Eat Your Own Dogfood, or How To Improve LibreOffice Draw

Introduction

Dogfooding is not only important to demonstrate confidence in the own product but also to (im)prove usability in the actual workflow. Therefore, a member of the LibreOffice Design Group, Michel Renon, suggested to use LibreOffice Draw to create mockups instead of Balsamiq Mockups (or any other prototyping tool). So let’s give it a try.

Functionality

Stencils

One of the most relevant aspects of a prototyping tool is the number of provided stencils to quickly draw widgets. Of course you can draw everything manually but that’s annoying, error-prone, and not flexible.

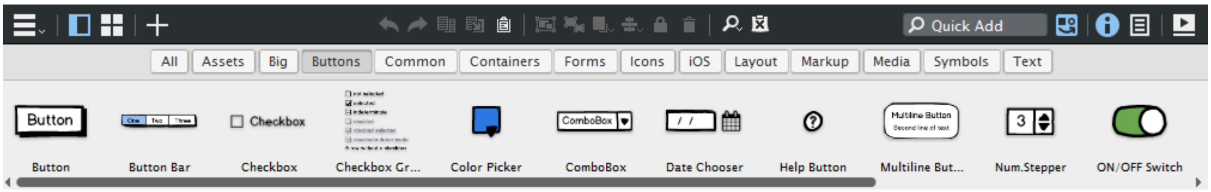


Figure 1: Stencils as available in Balsamiq Mockups.

LibreOffice Draw has the gallery that unfortunately is not helpful since it does not contain UI stencils. But luckily we can create our own gallery and load bitmaps (in case of an existing gallery you have to use *properties* from the context menu in order to add). It is quite simple to make a screenshot of a button and clear it from the caption.
Of course there are wireframe kits available. Most depend on a certain style, are not free, or limited in scope. However, this could be a good starting point. For example, http://www.jankoatwarpspeed.com/free-sketching-wireframing-kit/

Size

Bitmaps are tricky when the size is changed significantly and might results in bad quality due to aliasing effects. This will be particularly a problem when objects are differently sized.

States

But we do not only run into trouble when the bitmap is resized, every widget has also several states. Buttons, to take this example, can receive the keyboard focus (usually indicated by a frame around the caption), mouse focus (fed back by highlighting), being the default of a modal dialog for accept and close (under Windows a frame around the object is quite common here), and last but not least enabled or disabled. Not to speak about the pressed state when the button is down (shown by inverted gradient or raised vs. lowered frame).

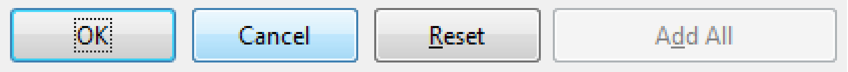


Figure 2: Different button states providing feedback for the user.

Vectorizing

Alternatively we can create a button using inbuilt objects. The leftmost button in figure 2 consists of a rounded rectangle with dark blue solid fill, a rounded rectangle with light blue line, a gradient, and something to frame the label.
First (minor) issue is that we cannot have only two steps for the gradient which is the the case in the original. Second (major) problem is the fact that rectangle text is well placed in the center but cannot have a frame. Replacing the rectangle text by a textbox makes it a pixel-perfect placement necessary. And the text box has a certain spacing between frame and text that is too large here. So we have to add another rectangle for the frame.

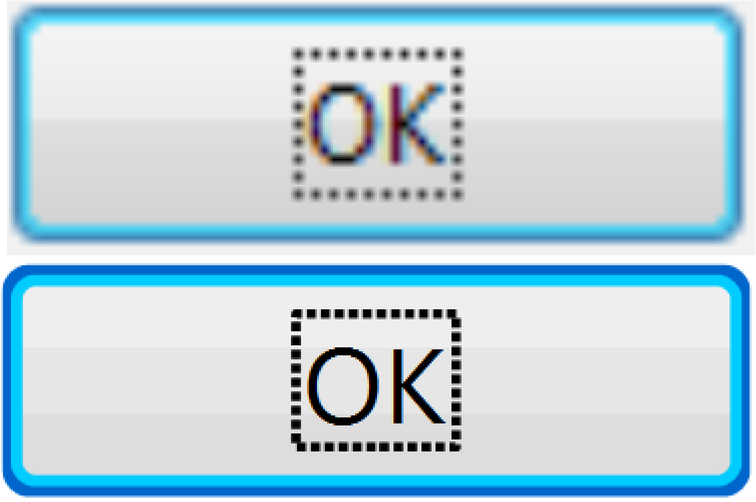


Figure 3: Bitmap of a button and assembled version at 400% zoom.

