantly more robust than the absolute XPath expressions. We evaluated 3910 queries of both types on several content sources (archived web pages from 2004). The study showed that absolute queries were robust in 47% of cases, while relative were robust in 76%. Robustness was defined as an average extraction success ratio measured as the percentage of correctly extracted content blocks in the whole year. The experiments also revealed that if website layout does not change dramatically (for example as a result of a complete website redesign), the relative XPath method's robustness is on average more than 95% (while absolute XPath stays below 75%) – these results were obtained by restricting the set of tested websites. The study was conducted using myPortal.

4. DEMO DESCRIPTION AND USE CASE OVERVIEW

Consider a user, who is interested in accessing information helpful in stock investments. He wants to receive general information about the market: main indicators, market news, and market commentary. He also wants to see the most important news of the day and be able to lookup stock prices quickly.

Our goal is to provide the user with an aggregated view containing the required information, ignoring all information that is not relevant – therefore containing only particular content blocks. We reach our goal by allowing the user to navigate to the web documents using myPortal and select desired content blocks with a point and click interface. In this demo we navigate to a site with market news and to a website with general news. Afterwards we select relevant content blocks and construct a new portal with a point and click interface of myPortal. During the demo we also show, based on archive data, that the queries are robust to changes in document structure. The viewers of the demo will be asked to

suggest any website to see how myPortal works in different environments.

Figure 3 A webpage with selected web content block. The selected content block is bordered; content block header is highlighted with a dashed border.

In this demo we will demonstrate the application of myPortal, its ease of use and robustness of content extraction. Several sources of Web information will be chosen within the sources, relevant content blocks will be chosen and pointed by users (Figure 3), and finally an aggregated web document will be generated.

Throughout the demo, we will show how to instantly create content block extraction definitions (relative XPath expressions), aggregate results within one webpage, how to refresh the views, and how to replace user profiles, specifying user needs. The result of the demonstration will be demonstrated by a view on web content blocks aggregated in one HTML document within myPortal. The view will include only requested content blocks, arranged according to user's specifications.

To compare myPortal with other work, during the part of the demonstration led by the authors, we will demonstrate robustness of absolute XPath queries (and compare it with relative ones), and also relate to content syndication technologies like RSS or ATOM.

Throughout the demonstration the viewers will be able to experience the unique features of myPortal. Additionally, superiority of relative XPath expressions compared to absolute ones will be shown. The method will be discussed, including selection of relevant content blocks, automatic construction of XPath queries, portal configuration and evaluation of XPath expressions.

5. CONCLUSIONS

MyPortal demonstrates both ease of use (simple point & click interface) and improved robustness of web content extraction and aggregation. This was obtained thanks to non-trivial use of relative XPath queries. The usage of relative XPath expressions shows serious advancements in robustness to content structure changes in comparison to previous works in this area focusing on absolute XPath queries. The method used in myPortal can also be easily used to populate databases with preprocessed information extracted from the Web.

6. REFERENCES

[1] Abe, M. and Hori, M. Robust Pointing by XPath Language: Authoring Support and Empirical Evaluation. in *Proceedings of 2003 Symposium on Applications and the Internet (SAINT 2003), 27-31 January 2003*, IEEE Computer Society, Orlando, FL, USA, 2003, 156-165.

[2] Berglund, A., Boag, S., Chamberlin, D., Fernández, M.F., Kay, M., Robie, J. and Siméon, J. XML Path Language (XPath) Version 2.0 - W3C Candidate Recommendation, World Wide Web Consortium (W3C), 2005.

[3] Cai, D., Yu, S., Wen, J.-R. and Ma, W.-Y. Extracting Content Structure for Web Pages Based on Visual Representation. in Zhou, X., Zhang, Y. and Orlowska, M.E. eds. *Web Technologies and Applications: 5th Asia-Pacific Web Conference, APWeb 2003, Xian, China, April 23-25, 2003. Proceedings*, Springer, 2003, 406-417.

[4] Clark, J. and DeRose, S. XML Path Language (XPath) Version 1.0 - W3C Recommendation, World Wide Web Consortium (W3C), 1999.

[5] Hua, Z., Xie, X., Liu, H., Lu, H. and Ma, W.-Y. MobiDNA: A Unified Framework for Browsing Dynamic Web Pages on Mobile Devices. in *WWW 2005*, 2005.

[6] Kowalkiewicz, M., Orlowska, M., Kaczmarek, T. and Abramowicz, W. Towards more personalized Web: Extraction and integration of dynamic content from the Web. in *Proceedings of the 8th Asia Pacific Web Conference APWeb 2006*, Harbin, China, 2006.

[7] Laender, A.H.F., Ribeiro-Neto, B.A., Silva, A.S.d. and Teixeira, J.S. A brief survey of web data extraction tools. *ACM SIGMOD Record*, *31* (2). 84-93.

[8] Raggett, D. Clean up your Web pages with HP's HTML tidy. in Philip H. Enslow, J. and Ellis, A. eds. *Proceedings of the seventh international conference on World Wide Web 7*, Elsevier Science Publishers B. V. Amsterdam, The Netherlands, Brisbane, Australia, 1998, 730 - 732.

[9] Song, R., Liu, H., Wen, J.-R. and Ma, W.-Y. Learning Block Importance Models for Web Pages. in *WWW 2004*, ACM Press, New York, USA, 2004, 203-211.