HTML to XML Conversion for Non-Programmers

Jure Mijić*, Marko Tadić†, Matija Jančec*, Goran Jovanov*
*Faculty of Electrical Engineering and Computing, University of Zagreb
Unska 3, 10000 Zagreb, Croatia
jure.mijic@fer.hr, matija.jancec@fer.hr, goran.jovanov@fer.hr

†Faculty of Philosophy, University of Zagreb
Ivana Lučića 3, 10000 Zagreb, Croatia
marko.tadic@ffzg.hr

Abstract. *Any type of processing of the increasing number of e-text documents appearing today, particularly on the Internet, requires their conversion to a standard format like XML since they usually appear in a variety of proprietary and public formats.*

We present our solution for a generic HTML to XML conversion. The conversion is done using simple rules specified by the user. By defining these rules the user can divide the document into logical divisions (i.e. heading, body, signature) and achieve the desired output document structure. Our solution requires no programming skills because the script for the conversion is built interactively through a graphical user interface (GUI) and is suitable for all types of users.

Keywords. HTML, XML, conversion tool.

1. Introduction

XML [1] has become widely popular format for encoding data in a large number of data-processing fields. Since its emergence, in the field of Natural Language Processing (NLP) in particular, it has become *de facto* standard where it is used for encoding primary data i.e. textual data from text collections or corpora as well as for secondary and/or derived data i.e. interpretations or models in the form of annotations superposed on and linked to primary data.

Today, primary textual data are rarely entered manually for research purposes — they are usually achievable in a far cheaper way since they already exist in e-text format i.e. they are already stored in some kind of digital form. The only serious obstacle to unlimited usage of e-text is the variety of textual formats in which e-texts can appear: from simple, unformatted (.txt), to more complex formats with some kind of formatting (.txt with formatting), to more complex publicly available formats (.html, .rtf, .tex), to proprietary formats such as (.doc), and to formats available from DTP sources (.qxd, .pm, .ind) etc. It is normally expected that processing routines and programs, which are preferably standardized, also require standardized, transparent and yet easily expandable format for structuring data. The problem of the abundance of different text formats can be solved by their conversion into the same format which would fulfill given conditions. For NLP purposes XML has been the format of choice.

The rationale behind this work was to offer a solution, even to non-programmers, which would allow easy conversion from HTML, as one of e-text formats which is widely available and to which other different e-text formats could be easily converted. The serious limitations of using HTML as a source document format for NLP purposes are its non-expandability and its function to encode the layout of the document and not its real data-structure, which disqualifies it for natural language data structuring and processing. Therefore we decided to leave the conversion from different e-text formats into HTML to already existing and well-documented programs and build our own tool for conversion from HTML (of any generation) to XML. This tool, named HTML2XML, uses HTML elements (i.e. their tags), attributes attached to them and values of these attributes and enables the user to build his/hers own scripts which anchor each command to selected HTML element(s), their attributes and their values in order to produce fully strucured output XML document.

The programe package was developed for the purposes of a wider project led by Croatian Information, Documentation and Referral

Agency (HIDRA) in cooperation with two faculties from the University of Zagreb: Faculty of Electrical Engineering and Computing and Faculty of Philosophy. The project AIDE (Automatic Indexing of Documets with Eurovoc) aimed to develop the "indexing workstation" which would facilitate and speed up content indexing (i.e. descriptor attachment) of official documentation of the Republic of Croatia. The conversion of these documents to XML was considered as the first step in the sequence of different document processing steps.

After this introduction, in the paper the overview and problems of existing solutions to the conversion from HTML to XML is given, followed by our solution the program description. At the end, the conclusion tries to open new directions for similar software development.

2. Problem and current solutions

There are many tools for HTML to XML conversion (e.g. [2], [4], [5]) either public or commercially oriented. Most of them use a constant script that is integrated in the code of the program and therefore are unsuitable for processing many documents with variable structures. Others only have the ability to extract information from the document partially. In their cases, some data which could be considered useful in further processing are being discarded. This goes not only for the content of HTML elements but also for the values of HTML attributes.