In the end the simulated button looks not that bad and has at least the advantage of being resizable. But that’s not true because changes are not applied appropriately to the frame and the font, ending up in something strange.

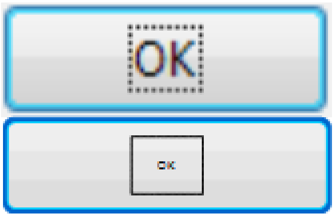


Figure 4: Bitmap and shapes group resized to 400% at 50% zoom factor.

Theming

Some prototyping tools provide the opportunity to switch between visualization modes. The mockup may look like a scribble (default in this blog) or near to the end product. But the more pixel-perfect a mockup looks like the more it comes to bikeshedding. People ask if colors can be changed, why there is no shadow, etc. Those questions are relevant for design tasks but not when the focus is on the workflow and functionality. The scribble mode is perfect to illustrate the difference.

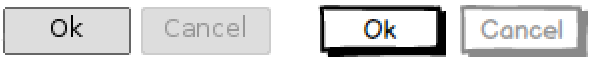


Figure 5: Skins may be useful for different tasks.

But themes play also a role for the different operating systems Windows, Linux, MacOS, Android, or rather desktop environments Gnome, KDE, etc. And you are free to individualize the theme on your environment having, for instance, a cross or a check mark as indicator for check boxes - or a switch at iOS.

Interactivity

Developing a mockup can be a time-consuming process. And it is always a matter of agreement between design, usability, accessibility, simplicity and functionality. At LibreOffice we work in a team and develop mockups in successive sessions. Using the same mockup tool over the group would be a big advantage.
LibreOffice can deal with remote files (content management integration system, CMIS). So sharing the result with colleagues is a piece of cake.

Ease of use

Placement and Arrangement

Normal mockup tools have the option to align controls in several ways. Left, right, top, bottom, vertically and horizontally centered, but also to space out and distribute in both directions. LibreOffice has these features too but not in the sidebar. The same is true for the layering. To bring a shape into front you have to use *arrange* from the context menu (or main menu > modify).

Standardization

The default units are centimeter (at least in the european installation) but not pixel. In order to be able to place control exactly it is necessary to go to tools > options > draw > general and switch the units to point (alternatively you may get a context menu at the rulers). Neither pica nor points are pixels but close enough to start working (see also AOO bug #35835).

Grid layer

Design is much about having a good grid, especially for web site development. Such a grid is used to align the UI elements at defined positions and to have meaningful white space. Libreoffice Draw has a grid feature that can be enabled at view > grid > display grid. But that is more like a cross-section or scaled paper. At least auxiliary lines would be good.

Grouping/Inheritance

Using individual groups of objects, like a rectangular frame with two buttons as mockup for modal dialogs, speeds up the creation significantly. It happens quite often that some aspects of a mockup are changed in a copy, for instance to show effects of checking an option. While

it is easy to copy an object or a whole group it’s a tricky thing to keep both ‘pages’ coherent when the original is changed afterwards.
This issue is solved differently over the existing tools. Basically the group of objects would need to go into the gallery section as a new stencil with the option to change this stencil group when inserted but also to change the parent and update to all pages.

Collaborativity

Openness

LibreOffice is free as in beer, open to everybody with a huge community. The development tools are simple and commonly used, processes are transparent and user-friendly. We define the status quo in respect to collaborativity.

Testability

Mockups are meant to test ideas. The layout should be shown to people in order to ask for opinions or even better to watch how users operate the prototype. There are tools that produce clickable prototypes with a good number of available events to make the mockup interactive. Other tools do only link different pages on click. In any case it’s better to present just a picture with some description than nothing.

Conclusion or How to improve LibreOffice Draw

A very easy improvement would be to promote alignment and layering functions at the sidebar. It is an essential feature in LibreOffice Draw, not only required for building mockups.

Second idea is to add another gallery with stencils for one or more themes. But as discussed above this might not be the best solution. Alternatively, we should consider to have vectorized objects with states that have a certain visualization. In this case some features should be added: that is first of all whether or not a property is resized on scaling up/down. Line size, font size, spacing, indentation, shadow, and probably more would be affected. And it would be useful to have interactive events. The checkbox toggle its status on click, a dropdown expands the attached menu, lists allows scrolling, and so on.

