Firstly, I suggest that DOM manipulation be used to assign priority to paragraphs of text. I suggest that if a paragraph with significantly lower priority than other paragraphs is found, it be reordered towards the end of the page. These paragraphs could be identified by the link-text to standard text ratio, or by using some simple lexical analysis, such as counting the amount of prepositions or calculating the space to word ratio. Most lexical analysis techniques have the side-effect of making any process locale-specific, so if possible, it is to be avoided, or made optional.

Secondly, I suggest that a stylesheet be used to flatten and resize content, and to highlight links. At this point, any given web-page would be rendered similarly to Opera's small screen rendering[4], as shown in figure 5.

Figure 5 – Opera small screen rendering (Image from www.linux-user.de)

At this point, I suggest that paragraphs over a particular length be collapsed into an expandable link made up of keywords from the paragraph, or from a header, if such a header can be extracted from the main body of the page. For example, on a news site, news stories may be collapsed under an expandable link of their heading. To avoid becoming locale-specific when using keyword extraction, these words could be identified via length, or the link could be made from the first few words of the paragraph, followed by an ellipsis. When these links are activated, the full paragraph could be expanded underneath them. If the paragraph is already expanded, it would be collapsed. This could be achieved by modifying the CSS visibility property in the DOM tree and adding a new text block at the same level in the tree that contains the link. Alternatively, it could be done at the renderer level, without modifying the DOM tree, although this may prove to be too complex a modification to perform in this manner.

This is similar to The Gateway and Flip Zooming in that it puts the focus on a single part of the page, but differs in that it can be implemented in a generic way in any browser, using a combination of JavaScript and CSS. The combination of these methods should eliminate horizontal scrolling, minimise vertical scrolling and highlight the user's attention to the important parts of the page, without too heavily altering the content and running the risk of making any information inaccessible, or worse, inaccurate.

7. Conclusion

It has been shown that the only assured method to make general web content accessible on a device with a small display is to manually reformat said content. As portable devices start to converge and become more common, this may happen more often. An intermediary, automated method, however, is required if today's devices are to be used efficiently to browse the web. A number of methods for achieving this have been examined, with varied amounts of success. Perhaps the most promising techniques are those that provide an entirely new way of representing a document, such as The Gateway[8], or Flip Zooming[11], however, none of those covered in this document have been developed to an extent where someone might use them today.

The most popular techniques being used in browsers today involve styling (used in the Minimo[6] and Opera[4] browsers) and DOM modification, either at the client-side or by special proxies (used by the Opera[4] browser on mobile phones). As portable devices become cheaper and more commonplace, it is likely that more and more websites will provide special stylesheets for small screens, or even recreate their content specifically for small devices (via methods such as news feeds or alternative locations).

As with any emerging technology, there will likely be many new developments in the years to come.

8. References

[1] Suhit Gupta, Gail Kaiser, David Neistadt and Peter Grimm, *DOM-based Content Extraction of HTML Documents*, Dept. of Comp. Sci., Columbia University. International World Wide Web Conference, 207-214 (2003).

[2] Tomonari Kamba, Shawn A. Elson, Terry Harpold, Tim Stamper and Piyawadee “Noi” Sukaviriya, *Using small screen space more efficiently*. Conference on Human Factors in Computing Systems, 383-390 (1996)

[3] Bonnie MacKay, Carolyn Watters and Jack Duffy, *Web Page Transformation when Switching Devices*, Faculty of Computer Science, Dalhousie University. MobileHCI 2004, 228-239 (2004).

[4] *Opera's Small-Screen Rendering(tm)*, Opera Software, http://www.opera.com/products/mobile/smallscreen/, last accessed 19th November 2005

[5] Simon Corston-Oliver, *Text compaction for display on very small screens*, Microsoft Research (2001).

[6] *Minimo Project*, Mozilla Foundation, http://www.mozilla.org/projects/minimo/, last accessed 19th November 2005

[7] *Minimo small screen rendering stylesheet*, Mozilla Foundation,