

METHODOLOGY FOR DYNAMIC WEB PAGES ASSEMBLY

Óscar Narciso Mortágua Pereira

*IEETA – University of Aveiro, 3810-193 Aveiro, Portugal
oscar@ieeta.pt*

Joaquim de Sousa Pinto

*IEETA – University of Aveiro, 3810-193 Aveiro, Portugal
jsp@ieeta.pt*

António José Batel Anjo

*Dep. Mathematics – University of Aveiro, 3810-193 Aveiro, Portugal
batel@ua.pt*

ABSTRACT

Structure, presentation and content are the 3 main blocks in web pages. Although HTML is the main technology for web pages development, it doesn't provide mechanisms for a clear separation of these 3 blocks, which would be a great step towards the construction of flexible web pages. If one could define on the fly, which structure, presentation and content to use, web pages would reach their maximum flexibility. With this flexibility, each web page could reach specific goals in terms of users needs, namely, user disabilities (visual, hearing, motor, cognitive), user learning profile, etc. Here, it is presented a methodology to achieve the above mentioned goals.

KEYWORDS

HTML, CSS, dynamic web pages

1. INTRODUCTION

The goal of the methodology here presented is to explain how to build web pages with a clear separation of its structure, presentation and content. Meanwhile, some small steps will be taken to explain how important is the separation between the 3 blocks. HTML traditional use is mainly focused on information rendering, preventing the separation of structure, presentation and content. With the internet development, it soon became clear that great advantages should arise if structure, presentation and content were separated and treated as distinct blocks. The improvement achieved is noticed at the accessibility, maintenance and portability levels. Structure is mainly concerned with the content layout, presentation is mainly concerned with how the content is presented and the content is mainly concerned with the information to be presented to the user. The separation of the blocks meant that a change in one of these blocks shouldn't have any impact in the other blocks. This means that, for example, for the same content and structure, there could be several versions for the same document, each one with a specific presentation. Each presentation may aim at users with specific needs or constraints. In other situations such as e-learning, one may need the same structure and presentation for several different contents. Even in an e-learning environment, one may need for each target public, for each web page, a specific structure and a specific presentation, due to disabilities constraints, and a specific content, due to the student profile. Internet and HTML development, in association with the emersion of other technologies as Cascading Style Sheets, XML, Active Server Pages, etc., have provided several solutions for the announced challenge: structure, presentation and content separation.

The methodology is based on 3 different technologies, 2 of them standard: HTML and CSS. The non-standard can be any technology able to build full dynamic HTML pages. In our case, we will use ASP.NET from Microsoft.

2. STRATEGY

The methodology here proposed is based on a concept called 'cell'. A cell is any asset from an HTML document that has existence in terms of structure, presentation or content, or any combination of the 3 of them. One cell may contain none, one or more other cells. The fundamental structure of a cell consists of an HTML tag and a definition for its ID property. In our examples we will always use the tag <div>, which is a general purpose block-level tag. Figure 1 presents 6 basic cells, none of which with explicit structure, presentation or content. Each cell may be seen as an HTML translation of an XML tag. The translation is made by defining one <div> tag with its ID property set to the XML tag name, per XML element.

```
<div id="student1">
  <div id="studentNumber1"></div>
  <div id="course1">
    <div id="courseNumber1"></div>
    <div id="courseRef1"></div>
  </div>
  <div id="studentName1"></div>
</div>
```

Figure 1. Basic cells structure

3. STRUCTURE

Structure of HTML documents is usually defined through a set of HTML tags as <table>, <p>,
, etc. These tags may be used, but our alternative is based on the cell concept already presented. The structure definition is set through the use of CSS attached to the <div> tags, as shown in Figure 2 and Figure 3. Although CSS may be defined through either HTML tags, ID selectors, classes selectors or even through context selectors, in our examples we will only use class selectors.

```
.clsStudent { position: absolute; left: 300px; top: 200px }
.clsStudentNumber { position: absolute; left: 0px; top: 0px }
.clsStudentName { position: absolute; left: 100px; top: 0px }
.clsCourse { position: absolute; left: 0px; top: 20px }
.clsCourseNumber { position: absolute; left: 0px; top: 0px }
.clsCourseRef { position: absolute; left: 100px; top: 0px }
```

Figure 2. CSS for document structure

Figure 2 presents the CSS for each <div> tag, whose properties are related to the position of the tags. Each CSS must be attached to the corresponding <div> tag, either by setting the correspondent CLASS attribute, as shown in Figure 3, or by setting it at run time, as it will be described later.

```
<div id="student1" class="clsStudent">
  <div id="studentNumber1" class="clsStudentNumber"></div>
  <div id="studentName1" class="clsStudentName"></div>
  <div id="course1" class="clsCourse">
    <div id="courseNumber1" class="clsCourseNumber"></div>
    <div id="courseRef1" class="clsCourseRef"></div>
  </div>
</div>
```

