Wave as a scholarly document editor: not promising at this stage

I have had a couple of looks at Google Wave here on this blog, initially thinking about whether it can be used to create academic documents.

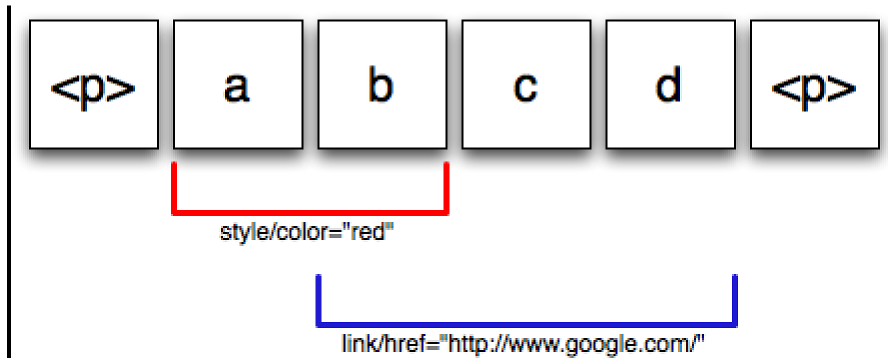
At this stage it **doesn't look promising for document collaboration on scholarly works to me**, although I will keep monitoring developments. In the near future I'll write about whether wave could be useful for a annotation, commenting and marking in the academic space. Can we use it for thesis examination, peer review, assignment marking, thesis supervision and so on? Could you move around a wave in the Open Journal Systems, for example?

For now, here's a summary of what I've found out so far, apart from the fact that it's still running at glacier-speed and is thus completely unusable, even for small waves.

1. **Blips and wavelets are time-ordered.** Not intrinsically a bad thing, particularly for keeping track of conversations but not what you want in an editing application. This means that you can't use Wave like an outliner, or a usable document editor.
2. While there is talk of 'XML documents' in the whitepapers etc, **a wave document in the current implementation is apparently a series of lines of text**. All formatting and what you might think of as structure, such as whether something is a heading or not, is considered an annotation. When the Google Wave client sends stuff to a robot for processing, there are no XML elements there at all, just line-breaks like this; ***\n***. (I could be missing something here, but as far as I can tell paragraph level formatting like headings or list items are not sent to robots.)

   This seems to be at odds with this example from the white paper on OT (Operational Transformation):

   > In Wave, every 16-bit Unicode code unit (as used in javascript, JSON, and Java strings), start tag or end tag in an XML document is called an item. Gaps between items are called positions. Position 0 is before the first item. A document operation can contain mutations that reference positions. For example, a "Skip" mutation specifies how many positions to skip ahead in the XML document before applying the next mutation.
   >
   > $_0$<blip>$_1$<p>$_2$e$_3$x$_4$a$_5$m$_6$p$_7$l$_8$e$_9$</p>$_{10}$</blip>$_{11}$
   >
   > Wave document operations also support annotations. An annotation is some meta-data associated with an item range, i.e. a start position and an end position. This is particularly useful for describing text formatting and spelling suggestions, as it does not unecessarily complicate the underlying XML document format.



http://www.waveprotocol.org/whitepapers/operational-transform

The formatting here overlaps in a very un-XML way. Again, there's nothing intrinsically wrong with this approach, the tree-hierarchy in XML is not a natural property of language, merely a convenience for machine processing. But when it says in the white paper this does not "unnecessarily complicate the underlying XML document format" I'd say it just moves that complication elsewhere. For example, in a document editor application you'd want to be able to serialise those overlapping annotations as XML / HTML, which is going to be complicated. And what if I add another annotation ***style/color="blue"*** at positions ***2 to 4***? Does ***b*** go purple?

As I said before, in my second post on Wave as a scholarly HTML editor, it's disappointing to see such silly examples of documents in the Wave documentation, like this from the Draft Protocol Spec for the Google Wave Conversation Model , which should not be broken into lines at all:

```
<body>

        <line></line>There is a theory which states that if ever anybody

        <line></line>discovers exactly what the Universe is for and why it

        <line></line>is here, it will instantly disappear and be replaced 
by

        <line></line>something even more bizarre and inexplicable. There 
is another

        <line></line>theory which states that this has already happened.

</body>
```