

Flexibility And Adequacy Of The Output 's Layout of the Content Displayed In The Hyper-automaton System

César C. Machado, Gustavo L. Federizzi, Paulo B. Menezes

Computer Science Department

Universidade Federal do Rio Grande do Sul - UFRGS

Porto Alegre, RS, Brasil

and

Centro Federal de Educação Tecnológica de Pelotas - CEFET-RS

Pelotas, RS, Brasil

Abstract *This work continues former studies reaching for a more intelligent and flexible way of displaying material from courses for the Internet that uses the formal constructions navigating system known as Deterministic Finite Automata with Output. The study based on the technologies evolution of the Internet suggests that the instructional material well structured in XML, used by today's systems, incorporates the benefits of the usage of cascade style sheets and advanced style sheet systems, ruled by important criteria of modern styling and formatting of the visually available content, along with the necessities and technological benefits of XML. The main goal of this work is to schematize and describe the rules according to a bibliographically supported and applied view. Such rules, which in a general meaning are optionally part of a XML document, in the Hyper-Automaton System, which we developed, will be undoubtedly indispensable and permanently integrating elements. This work continues the Web Courses Modeling Using Formal Systems work and has as goal to provide alternatives regarding the formatting of the user interface using XML and its applications.*

Keywords: WWW, XML, XSL, Finite Automata, Hyper-documents Management, Web courses.

1 Introduction

This article continues the "Modeling Web Courses using Formal Systems" [1], [2] work, and it has as objective identify alternatives for user interface formatting by use of XML and applications. Web courses using formal systems uses formal constructions known as Deterministic Finite Automaton [3]. The group introduced the concept that "courses are automata" as a structure that requires little effort for implementation, creation of hypermedia material that is independent from the automata, and encourages the reuse of web pages in several courses, which can be of different focus, diminishing redundancy in page creation [1], [3], [4].

The general goal of the hyper-automaton model centers on the study of the application of the Finite automaton with output formalism (Mealy and Moore machine) as a structured model for organization of instructional hyper-documents, especially Web courses. The model was inspired by classical research in the hyper-documents area and in recent initiatives in the World Wide Web, mainly in the development of hypertext systems that the base of hyper-documents is independent of the hypermedia application's structure, and that supports some conveniences as hierarchal struc-

ture composition, the specification of several links for the same set of hyper-documents and objects separated from the navigational structure.

Each automaton defines a course, and consists of a set of independent hyper-documents, which can be used in other courses. The transition function works as a logic bind between the hyper-documents and the output function composes the web pages. The final result is a basic structure of pages and links in a web site. This model results in a high level of instructional material modularization, with the following advantages: ease to reuse pages in different web courses diminishing redundancy; hyper-documents independent of the navigation structure - update content without worrying about navigation structure and vice-versa; enable users to create links from any document to any document; ease of implementation and maintenance; customizable graphic interface, elaboration of instructional sequences with specifics focus and capable of customizing the learning process; categorical operations enables the construction of new courses based on existing courses by means of high level procedures [1], [4].

The hyper-automaton system is a semi-automatic system for supporting Web Courses (based on and client/served architecture and html developed interface) by means of concepts that are inherent of Computer Science, mainly of the Automaton Theory, Hypermedia technology and Categorical Theory, exploiting the advantages of all. Although the focus used validating the model was the implementation of a distance learning system, the results can be extended for any hypertext system.

Our work in the current stage is centered in the several alternatives of displaying content, working on an adequate user interface, where the customization of formatting instructional material in the output function of the automaton will be an application that mounts the document with the chosen parts with the most adequate characteristics related to the user's and course's needs.

The study being made will bring more flex-

ibility, since in the current state the only possibility is a rigid concatenation of web documents. By considerations that exclude any commercial reason, the work developed was tested with Internet Explorer 5.5, being the only web browser that supports standard XML.

2 A vision of the system adequate to styles

The diagram detailed in figure 1 illustrates the system's parts and gives us a good of what is being developed, working on a adequate, right and standard way of displaying well structured XML documents, well formed and valid against a DTD [5], [6].

The study of rules for the Hyper-Automaton system was discussed in former studies. These changes has as main objective the increase of flexibility of the output function by the application of multiple styles on the material being displayed, as well as a technological update of the system according to the benefits of XML described in former studies [9], [10].

By the use of XSL templates and/or cascade style sheets (CSS) [11], [8], we will obtain visualization format and the layout of the web courses.

Now will be describe one of the most important features of XML related to the study of the possible ways of visualization of previously structured content obeying strongly the rules of the standard described by the World Wide Web Consortium (W3C).