Quite important is the need for user-defined stencils with grouping and inheritance. We would need options to modify the group when placed at a page but also to edit the source. Updates should be applied manually in case individual modifications has been done in the document.

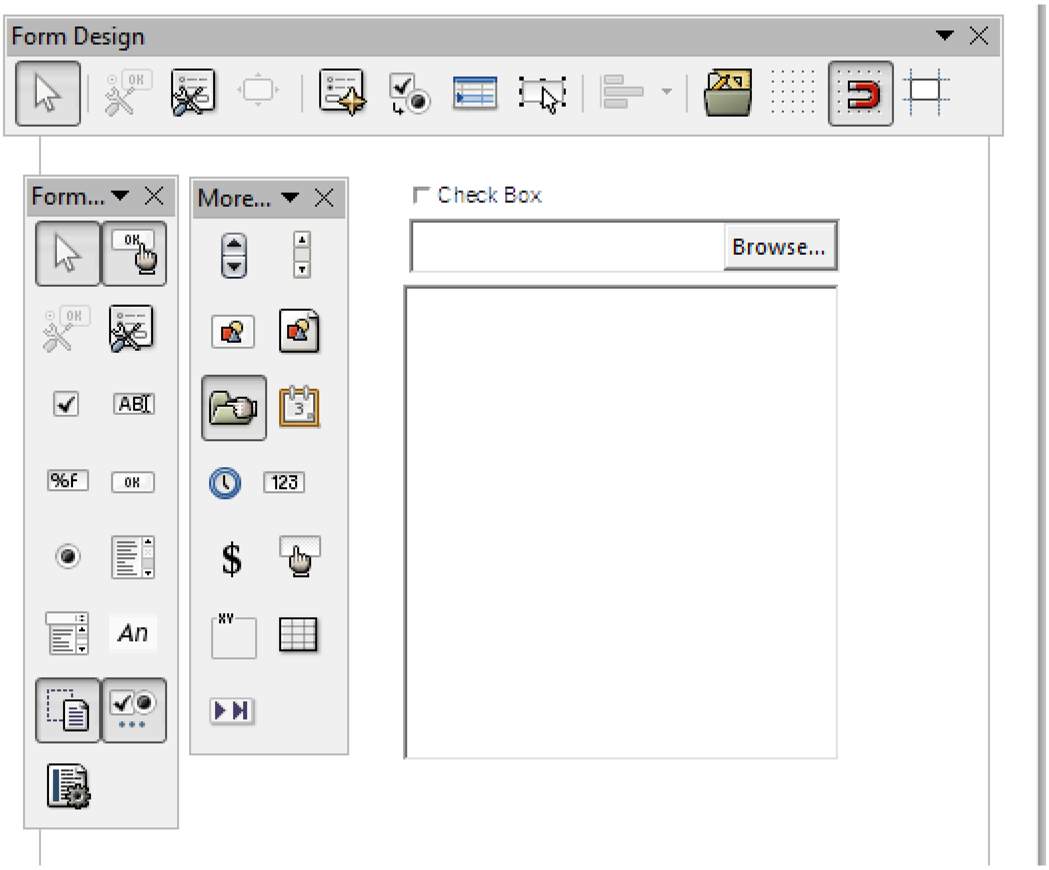
Another feature that has a general benefit is a list of objects in the current document. If more than a few objects whether simple shapes or complex stencils are placed in one document, at best stacked above each other and grouped in a way that is hardly comprehensible, it is not easy to find and select the target. Proving a list of objects supports this situation.

The more objects are included in the gallery the more useful becomes a filter. We suggested an update of the shapes sidebar that comprises of such a feature. But it requires also to have good names for the objects. For instance, icons are a relevant kind of object, available in a huge number (Breeze consists of >2500 icons) what could lead to a performance issue. If you look for the symbol to start the file dialog you would probably search for load, open, or directory. This might be successful but how to find the icon for text box? Furthermore the sidebar is limited to three columns and a certain width. This is not flexible enough for browsing a large number of small objects.

Vision

This discussion is also a question about Draw's vision. Do we want to compete with, for instance, Krita and create versatile raster graphics focusing on brushes, filters, effects, etc. Or is it rather a vector drawing tool like Inkscape stressing objects, layers, connectors. Is it a simple tool to just add a few shapes or do we want support complex workflows?

+ Form controls (View>Toolbar>For...)


Form Design
OK
Form...
More...
OK
ABI
%F
OK
An
123
$
XY
3
Check Box
Browse...