Purpose:

To develop a new section for the Guidelines on recommendations for encoding a variety of textual features related to text directionality and orientation.
Members

- Martin Holmes (TEI Council)
- Deborah W. Anderson (Unicode Consortium)
- Robert Whalen (Northern Michigan University)
- Marcus Bingenheimer (Temple University)
- Stella Dee (King's College, London)
Order of tasks

- Enumerate textual features to be covered
- Collate existing standards and recommendations and relate them to features
- Identify any gaps which might require new TEI elements or attributes
- Outline the new section(s)
- Write the first drafts for consideration by Council
- Identify other places in the Guidelines where information or links need to be included
Every which way: some examples

English is left to right, top to bottom.
Ishi no ue ni mo san nen

(Source: http://iroiro-ensyutu.at.webry.info/200606/article_4.html)
The Forum inscription: boustrophedon

- Lines 1, 3, 5, 7, 9, 11, 12, 14, 16: right-to-left, letters reversed
- Lines 2, 4, 6, 8, 10, 13, 15: left-to-right
- Lines 8, 9, 16: upside down

(Source: Wikipedia)
Ancient Berber

- Text in lines runs bottom-to-top
- Lines are arranged right-to-left

(Source: lbi-project.org)
Rongorongo

- “Reverse boustrophedon”
- Alternate lines are upside down
- Start at bottom left
- Read bottom-to-top
- Rotate 180° every line

(Source: Wikipedia)
Properties of a run of text

- Block progression, or block flow direction
- Inline progression, or inline base direction
- Glyph orientation, or line orientation

(Source: CSS Writing Modes)
“True” directionality includes all of the above.

But what about:

Source: http://www.ccel.org
Unicode and horizontal directionality

- All codepoints in the Unicode Character Database have intrinsic horizontal directionality (strong, weak or neutral).
- In mixed-mode text, the Unicode bidi algorithm (UAX #9) provides rules for working out which runs should go in which direction.
- The algorithm can be manipulated using 5 explicit codepoints to force directionality for a text run.
- Unicode 6.3 will add four new *bidi isolate* codepoints to this system.
- Most of this functionality, and its complexity, relate to mixed-mode horizontal text only.
Unicode and vertical directionality

- However, very little progress seems to have been made.
- UTR #50 addresses glyph orientation in vertical text.
- It proposes a new property called mvo (or vo), which gives a default glyph orientation when a horizontal glyph (or grapheme cluster) is part of a vertical text run.
- The orientation is defined relative to the glyph as it appears in the Unicode code charts.
- That's it for Unicode and vertical directionality.
Proposal #1: ignore Unicode

- UTR #20 *Unicode in XML and other Markup Languages* advises that bidi embedding (and presumably isolate) control characters NOT be used in XML markup.

- Inherent directionality from the UCD is an issue for text editing tools, not for markup (so Oxygen's worry, not ours).

- However, we need to warn people about possible interactions between bidi characters and markup.

- Should we also remove those characters during transformations?
CSS Writing Modes module

- Currently in draft mode
- Support for ltr, rtl, mixed-mode, and vertical (e.g. Asian)
- *No support for bottom-to-top* (I don't yet understand why)
- Relies on UAX #9 (Unicode bidi algorithm) for handling mixed-mode text.
- Draft proposes five properties:
<table>
<thead>
<tr>
<th>Property</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>writing-mode</td>
<td>horizontal-tb, vertical-rl, vertical-lr</td>
</tr>
<tr>
<td>direction</td>
<td>ltr, rtl</td>
</tr>
<tr>
<td>text-orientation</td>
<td>mixed, upright, sideways-right, sideways-left, sideways, sideways-right, use-glyph-orientation</td>
</tr>
<tr>
<td>text-combine-horizontal</td>
<td>none, all, [ digits &lt;integer&gt;? ]</td>
</tr>
<tr>
<td>unicode-bidi</td>
<td>normal, embed, isolate, bidi-override, isolate-override, plaintext</td>
</tr>
</tbody>
</table>
## CSS Writing Modes properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Values</th>
</tr>
</thead>
</table>
| `writing-mode`          | `horizontal-tb`  
                          | `vertical-rl`       
                          | `vertical-lr`       |
| `direction`             | `ltr`   
                          | `rtl`               |
| `text-orientation`      | `mixed` | `upright` | `sideways-right` | `sideways-left` | `sideways` | `use-glyph-orientation` |
| `text-combine-horizontal` | `none` | `all` | `[ digits <integer>? ]` |
| `unicode-bidi`          | `normal` | `embed` | `isolate` | `bidi-override` | `isolate-override` | `plaintext` |
Proposal #2: adopt/recommend CSS Writing Modes

- It looks well-organized and integrates well with Unicode and the UAX #9 algorithm.
- It covers most of what we need.
- We have @style where it can be inserted already (so we have in a sense already adopted it).
- No bottom-to-top, though.
My head hurts

- This stuff is really hard, and not necessary for many simple contexts:
  - Easter Wings
  - marcus
  - TEI-C.ORG

- We can cover a lot of simple requirements with better rotation features.
Proposal #3:

- @rotate-x (rotation on x axis)
- @rotate-y (rotation on y axis)
- @rotate-z (rotation on z axis)

This would handle:
- boustrophedon (mirror writing; rotate-y="180")
- ΜΠΡΕΖ (rotate-x="180")
- (rotate-z="45")
Other issues

- We haven't dealt with writing along paths:
- SVG is probably better for this.

(Source: Wikipedia)
Summary of proposals

- Ignore Unicode directionality features for the moment, but explain why in the Guidelines.
- Adopt CSS Writing Modes, and provide examples of how to use it.
- Create @rotate-x, @rotate-y and @rotate-z to allow all manner of rotation.

Thorts?
Summary of proposals

- Ignore Unicode directionality features for the moment, but explain why in the Guidelines.
- Adopt CSS Writing Modes, and provide examples of how to use it.
- Create @rotate-x, @rotate-y and @rotate-z to allow all manner of rotation.

Thorts?