Known as style sheets, theses applications are strongly linked to our goals, since they establish ways of manipulating documents, simply inserting direct formatting criteria or a complex transformation of the XML document's tree [13].

The advanced style sheet make possible several ways of displaying the same material, by the use of adequate formatting commands and XPath expressions, resulting in a adequate visualization of instructional content in web browsers that supports XML documents and

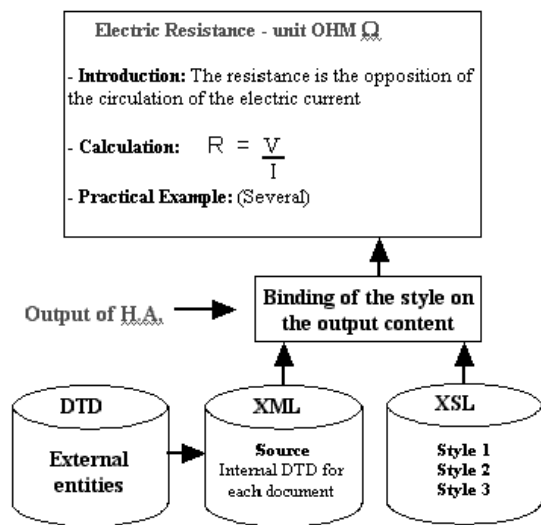


Figure 1: Dynamic Binding of Style Sheets

others used on the web.

Detailed in the figure above, these characteristics certainly will be a part and widely used in the new proposal of the Hyper-Automaton system.

3 Cascading Style Sheets

Represent one of the most interesting new features made in HTML 4.0 and the web. It was introduced by W3C to incorporate consistency to the web, since it promises a great improvement to the current state of art, somehow chaotic [11]. The Style Sheets enables the authors to specify layout and projects elements of a whole web site, such as fonts, colors, indentation, and a precise control of the way the elements are displayed in a web page.

The most interesting and important is that neither badly configured web browsers nor users' whims will spoil the exhibition of web documents that depends on styles.

In the bibliography [11], style sheets pertain in the advanced markup, since they are very complex. A style sheet defines the project and layout information. In general the style sheets also specifies fonts, colors, indentation, kern-

ing, leading, margins and even the page dimensions of any page that cast them.

The consistency is a highly desirable feature in the final product, as much for printed documents as for electronic documents (on-line). The cascade style sheet is one of the most important features of the system being proposed, since it is an indispensable element on the trinomial content, rules and display.

The cascade style sheet is a version of the CSS standard, known as CSS1, but so complete that the future versions will tend to have only small changes [11].

One of the most fundamental features of the CSS1 is that it supposes that several related style sheets can be cascaded, meaning that the authors can attach their favorite style sheet to the web documents, and the readers can attach their own style sheets to the same documents. This enables correction of human and technological deficiencies, like adjustments for printing the document and screen resolution.

Basically, the CSS has a set of rules that resolves conflicting style problems that appears when several style sheets are applied to the same document. Since the conflicts can arise, some resolving method is essential to a adequate display of document content to the user.

In order to prevent that the style sheets some important personal definitions, especially with people with sight deficiencies, the web browser only resolves the conflicts by priority after all the style sheets are loaded to the memory, applying the selected styles and ignoring the others.

A document that attaches only cascading style sheets have an important disadvantage when it comes to flexibility of output format of XML documents. The XML document is scanned and it is shown to the user exactly how the document was constructed. In other words, it is not allowed to transform the tree of the original XML document. Figure 2 illustrates the use of cascade style sheets with an XML document.



Figure 2: Use of CSS

4 XSL

XSL stands for eXtensible Stylesheet Language, and in its early days it was used to describe its specifications. Some features began to become very complex and at the moment standard XSL is a collection of three standards: XSL, XSLT and XPath. The rules of these standards can be described independently. It is possible to describe and exemplify the way that these three standards work together to format an XML document. The original term XSL now is used in a more restricted way, describing how to specify output styles. The XSL standard defines an application of XML that breaks the rules of XML structured documents: an element is used to describe the look of the content, not its meaning [14].

The XSLT (XSL Transformation) is an XML Language for transformation. Using XSLT is possible to convert XML in another data format, including several formatting languages that use rendering controls.

The XSLT standard specifies the documents style sheet format that includes rule-mapping definitions. An XSLT document is an XML document that conforms to an XSLT DTD [14]. At the present time the more frequent use of XSLT is to convert XML documents to other formats that are recognized by web browsers.

XSLT is used for several purposes in addition to formatting. XML documents can simply be transformed in another XML doc-

ument with different structures according to another DTD. Doing this, the elements can be reused, relocated, chosen, separated, joined with other content and transformed according to attributes or values.