Some tools (e.g. [6]) also use elaborate ontology-based procedures which are in fact too complex for general usage. The main advantage of this approach in HTML to XML conversion is in its usage of RDF based ontology for guiding the agents (i.e. crawlers) through web-pages and helping it "decide" whether to collect a certain HTML document from the site or not and from which URL. For each site (or set of similar web-pages) a manually defined RDF ontology has to be formed in advance. This approach shows its advantage when used on regularly basis, particularly with procedures which include "site-scanning" for new documents.

Our approach aimed to be more simple and applicable to HTML documents already existing on local disks leaving that type of (sometimes quite complex) decisions either to humans or other programs. We believe that ontology based approach requires more profound knowledge of HTML document source analysis than simple recognition of HTML tags and attribute values needed for this program by general user.

As far as we know, there is no program package which would offer a generic solution in such a way and thus enable conversion of many documents with variable structures. Even more, there is no package which would allow non-programmers to develop their own, customized solutions for HTML to XML conversion and all this in a practical, user-friendly graphical interface. This is exactly what we had in mind while building this HTML2XML converter on the experience of its prototype predecessor 2XML built back in 1999 for the purposes of documents conversion for the Croatian National Corpus [7].

3. Our solution

Our solution enables the construction of different user-defined scripts based on the elements and attributes of the input HTML document. With the ability to change the script, different structures of the output XML document can be defined and produced. Also input HTML documents of different structures can have the same or similar structure as the output XML document by applying different scripts to different types of input HTML documents.

The script is being developed interactively and no programming skills are necessary. Construction of the script is done in two steps.

The first step is preprocessing the input HTML document and gathering the information about all the elements, attributes and detected values that exist in the document. The default results of the first step are displayed in two parallel browser windows (see Fig. 1) and the user uses them to recognize and define the parameters of commands of the script.

The second step consists of applying the developed script to HTML elements and corresponding attributes with their values in order to achieve structured XML document at the output. Checking the output XML is required for fine-tuning the script until the desired output is achieved.

3.1. Scripts and their commands

A command of the script is defined by input and output parameters. The input parameters are selected from the results of preprocessing. They consist of names of HTML tags and names of HTML attributes with their values: classes, font

types, font sizes, font colors, align parameters and table levels found in the input HTML document. Additional input parameters can be specified, like: process only first/last *n* elements or the parameter 'containing text' that defines that input HTML element must contain certain string. The output parameters are defined by the user with commands in scripts. Output parameters are defined by the name of the output XML tag and the value of XML attribute 'type' of the output tag. There is also a parameter which omits the output element as well as a parameter which opens up to six division tags (DIV0-DIV5) and their attributes 'type'. Division tags are used to group output elements into logical divisions and thus structure the output XML document. If both of the parameters: name of the output tag and the option to delete the element, are not defined, than that command is declared as an 'empty command' because the output conversion is not specified.

3.2. Applying the scripts

A command is applied on the input HTML element that has the same parameters as those defined as the input parameters of that command. In that way each command in the script is used as a filter for accessing elements in input HTML document. If some of the input parameters of the command are not specified, then that parameters of the input element are not taken into account. Input parameters that are defined in the command must match the parameters from the input element in order to apply that command to that element. Two or more similar commands can be defined, but if one of them is more specialized (has more input parameters defined) and they both can be applied to the input element, then the more specialized command is applied. If both of the commands have the same number of input parameters defined, then the command that is first in order is applied. Also, the option to delete the element can be specified in the command, in which case the input element is discarded. If none of the defined commands can be applied to an input element, the default script is applied. It converts input elements with the name of HTML tag, which represent the formatting tags and tables to the output elements of the same name and all others to output elements with the tag 'p'.

The division tags serve to divide the document into a logical division. Opening new division tags implies closing previous opened divisions to that of the lowest division selected in the command.

In the process of developing the script, the commands defined so far can be tested by applying the script to the document in order to check if that command(s) produce the desired result.

