

CSS pagination and editing support

Angelo Di Iorio — University of Bologna
diiorio@cs.unibo.it

1 Author's background and perspective

I am Assistant Professor at the Department of Computer Science and Engineering at the University of Bologna. My research has been focused for quite long time on collaborative editing and documents' segmentation. Afterward I have moved to formatting and layout generation. I have been member of the XSL-FO WG from 2007 to 2011.

My research team has developed a pagination algorithm that produces documents without unwanted empty areas[4]. The algorithm has been implemented for XSL-FO in the Apache FOP formatter. The properties our algorithm works on have been included in the last XSL-FO Working Draft and extended to also enable copyfitting.

I would like to attend this workshop for discussing multiple aspects outlined in the call. First of all, to discuss the future of XSL-FO and to investigate how CSS for printing could be. The XSL-FO WG is currently inactive and CSS is moving fast towards high-quality printing, though several features are still missing. My question is: can we conclude today that CSS is going to supersede XSL-FO? If so, I think that W3C should capitalize as much as possible the experience of XSL-FO and should support the convergence of XSL-FO results into CSS.

This workshop is for me a way to start working again on these topics. Section 2 discusses some extensions to CSS pagination based on my previous work with XSL-FO.

I would also like to underline an important application of CSS pagination. CSS Paged Media is primarily meant to provide an alternative view for content that was originally designed for being published on the Web. The CSS pagination could seem to be needed only in the final steps of the production chain, when producing physical books.

On the other hand, editors and proofreaders use to edit drafts *that have the same layout of the final printed books*. This makes the overall editing process much easier and more effective. Also, authors check the final layout of a book before approving it for the final publication. Today this often happens on printed copies of drafts, that are prepared off-line by professional typesetters and sent back to the editors.

The ability of showing the paginated content directly in the browser and allowing editors and proofreaders to edit drafts in the browser can radically change the landscape of professional publishing.

Note that, if we only consider this editing task, there is no need of managing high-quality images and supporting generation of indexes, appendices and other parts that, instead, are extremely important for the final printed publication.

2 Improving CSS pagination and copyfit

The support for pagination already exists in CSS. The CSS Fragmentation Module[1] indicates rules for breaking content and explains how breaks can be controlled by the final user. When producing books, for instance, the properties `widows` and `orphans` control the number of isolated lines at the beginning or at the end of a page. The CSS layout engine is in charge of splitting the content flow in order to meet these requirements.

One strategy consists of moving some lines from one page to the following ones. This might lead to the creation of pages that are not fully filled (in the vertical direction). Page filling is instead a very important requirement whose fulfillment is crucial for the quality standard of many publishing houses.

Alignment can be set with properties defined in the CSS Box Alignment Module[2]: the distribution alignment properties (and its values `space-between`, `space-around`, `space-evenly` and `stretch`) allow users to control how items are distributed in the alignment container and how the space between adjacent items is handled.

But users might prefer to fill pages by changing other traits than vertical space. More important, there are several cases where space distribution cannot be altered at all. In those cases other properties are modified like word-spacing, line-spacing, letter-spacing, margins, etc. These changes are often applied locally to some text blocks in order to adjust the overall layout.

One possible solution is to indicate the set of adjustable properties in the CSS pagination directives, together with their corresponding validity ranges, and to let the layout engine tune these properties for the final pagination. A possible syntax is sketched out in the following CSS example:

```
book {
  word-spacing: 10px;
  orphan: 2;
  widow: 2;
  justify-content: fill;
  justify-adjust-properties: word-spacing, letter-spacing;
}

p.adjustable {
  min-word-spacing: 10px;
  max-word-spacing: 30px;
}
```

The existing property `justify-content` indicates the layout engine to align the content of the element `book` in each page¹.

The new property `justify-adjust-properties` indicates the list of properties that can be modified. Each property can be modified within its variation ranges, that are in turn defined for each element. In the example, the properties `min-word-spacing` and `max-word-spacing` indicate the minimum and maximum values allowed for word-spacing. The layout engine is expected to fill all pages and to fulfill widow/orphan constraints by tuning this property.

¹The new value `fill` has been added but the existing value `auto` would probably be enough.

The local indication of ranges (minimum, optimum and maximum) in XSL-FO relies on space-specifiers. CSS uses a similar approach, for instance, to indicate the minimum and maximum height of an object. The idea is to extend these length specifiers to other properties in order to provide a general mechanism for indicating possible variations.

To the best of my knowledge, CSS does not provide a single consistent mechanisms to indicate lists of modifiable properties and their variation ranges. Some overlap exists with the CSS Flexible Model[3]. It is interesting to investigate similarities and possible extensions for pagination and printing.

There are a lot of related aspects not covered here. In the first place, techniques for taming the computational complexity of such pagination process. It will also be necessary to introduce fallbacks and rules to apply when complete alignment is not possible. A deeper investigation on how these properties interact with other CSS properties and values is also needed.

A strongly related topic is the support for copyfitting in CSS. The idea of listing adjustable properties and their variation ranges, in fact, can be extended straightforward to also support copyfit and automatic content adjustment.

3 Conclusions

There are many other features of CSS for printing that I would like to investigate, for instance the support for footnotes, marginalia and side-by-side content. Though I am not involved in the development of any XSL-FO and CSS formatter now, I am very interested in having these features available in the future and I would like to contribute to their definition.

There is also a different area that will be explored in the workshop and I am very interested in: the browser/editor support for change-tracking and visualization of edits. My research is also on editing patterns and the relation between basic operations and higher-level ones, that can be better interpreted by human users. Such a research can be split in three parts: (i) the analysis of high-level changes, (ii) the development of algorithms that detect those changes and (iii) the development of tools for visualizing them. My research group has investigated the first two aspects and has implemented a XML diff algorithm specialized on literary documents[5]. We have developed our own model of changes and we are now investigating how to display and connect these changes, especially in web browsers. To the best of my knowledge, there is no standard way of describing high-level changes on XML documents. This topic is worth discussing and can be very useful for building publishing tools.

References

- [1] ATANASSOV, R., AND ETEMAD, E. J. CSS Fragmentation Module Level 3. <http://www.w3.org/TR/2012/WD-css3-break-20120823/>, 2012. W3C Working Draft.
- [2] ATKINS, T. J., AND ETEMAD, E. J. CSS Box Alignment Module Level 3. <http://www.w3.org/TR/2013/WD-css3-align-20130514/>, 2013. W3C Working Draft.
- [3] ATKINS, T. J., ETEMAD, E. J., AND MOGILEVSKY, A. . CSS Flexible Box Layout Module. <http://www.w3.org/TR/2012/CR-css3-flexbox-20120918/>, 2012. W3C Candidate Recommendation.
- [4] CIANCARINI, P., DI IORIO, A., FURINI, L., AND VITALI, F. High-quality pagination for publishing. *Software: Practice and Experience* 42, 6 (2012), 733–751.
- [5] DI IORIO, A., SCHIRINZI, M., VITALI, F., AND MARCHETTI, C. A Natural and Multi-layered Approach to Detect Changes in Tree-Based Textual Documents. In *Lecture Notes in Business Information Processing: ICEIS 2009* (2009), Springer-Verlag, pp. 90–101.