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 rowpsans at different levels

III. Conceptual Model

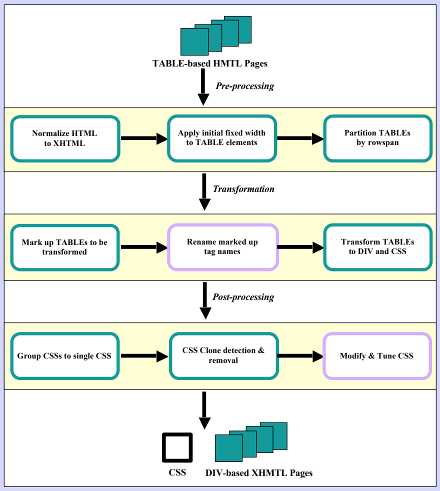

TABLE-based HMTL Pages
Pre-processing
Normalize HTML to XHTML
Apply initial fixed width to TABLE elements
Partition TABLEs by rowspan
Transformation
Mark up TABLEs to be transformed
Rename marked up tag names
Transform TABLEs to DIV and CSS
Post-processing
Group CSSs to single CSS
CSS Clone detection & removal
Modify & Tune CSS
CSS
DIV-based XHMTL Pages

IV. TXL

Tree Transformation Language

- Hybrid functional and rule-based programming language

- Example grammar to allow “tag” markup on TABLEs:

```
define markup_start
  [NL] '<tag 'id=[stringlit]'> [NL]
end define

define markup_end
  '</tag'> [NL]
end define

redefine html_table_tag
  [markup_start]
  '<table [repeat html_any_tag_parameter]'> [NL][IN]
     [repeat html_table_content]             [NL][EX]
  [html_table_tag_closing]                   [NL]
  [markup_end]
| ...
end redefine
```

- Example rule to remove “tbody” markup from TABLEs:

```
rule removeTbody
  replace $ [html_table_tag]
    <table TableParams [html_any_tag_parameter]>
      <tbody>
        RptTrs [repeat html_table_content]
      </tbody>
    </table>
  by
    <table TableParams>
      RptTrs
    </table>
end rule
```

V. Example Model

- **Input: TABLE-based HTML pages**

```
<table width=“700” border=“1” align=“left”>
  <tr width=“700”>
    <td width=“250”>
      content 1
    </td>
    <td width=“300”>
      content 2
    </td>
    <td width=“150”>
      content 3
    </td>
  </tr>
</table>
```

- **Output: DIV-based XHTML pages …**

```
<div id=“container”>
  <div id=“content”>
    <div id=“left”>
      content 1
    </div>
    <div id=“middle”>
      content 2
    </div>
    <div id=“right”>
      content 3
    </div>
    <br clear=“both”/>
  </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 {
}                             display:inline;
#left {                       float:left;
  display:inline;            width:150px;
  float:left;               }
  width:250px;
}
```

VI. Future Work

- Extend transformation system to Web applications using server side languages, e.g., ASP, JSP
- Extend system to handle more complex Table structures, such as multiple rowspans at different   levels
- Extend system to generate multiple CSS style sheets to support different browsers
- Improve user interface for renaming DIV tags

Presented at the
CSER Meeting Fall 2006
Toronto, October 15th