

Presenting in HTML

Erik Wilde
UC Berkeley

Philippe Cattin
ETH Zürich

ABSTRACT

The management and publishing of complex presentations is poorly supported by available presentation software. This makes it hard to publish usable and accessible presentation material, and to reuse that material for continuously evolving events. XSLidy provides a XSLT-based approach to generate presentations out of a mix of HTML and structural elements. Using XSLidy, the management and reuse of complex presentations becomes easier, and the results are more user-friendly in terms of usability and accessibility.

Categories and Subject Descriptors: I.7.4 [Document and Text Processing]: Electronic Publishing

General Terms: Documentation

1. INTRODUCTION

Presentation material is often prepared using commercial software packages and then “published” on the Web as PDF. Presentation software excels at providing visual effects, but provides little support for handling complex presentations. Because of the very nature of dedicated software, presentations prepared with these packages are not very usable and accessible for anybody not using that particular software.

2. STRUCTURING

One of the most important drawbacks of today’s presentation packages is that they fail to support the structuring capabilities which are required when creating complex presentations. As a result, users have to resort to makeshift ways of capturing this structure; these brittle structures often fail to adapt properly to the reuse of material in different contexts.

XSLidy manages presentations as one XML document. If required, users can use inclusion structures which access various physical documents, but the logical document which is processed by the XSLidy code is one XML document. Most of the presentation markup in XSLidy is XHTML, XSLidy adds a small number of elements for structures and additional functionality, which are then transformed into XHTML. In addition, XSLidy also attempts to preserve as much structural information as possible, for example using HTML’s document relationships with `link` elements.

3. HYPERMEDIA

Because of the XSLT processing step, XSLidy can support hyperlinks on all elements (as suggested for XHTML 2.0), so that elements which should be links simply get an additional `href` attribute. In addition to being more succinct than

using individual elements, this also allows for the dedicated styling of these links, image links can for example be styled using a CSS `a.img` selector. Additionally, links are classified whether they are intra-presentation or external links, and can then be visualized differently, if required.

One common property of presentations is that they include additional resources, e.g. images or examples. HTML does not support the concept of *transclusion* (seamless integration of resources from various locations). XSLidy provides as much transclusion support as possible, for example text transclusion: the transcluded fragment is included in a presentation, and its text is turned into a link to the source.

4. EXTENSIONS

XSLidy does not provide support for specific application areas. Thus, an extension mechanism is provided which can be used to plug extensions into XSLidy. Extensions can plug into XSLidy’s XSLT, and/or can have additional pre- or post-processing stages.

Formula \TeX allows for seamlessly integrating \TeX code into presentations. Formula \TeX consists of an XSLT that produces files containing the \TeX code for each formula, and a Perl script that subsequently converts the \TeX snippets into images. Scale and baseline information for each formula is stored in a separate file. These parameters are required to correctly position the formulas within the text flow. Since all sizes are given in `ex` units, the scale and baseline settings are invariant to dynamic font size changes during a presentation.

The following code shows how Formula \TeX is used:

```
<tex>$x=\frac{-b\pm\sqrt{b^2-4ac}}{2a}$</tex>
```

This code displays the quadratic equation: $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

By tightly controlling the image rendering, scaling, and positioning, Formula \TeX ’s results are of consistent high quality, independent from scaling issues, or the particular output device. Users working with complex formulas often have their formulas readily available in \TeX notation, which can then be reused in Formula \TeX .

5. CONCLUSIONS

XSLidy is useful for managing and publishing complex presentations, but there currently is no interface for creating XSLidy files. This means that users have to manually create XSLidy XML, which is a big obstacle. Working towards an interface for XSLidy and providing import capabilities (for example using the XML exports of dedicated presentation software) are the two possibilities we are investigating as a way to make XSLidy easier usable for presentation authors.