WYSIWYG Authoring Analysis

After analyzing many different Content Management Systems, there one specific clear distinction between many of the CMS I have reviewed and Composr. The difference is how the systems author content editing.

In about 50% or more of the systems I have reviewed, those systems offer in-line page editing, whereas Composr is one of the systems that offer very limited editing.

A majority of the systems that offer inline editing offer a WYSIWYG inline solution. When you're in the editing mode, you can edit virtually any of the text on the page. Anything that can be edited shows up when hovered on it inside a broken-line box. Clicking on the text transforms that area into an editable text box, usually with basic WYSIWYG controls above the box (text styling, inserting media, justification, lists, colors, and so on). You can make your edits in the box and then click outside of the box to stop editing. At this point, systems differ. Some systems immediately save your changes after doing this (Composr, for example). Others give you an extra step to confirm your changes before saving.

Systems differ again in terms of editing actual blocks, columns, and so on. A few systems simply don't let you do this without editing the template files directly. But many systems do offer some inline way to edit the styling of blocks, modules, columns, and so on. Some systems will allow you to drag and drop blocks and so on around the page (it is absolutely necessary that the theme is responsive for this kind of editing functionality to work properly). Other systems such as Boonex do this via a page builder admin screen, where you're not necessarily dragging and dropping the literal blocks around the page in a WYSIWYG fashion. But instead, you're sizing columns using sliders and choosing which blocks to have on the columns by dragging tabs with the names of the blocks on them (kind of like drag and drop ordering of things). Boonex does not let you control the width/height of the blocks directly (actually, few CMS systems I've tried actually let you edit the size of blocks directly; you have to edit the size of their containers); it determines these based on the size of the column they're placed in, the amount of content in the block, and the size of the user's overall page. I actually think this is a nice way to do it because it still maintains the intuitive aspect but reduces the possibility of breakage due to direct editing of block sizes, which is a bad idea for responsive themeing (unless sizes are based on a percentage, of course). Boonex also has a separate “mobile” layout you can edit as well, so there's no conflict between desktop and mobile layouts; they're separated.

Given Composr's philosophy, it is hard to really determine what WYSIWYG authoring would be great for Composr. Many of the suggestions I would normally provide are not possible without conflicting with the design philosophy. However, here are some suggestions I do have for Composr:

* I am personally not that keen on the inline editing functionality of Composr. Not only is it very limiting, but it also has always been very buggy. However, I'm not exactly sure on how to improve it since it allows editing of titles of content from blocks, something you can't necessarily do via. The comcode page editor. So that's a plus. But often, editing a field messes up other fields, or editing a translatable field with comcode does some weird things after it's saved.
* I think MAYBE (though I'm not entirely confident on this idea) one way we can improve WYSIWYG authoring is the ability to preview while editing. I don't know if this can be done in the editor, but Composr currently lacks true WYSIWYG because you can't actually see the content and its structure on the page as you edit; you're only looking at styled text, unrendered blocks, and so on. Perhaps, if the WYSIWYG editor could be expanded to render the blocks and page etc. within the editor so that people can see the page (to a degree) as they edit, this might improve on the concept of WYSIWYG. However I can understand this may cause styling breaks and performance use.
* Blocks (and any element, really) should also be capable of being dragged around in the editor to their desired places and desired containers (such as div containers) easily. I don't think block sizes should be able to be edited though; the user should do this via. Putting them in containers with sizes defined. But maybe the div container sizes can be edited. They should accept the ability to add constant sizes, min/max sizes, and sizes in px, em, or percents.
* I think the WYSIWYG editor should have a button to allow users to insert basic HTML elements, such as Div containers, flex containers, flex boxes, canvas elements, audio/video elements, anything else you can think of.
* WYSIWYG editing is mainly text based... eg. You edit the positions and stuff of things by moving them around via. Editing spacing and such with your keyboard (and putting your text cursor in the box where you want to add new elements). I think WYSIWYG could benefit a lot better if it combined drag and drop as well. For example, when you add new elements, such as blocks, boxes, containers, and so on... you drag it with your mouse cursor in the editor to where you want it to go instead of using your text cursor.
* Perhaps WYSIWYG should also have the ability for users to put in their own CSS on a per-page basis as well (which gets thrown in the page's head tag). This could allow users to define how media on the page is rendered based on the size of the screen without having to build a new CSS file for the theme or modify existing ones. This might already be possible, I'm not sure. But I was thinking more or less a code editor alongside the WYSIWYG to be able to add things like CSS and Javascript. Or maybe we could make a dynamic block for this that operates similarly to $REQUIRE\_CSS and $REQUIRE\_JAVASCRIPT but instead includes code typed in by the user in a parameter instead of a file.