Translating TABLE Layout to Cascading Style Sheets

Andy Y. Mao, James R. Cordy and Thomas R. Dean
School of Computing
Queen’s University, Kingston, Ontario, Canada

I. Introduction

Motivation:
- Web applications age as new Web standards are introduced
- Web standards bring benefits to organizations, developers and users
- Manual migration of Web applications to new standards is expensive and undesirable

Vision:
- Automatic analysis and semi-automatic transformation
- Save humans from boring, uninteresting tasks

Key Ingredients:
- Automatic: normalization, TABLE partitioning, transformation, generation of CSS, clone detection and removal
- Human assistance: renaming of DIV tag names, modification of automatically generated CSS

II. The Web Standard

What is the Web standard?
- XHTML (XML + HTML)
- DIV + CSS

Why a Web standard?
- Improve user experience
- Speed up development
- Increase reusability and maintainability
- Shorten time to market
- Provide a foundation for Web 2.0

III. Implementation

Multi-pass Transformation
- Normalize HTML pages to XHTML format (TXL)
- Apply fixed widths to Table elements (TXL)
- Assume that the Table width is 800 pixels
- Partition Tables by rowspan to a set up Tables (TXL)
- Mark up Tables to transformed (TXL)
- Rename tag names to meaningful names (JSP & AJAX)
- Transform Tables to DIV and CSS (TXL)
- CSS Clone detection and removal (TXL)
- Modify and tune CSS to match original look (Manual)

Current Limitations
- Only deal with pure HTML Web pages
- Can’t deal with multiple rowspans at different levels

IV. TXL

Tree Transformation Language
- Hybrid functional and rule-based programming language

Example rule to allow “tag” markup on TABLEs:
```txl
define markup_start ['tag' id='stringlit'] { [HL]
end define
```
```
define markup_end ['tag'] { [HL]
end define
```
```
redefine html_table_tag [markup_start]
{ table
  [repeat html_my_tag parameter]
} [HL][IN]
[html_table_tag_close]
[HL][IN]
[markup_end]
end redefine
```

Example rule to remove “tbody” markup from TABLEs:
```txl
rule removeTbody
redefine $ [html_table_tag]
{ table
  [repeat html_my_tag parameter]
} [IN]
[html_table_tag_close]

ApFix [repeat html_table_content]

</tbody>
</table>
end rule
```

V. Example Model

Input: TABLE-based HTML pages
```
<table width="700" border="1" align="left">
  <tr width="100">
    <td width="300">
      content 1
    </td>
    <td width="300">
      content 2
    </td>
  </tr>
  <tr width="150">
    <td width="300">
      content 3
    </td>
  </tr>
</table>
```

Output: DIV-based XHTML pages ...
```
<div id="container">
  <div id="content">
    <div id="middle">
      content 1
    </div>
    <div id="right">
      content 2
    </div>
    <div id="container">
      <div id="content">
        <div id="middle">
          content 3
        </div>
      </div>
    </div>
  </div>
</div>
```

... and Cascading Style Sheet
```
#container { #middle {
    border:1px; display:inline;
    text-align:left; float:left;
    width:700px; width:300px;
  }
  #content {
    width:700px; #right {
      right {
        display:inline;
        float:right;
        display:inline;
        float:left;
        width:150px;
      }
    }
```