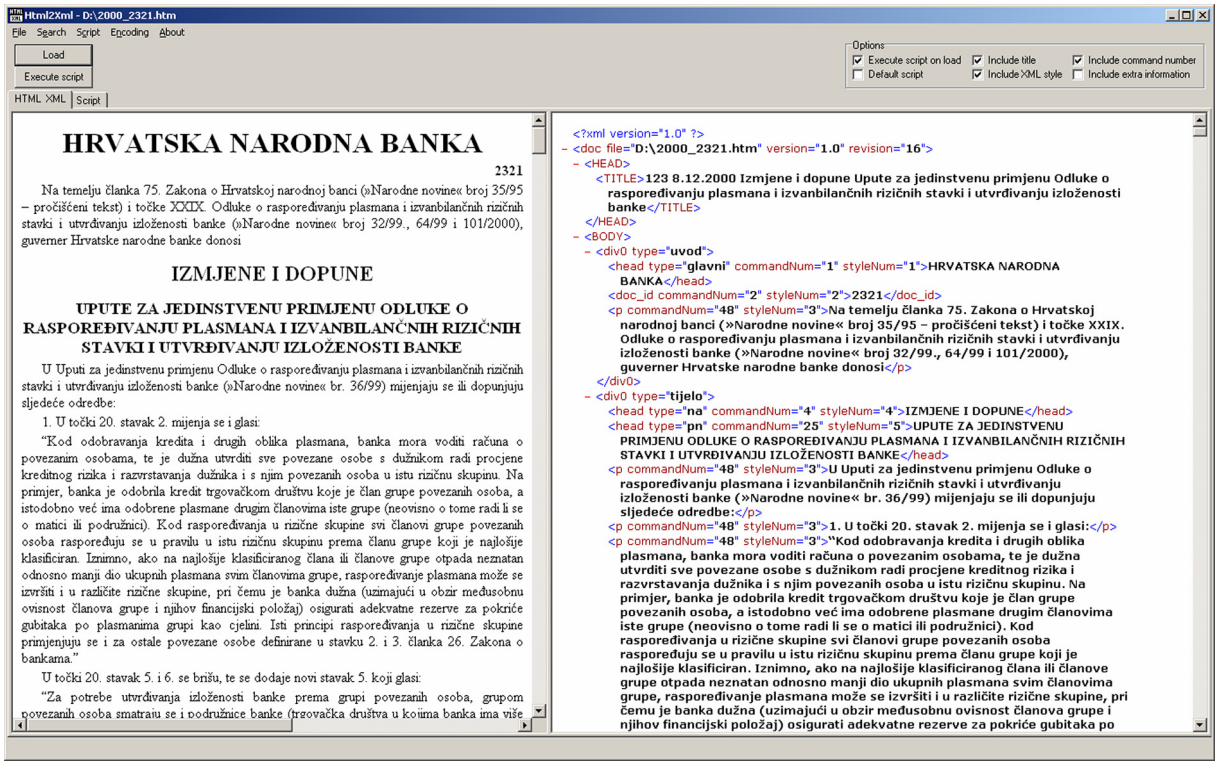

Figure 1. The first tab window with two browsers and the first step in HTML to XML conversion

4. The program

For the HTML and XML parsing we used 'libxml2' toolkit [3].

The interface of the program consists of two main tab windows and additional options. The first tab window ('HTML XML') is used to display the input HTML and the output XML document in two parallel browser windows, thus allowing the user to compare the input and output documents side by side (see Fig. 1). The second tab window ('Script') displays the script, command options and commands for the script construction (see Fig. 2).

Selecting the command from the script box fills the command options on the right hand side with the values defined in selected command. Command options are initially empty if the input HTML document has not been loaded. While loading the input document, the program automatically performs a preparsing of the document and displays all possible detected elements, attributes and their values in the command options. Construction of the script is done by adding commands and setting the command options for each command. Command is displayed as 'Empty command' if both of the following two output parameters are not specified: name of the output tag and option 'should be deleted'.

4.1. The input parameters

The command can be set up to be applied to all elements or the first/last *n* elements. The results from preparsing the document are displayed in options: fontFamily, fontSize, fontColor, align, htmlTag, classTag, tableAlign and tableLevel. These options can also be entered manually. Additional input parameters 'paracount' and 'containing text' are defined by the user.

