

A special-format multilingual document

Arthur Reutenauer

The topic of my talks usually revolve around Unicode and hyphenation, but for this meeting I wanted to take a break from T_EXnical subjects, as I had just got engaged to my girlfriend of seven years, Tess. Since one doesn't come to a ConT_EXt meeting to hear about the details of a wedding church service or conversations with restaurant owners, I was afraid I wouldn't have anything to say, but I then realised that there was one typesetting issue worth mentioning, namely that of the wedding dinner menu. We had recently seen at a banquet a menu with an interesting layout, as if the pages were cut up and folded back into each other. By folding out from the middle of the spread, other pages appear; in total there are eight pages and three different positions. We wanted to have the same layout for our wedding menu.

There was thus a little bit of *découpage* to be done, and more important an interesting imposition for the typesetting phase. The actual contents of the menu will be in three languages: Swedish, French, and English; but in this article I will focus on the typesetting part.

The layout can be seen as eight columns of three cells each, arranged over the two sides of a sheet. The cells will be labelled 1a, 1b, 1c for column 1, 2a, 2b, 2c for column 2, etc. We start with an XY imposition scheme:

```
\setuppapersize[XY][A4,landscape]
\setuppaper[nx=4,ny=3]
\setuparranging[XY]
```

Some of these cells will be shuffled around during the *découpage*, and we thus need ConT_EXt to prearrange them so that they'll end up in the right order in the final layout.



This is easily done by putting each cell in a buffer, and rearranging them with a bit of Lua code:

```
userdata = userdata or {}

userdata.cells = {
  '2a', '3a', '4a', '5a',
  '4b', '5b', '6b', '7b',
  '2c', '3c', '4c', '5c',
  '6a', '7a', '8a', '1a',
  '8b', '1b', '2b', '3b',
  '6c', '7c', '8c', '1c' }
```

contextgroup > context meeting 2019

```
function userdata.imposemenu()
  for _, cell in ipairs(userdata.cells)
  do
    context.print'\\hskip1sp'
    -- In case the cell is empty
    context.getbuffer{ cell }
    context.page()
  end
end
```

Note that from ConT_EXt's point of view what I call a cell is a page.

Now it's time for the *découpage*! Since a few words are worth a thousand pictures, here are the instructions:

with the recto side of the sheet facing you, identify the two central cells, those that are completely landlocked in the page. Cut this two-cell area on every side but the left. You now have a little flap, that you're going to wrap around the column it's attached to by pulling towards you, so that it goes first in front, then behind the column. Now take the rightmost column and fold it to the left, pulling towards you. Repeat twice, so that you now have a single column. The loose edge of the flap should now align with the part it was cut from, and you can now tape it back together. You're done!

The date of the wedding was set to 23 May 2020 in Uppsala, where we live.



Photo: Frans Goddijn