Several benefits are linked to the fact that XSLT style sheets are XML documents. For starters, it is possible to use a DTD to validate an XSLT document. Second, style sheets can be stored in a repository, where they can be manipulated and identified by version. Third, style sheets can be constructed to format other XML style sheets for presentation or print.

XPath is used in the transformation of an XML document in another XML document. In a simple way, by the use of techniques shown below, tags can be renamed according to the definition included in the DTD. For example, a Paragraph element can be changed to the Para element or yet the P element.

XSLT has other important characteristics that enable content to be move, copied, and specifies formatting that can be applied to the elements when they appear on a specific place in the structure of the document. This advanced type of analysis and data manipulation is executed by the use of an expression language.

The original proposal of XSL included this expression language, but the XSLT standard references other standard called XPath. This standard defines a general proposal to interrogate the XML document structures. It is also used as a query language and for binding advanced schemas of hyperlinks. This language is not based on XML, but has a compact syntax that has been developed for a great number of circumstances. The following expression identifies the paragraph element that has the type attribute set as important in a chapter element:

```
chapter // paragraph [@type = important]
```

The language expressions give us powerful ways to interrogate XML structures. So far we can observe that XSLT, with the use of XPath, offers an ideal way to transform an XML document to another XML document, ready to be formatted for presentation, see illustration 3.

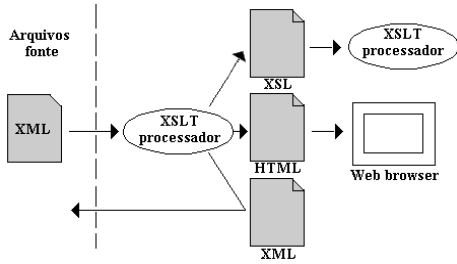


Figure 3: Visualization process

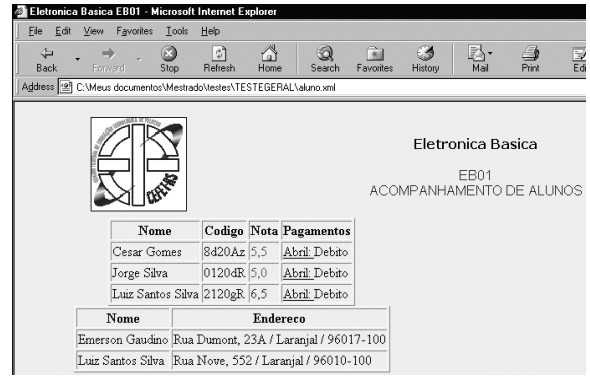


Figure 5: Output screen of the example.

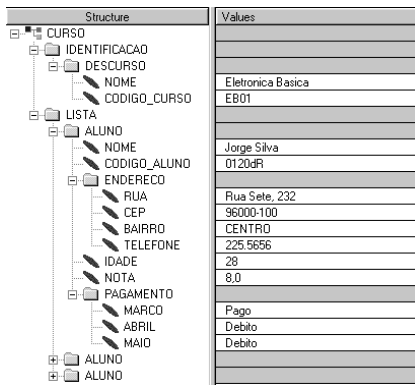


Figure 4: XML document structure

5 Language Characteristics and practical examples

The structure of the XML document's font, exemplified below, identifies a short document with information correctly structured of an on-line course. We can see that when the list of the XSL application is accessed according to the reference on the XML document. The application is responsible for the use of the style sheet, since it has the XPath expressions, searching in the file's tree displaying in the user's screen data according to the style sheet being used.

Below we can observe an associated style (shown in figure 4) that generates a specific output for the XML document example. The XSL style sheet, with XPath expressions attached to the XML document, enables the visualization of the chosen and formatted data. This style sheet is capable of searching the

XML structure displaying the students name and their code, which are in debit in April, and their final grade, displayed in red if not satisfactory. In the same page, there is a table that lists the students that resides in the Laranjal County.

It is important to notice that we tried to make the output's formatting as automatic as possible, were the HTML tags inserted are only used for display purposes to the users screen.

One of the greatest advantages of XML is the powerful way of identifying piece of information, which makes the formatting easy when applying advanced style sheets with XPath expressions [11], [6], [12].

Finally, in figure 5 we can see the output of the XML document combined with the XSL style sheet in the web browser.