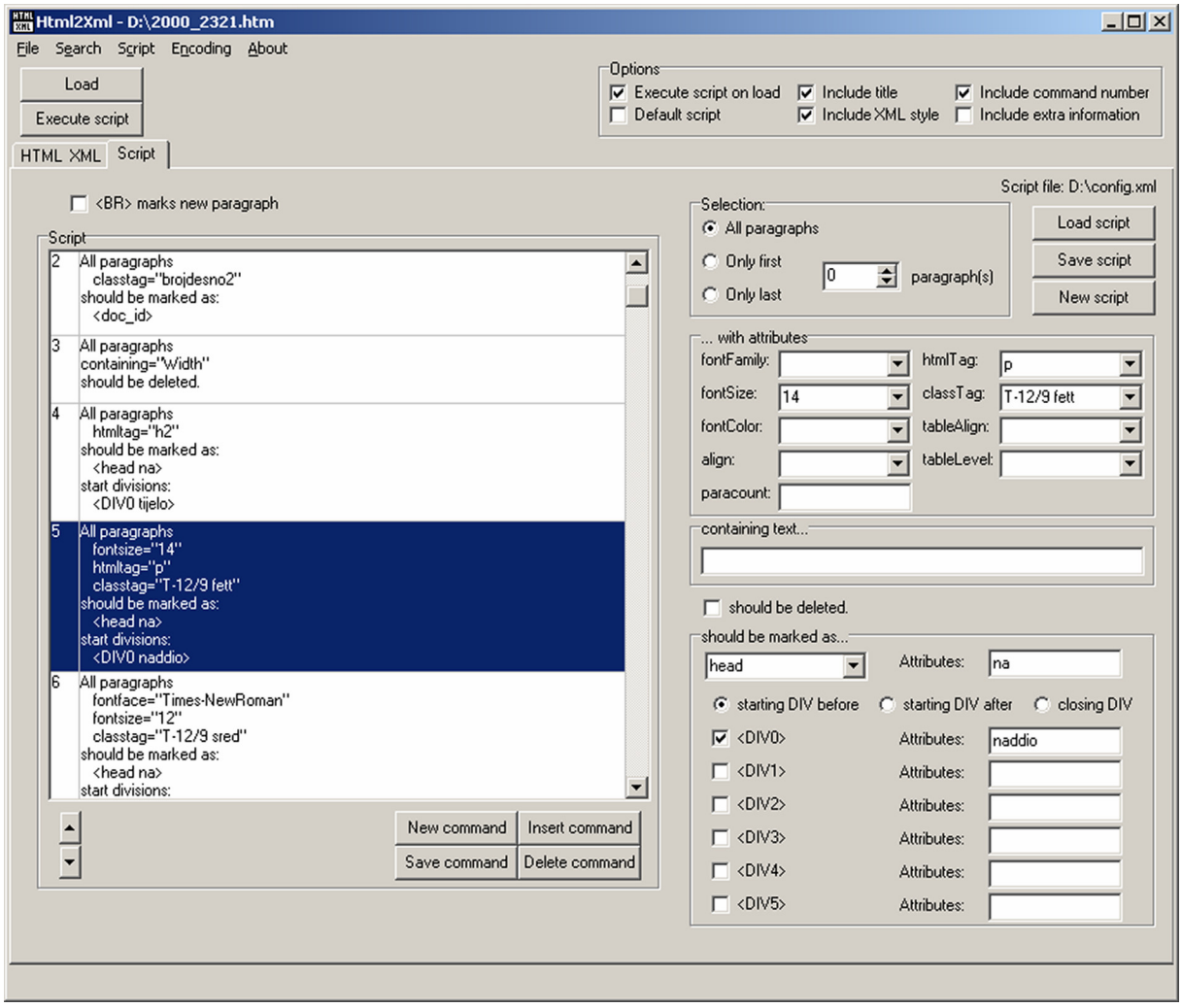

Figure 2. The second tab window with script and its command input and output parameters