Figure 3. Document with structure

4. PRESENTATION

The methodology used to define the document structure will also be used to define the document presentation. As several CSS may be attached to each HTML tag, this feature provides a way of organizing CSS in a way that conforms to our goal. So, presentation is defined by other groups of CSS as shown in Figure 4. In this case, we defined 2 more groups of CSS, the first aims at frame level and the second aims at font level.

```
.clsStudent { padding: 10px 10px 10px 10px; border: outset 2px #ff3388;
               background-color: #ee99cc; width: 240px; height: 45px }

.clsStudent { font-family:Verdana; font-style:normal;
               font-size:x-small; color:Navy; }
.clsStudentName { font-size:x-small; font-weight:bold;
                  text-decoration: underline }
```

Figure 4. CSS for document presentation

Figure 5 shows the rendering of the document in its final structure and presentation, with some added content.

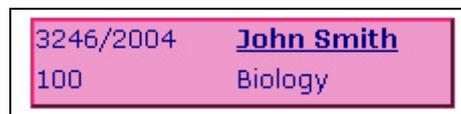


Figure 5. Document render

At this point, it would be possible to make changes at structure level without worrying about presentation or vice-versa.

5. CONTENT

Structure and presentation only require standard technologies, as HTML and CSS. For dynamic content definition, we propose the use of ASP.NET, but any other technology with similar features can be used. In ASP.NET, to be accessed as server-side objects, HTML tags must include the property `runat="server"`, as in `<div id="studentNumber1" class="studentNumber" runat="server">`. This property, in addition to the `id` property, allows the server to control any other properties associated with the tag, including the `innerText` property which, in the present case, can be used to define the content. In Figure 6 it is shown the server-side dynamic content binding. This code belongs to a C# code file, not to the HTML file. In ASP.NET, server-side code is not mixed with HTML. Gathering HTML and code from Figure 2, Figure 3, Figure 4, Figure 6 results in Figure 5.

```
studentNumber1.InnerText = "3246/2004";
studentName1.InnerText = "John Smith";
courseNumber1.InnerText = "100";
courseRef1.InnerText = "Biology";
```

Figure 6. Dynamic content attachment

6. FULL DYNAMIC

As the conclusion of this work, we extend the dynamic content binding methodology to the other 2 blocks: structure and presentation. All it is necessary to do, is to bind the CSS to the HTML tags through scripts, as it

has been done with the content, as it is shown in Figure 7. In this case, Figure 5 results from Figure 1 (runat = "server" must be set in all tags), Figure 2, Figure 4 and Figure 7.

```
studentNumber1.InnerText = "3246/2004";
studentName1.InnerText = "John Smith";
courseNumber1.InnerText = "100";
courseRef1.InnerText = "Biology";

student1.Attributes.Add( "class", "clsStudent" );
studentNumber1.Attributes.Add( "class", "clsStudentNumber" );
studentName1.Attributes.Add( "class", "clsStudentName" );
course1.Attributes.Add( "class", "clsCourse" );
courseNumber1.Attributes.Add( "class", "clsCourseNumber" );
courseRef1.Attributes.Add( "class", "clsCourseRef" );
```

Figure 7. Server-side code for full dynamic HTML.

7. CONCLUSION

The methodology here presented gathers 3 different technologies: HTML, CSS and ASP.NET (or other), and can be split in 2 steps: 1) structure and presentation - in this step , through <div> tags and CSS, one defines the possible structures and presentations for the document; CSS should be split in 2 main groups, one for structure and another for presentation, in order to implement a clear separation between them; 2) content and full dynamic - this step is accomplished through a specific tool, as ASP.NET. Although AP.NET is not a standard and open technology, the methodology here explained can be used with technologies with similar features. It is not so powerful, but one could even use client-side scripts such as Java Script to implement the methodology. Regardless the selected technology, structure, presentation and content can be built in a full dynamic way. This methodology may be seen as a good practice for generating any HTML documents, mainly those related with issues such as user accessibility needs, user profile, etc. This methodology has been used with great success in the development of a web based literacy application[1].

REFERENCES

- [1] Óscar Narciso Mortágua Pereira et al. "abcNet: a Literacy Tool in Developing Countries", In Proceedings of ICALT 2004, The 4th IEEE International Conference on Advanced Learning Technologies. IEEE Computer Society, Joensuu, Finland, 30 August - 1 September 2002, pag 948-952. ISBN: 0-7695-2181-9
- Ahmed, Mesbah et al. 2002. *ASP.NET Web Developers Guide*. Syngress Publishing, Inc., Rockland, MA, USA.
- Shafer, Dan. 2003. *HTML Utopia: Designing Without Tables Using CSS*. SitePoint Pty. Ltd., Australia.
- W3C – Cascading Style Sheets on line in 2004.09.16 at <http://www.w3.org/Style/CSS/>
- W3C – Web Accessibility Initiative on line in 2004.09.16 at <http://www.w3.org/WAI/>
- W3C – Extensible Markup Language on line in 2004.09.16 at <http://www.w3.org/XML/>