It is not the objective of this work the explanation of XSL syntax; therefore we will examine a little fragment of XSL code that facilitates the output of the screen seen on figure 6. The most important in this example, which clearly shows the power of XML document manipulation, is the capability to order a list of names alphabetically, obeying the restriction of being in debit in April, and displaying the final grade in red if not satisfactory (less than 6,0). The code shown in figure 6 is responsible for this manipulation.

```

<xsl:for-each select="CURSO/LISTA/ALUNO" order-by="> MORE">
<xsl:if match=".[PAGAMENTO/ABRIL='3abico']">
  <tr>
    <td><xsl:value-of select="NOME"/></td>
    <td><xsl:value-of select="CODIGO_ALUNO"/></td>
    <xsl:choose>
      <xsl:when match=".[NOTE lit: '6']">
        <td><font color="red"><xsl:value-of select="NOTA"/></font></td>
      </xsl:when>
      <xsl:otherwise>
        <td><font color="blue"><xsl:value-of select="NOTA"/></font></td>
      </xsl:otherwise>
    </xsl:choose>
    <td><xsl:value-of select="PAGAMENTO/ABRIL"/></td>
  </tr>
</xsl:if>
</xsl:for-each>

```

Figure 6: XSL source code example

6 Conclusions

This work tried to demonstrate practical examples that will be adequately reformulated according to our needs throughout the construction of our applications, and studies for new formatting capabilities and XML structuring.

It was noticed that by use of XML and XSL is possible to make flexible the manipulation of XML documents, more specifically in the display of relevant information to the user.

More exploration with on features of XML, XSL, XPath need to be done to make the structured manipulation of information easier and easier, since our goal is centered on offering multiple styles dynamically with the output function of the Hyper-automaton, shown in figure 1.

Problems still remain, such as the inclusion of files that uses plug-ins, like VRML, which solution is still not certain. Another problem is the manipulation of images, since in spite of being recognized by the XML file, they can't be used with value-of select, so the HTML way of showing images remains [14].

In spite of the fact that XSL is a XML application being widely developed, much is to be made to make the use of HTML templates easier for the generation of information processed by the current browsers.

References

- [1] Machado, Júlio Henrique Araújo Pereira. Hyper-Automaton: Hipertextos e Cursos na Web Utilizando Autômatos Finitos com Saída. Porto Alegre: CPGCC da UFRGS. 2000. (Dissertação de mestrado).[2]
- [2] Menezes, Paulo Blauth, Machado, Júlio Henrique Araújo Pereira. Web courses are automata: a categorial framework. In: II WORKSHOP DE MÉTODOS FORMAIS, 1999, Florianópolis. II Workshop On Formal Methods. Florianópolis: UFSC, Instituto de Informática da UFRGS, 1999. p.79-88.
- [3] Paulo F. B. Menezes. Linguagens Formais e Autômatos. Segunda Edição. Editora Sagra-Luzzatto. Brasil, 1998.
- [4] Júlio P. Machado, Paulo F. B. Menezes. Sistemas de Gerenciamento para o Ensino a Distância. In Proceedings of V Congresso Internacional de Educação a Distância. São Paulo - Brasil, 1998. <http://www.abed.org.br>
- [5] Light, Richard. Iniciando em XML. Makron Books do Brasil Editora Ltda. 1999.
- [6] Harold, Elliotte, R. XML Bible. IDG Books Worldwide. Ago. 1999.
- [7] Lie, Håkon W, Bos, Ha B. Cascading Style Sheets, level1 W3C Recommendation 17 Dec. 1996, revised 11 Jan. 1999. <http://www.w3.org/TR/REC-CSS1>
- [8] Bos, B; Lie, H.W.; Liley, C; Jacobs, I. Cascading Style Sheets, level 2 (CSS2) Specification. Maio 1998. <http://www.w3.org/TR/REC-CSS2>
- [9] Machado, César C.; Menezes, P.B. Características e vantagens na estruturação de documentos em XML na WWW para EAD In. I Workshop em Informática na Educação, Passo Fundo-RS, 2000

- [10] Machado, César C.; Machado, J.H.P.; Grandi, R.; Menezes, P.B.; Utilização do XML no Sistema Hyper-Automaton In. Internacional Symposium on Knowledge Management / Document Management - ISKM/DM, Curitiba-PR,2000
- [11] Lie, Håkon Wiun; Saarela, Janne. Multipurpose Web Publishing Using HTML, XML and CSS. Communications of the ACM. Outubro 1999/Vol.42, n°.8
- [12] Moulis, Natanya P., Kirk, Cheryl; XML Black Book. 1999
- [13] Adler, S; Berglund, A.; Extensible Stylesheet Language (XSL) Versão 1.0 - W3C. Março 2000.
- [14] Bradley, Neil; The XSL Companion. 2000.