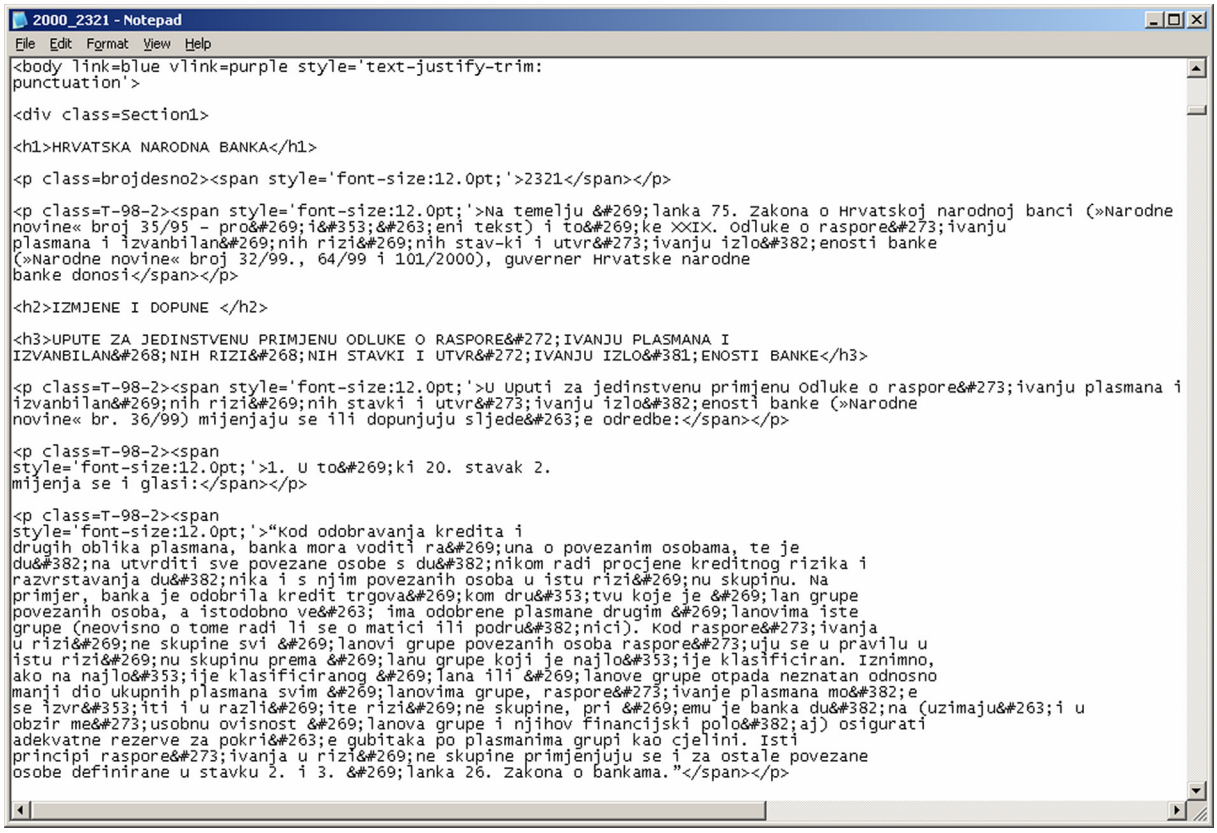

Figure 3. Example of the source of the input HTML document

4.2. The output parameters

Multiple divisions can be selected and for each of them their attribute ‘type’ can be defined. Additional options for division manipulation are added. ‘Starting DIV before’ opens structural divisions of the output XML document before opening the converted element. ‘Starting DIV after’ opens divisions after closing the converted element. ‘Closing DIV’ closes divisions to the lowest selected division level before opening the converted element.

Once defined, the scripts could easily be saved into files and also loaded for processing. The very scripts are also structured as XML documents (see Fig. 4) so they can be easily accessed from outside of this program package and manipulated/generated/modified by other simple XML-processing programs.

4.3. The program options

The control block of switches in the upper right corner of the Script tab window define the values of output XML attributes more precisely. The original HTML title of the document can be transferred to the output XML document or omitted. Also, the HTML style attributes can be transferred or omitted as well as extra information regarding the HTML formatting attributes such as ‘align’ etc. The switch which inserts into the output XML element the attribute with the number of command with which that element was produced is a valuable tool for developing and debugging scripts because it allows the user to backtrack through generated XML elements and fine-tune the command parameters to achieve the desired output structure.

5. Usage and evaluation

The program was used within the project AIDE (Automatic Indexing of Documents by Eurovoc) led by HIDRA (Croatian Information, Documentation and Referral Agency) where it was applied to several thousands of HTML documents from Narodne novine (Official Gazette of the Republic of Croatia). The script used for that conversion was developed within a few hours. The whole batch conversion (2,488 documents, 56 Kb average size) was finished in 10 minutes on moderately equipped PC (P4 2.4 GHz, 512 Mb RAM, 80 Gb HD). This result proves to be acceptable for most types of users,

measured in both the time and computational resources.

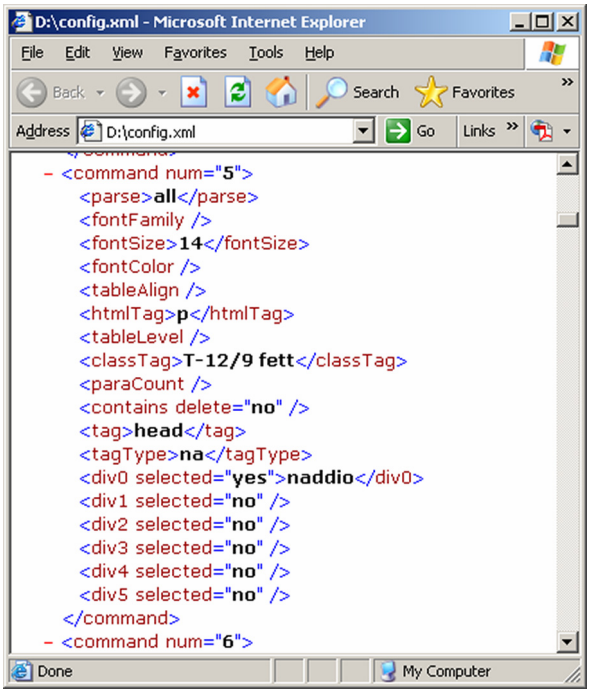

Figure 4. Internal XML format of scripts

6. Conclusion

We have shown a simple, generic solution for HTML to XML conversion suitable for custom made conversions even for non-programmers. Our programe package features easily definiable scripts, easy-to-use graphical interface and tools for debugging such as command line numbers in the output XML elements.

Further directions for this type of generic purpose converters to XML could be the building of the similar converters for RTF to XML conversion or even PDF to XML conversion. This platform also enables the development of other applications such as the tool which could compare a new, unseen HTML document with the one(s) for which we already have scripts developed, in order to automatically detect in advance which of the existing scripts would be the most appropriate for its conversion to XML. With such an application, higher level of automation (i.e. batch processing) could be achieved.

7. Acknowledgements

We would like to thank another team from Faculty of Electrical Engineering and Computing and the team from HIDRA for outstanding work on our joint project *Text mining system (2003-082)* funded by the Ministry of Science, Education and Sports of the Republic of Croatia.

8. References

[1] Bray, Tim; Paoli, Jean; Sperberg-McQueen, C.M.; Maler, Eve; Yergeau, François, editors. Extensible Markup Language (XML) 1.0 (3rd Edition). World Wide Web Consortium, 2004. http://www.w3.org/TR/2004/REC-xml-20040204 [02/21/2005]

[2] HTML Tidy. Sourceforge.net, 2000. http://tidy.sourceforge.net/ [02/24/2005]

[3] Libxml2 toolkit. Gnome project, 2004. http://www.xmlsoft.org/ [02/24/2005]

[4] Naper HTML to XML Conversion Utility. Naper Solutions, 2003. http://www.napersolutions.com/htmltoxml.html [02/24/2005]

[5] Potok, Thomas; Elmore, Mark; Reed, Joel; Samatova, Nagiza. An Ontology-based HTML to XML Conversion Using Intelligent Agents. Proceedings of the 35th Hawaii International Conference on System Sciences, 2002.

[6] Stylus Studio HTML to XML Importer. Stylus Studio, 2004. http://www.stylusstudio.com/html_to_xml.html [02/24/2005]

[7] Tadić, Marko. Building the Croatian National Corpus. LREC2002 Proceedings, Las Palmas-Paris, 2002, Vol II.