User Interface and Functionalities for KOffice

Author : Luc Marschall
email: lucmars@infonie.fr

Make with OpenOffice and Inkscape.

Introduction

From a software development point of view, it makes sense to consider personal computer and softwares as a whole system including the user; otherwise it looks like a P.C. without an operating system, or like an uninstalled O.S. But in this way the user appears to be an unknown; that's especially true since the P.C. is now a commodity, that is to say: built for everybody. Consequently, that has some involvements which should be taken in account for the purpose of an user interface. Though, one can address all the possible user's configuration material, mainly the size of the screen monitor and the various resolution setting, one cannot know exactly the user's context.

In fact, the user's context is always changing and all the knowledge that one can have about it will be fuzzy; otherwise one should put the user in a very determined situation, as if it was himself a automated part of a whole, but this particular case is not a commodity. That's due to a gap around the data shared between the computer and the user : the computer process data which are related to a context for the user, hence, the both are working together in two respective and different levels, data on one hand, and on the other hand, data related to a context, which is a common definition of the information.

This present document is a proposal for an U.I. according to this consideration. Mainly, the underlying principles is to provide an interface whom the structure is itself an information and by taking in account that the user works at this level; since the user is an unknown, despite the fact that all the functionalities are predefined, their organization is let to user. There are however some other important considerations: picture and linearity.

A picture is simply a frame, inside the frame this is the picture but not outside. Hence, this the case about monitor since it is a common material interface; it doesn't matter here to be in text or in graphical mode, one is still in a frame and his space. Thus, graphical interface just brings more frames. Moreover, text and graphic work differently: one understands a text but one recognize a picture. At the informational level, the both cannot be treated in the same way; for example, giving a name to an picture file don't help, one needs to view it. And using picture as a symbol or a sign, the icon, is indeed a way for graphical elements to work as a language; that requires a training for the user.

Linearity means that things go the one after the other, in time and/or in space. It doesn't matter that the computer can perform some simultaneous tasks since the user can not. From his point of view, linearity in time is a sequential or a step by step process; in space, through the monitor, that appears for example as a list, mainly the informations are displayed on a single axis (top-bottom or left-right).

Of course one experiments linearity both in time and space, but the user interface is not an environment with four dimensions, just three, since a monitor is bi dimensional. This later being the upper frame, the ability to manage space or frame is important; that can minimize the linearity in time (same space, same time), and by doing so, let the user to organize his informations.

General Structure of the KOffice User Interface.

One presents in this section the commons elements and their functionalities for KWord, KSpread and KPresenter. Since one wants an informative interface structure, one presents in first the terminology used in this document about these elements. The terms below and their definitions are associative in so far as that makes sense. Further, one will present the features and their descriptions.

Terminology

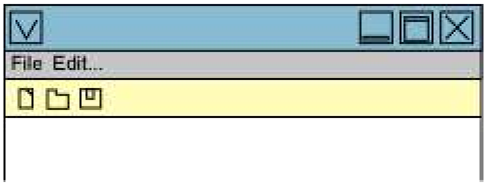
<u>Frame</u>: a shape delimiting the informative contain, textual or graphical.

<u>Window</u>: the usual frame in G.U.I., one just has to retain here the usual window' sfunctionalities or properties (minimizing, maximizing... and so on). Depending the case, one will add some new features or inhibit some existing features.

<u>Tool</u>: all functionality of the application available to the user under a textual or graphical presentation

<u>Menu</u>: a list of functionalities under a textual presentation

<u>Bar</u>: a frame inside a window and dependent of the window presenting a list of functionalities

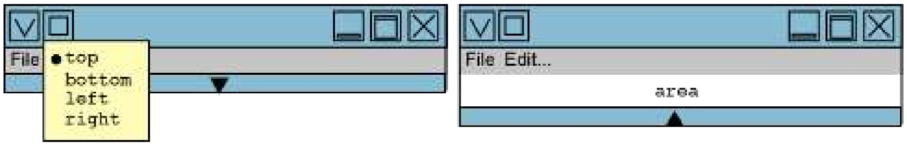


A Menu Bar and a Tool Bar

<u>Area</u>: a rolling frame inside a window and dependent of the window that the user can fill with some element of the interface.

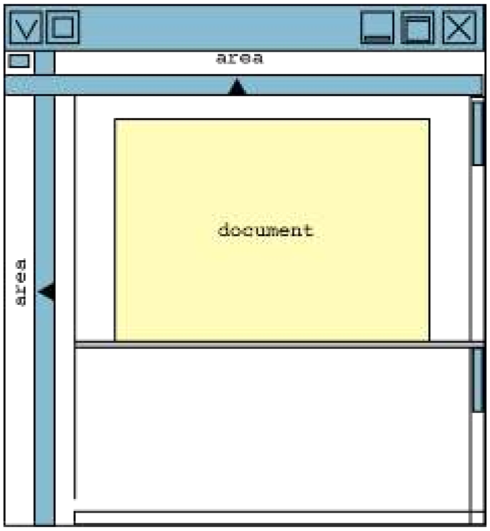
<u>Main</u>: a value meaning the access to all the functionalities.

Main Window: this is the application's window, with a menu bar and an area. His window menu has an additional feature to set his position.

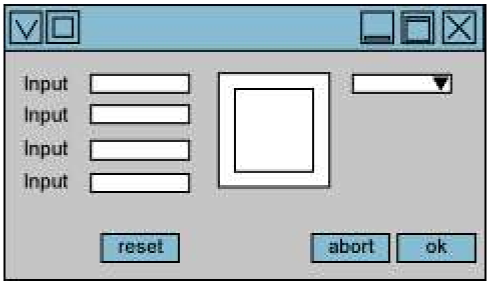


Document: data of an opened file presented in a frame

Document Window: the opened file's window without menu bar but with two areas (one at top and one at left). The document window can be split in frameset, horizontally or vertically. His window menu has an additional feature to arrange the windows between them.



Tool Window: when a tool requires several input to be set altogether. His window menu has an additional feature to place the tool window in a frameset of the front document window.



Tool Group: for all tool which needs no more than a button or just one value at once. This is a set of tools or just one tool in a frame. a tool group can be reduced to one tool or expanded up to all his tools. A tool group floats over all window excepted the main window, and one can drag it.

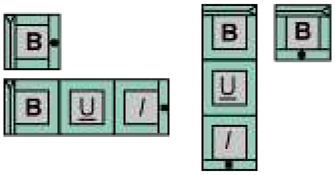
Tool Box: a floating frame over all window excepted the main window and containing the tool groups. It is sizable and, through a roll over menu, one can reduce it, expand it and set it as a tab or as a frame in an area.

Features and Descriptions

As usual the main menu gives access to all the functionalities and since the user can organize the windows and the tool boxes or tool groups in several ways, the main menu should have an entry about windows and an other entry about tool boxes. Moreover, as one will see further, the user can select windows and tool boxes or groups and as well as make a multi-selection. By this way, the functionalities of each element can be applied from the main menu.

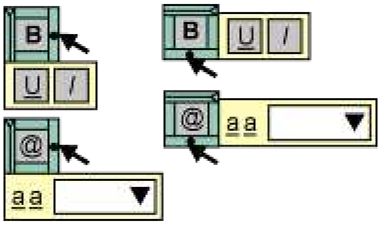
Tool Group

A tool group presents a set of functions. The tool group can be minimizing or maximizing by the user from one tool to all tools. There is no intermediate position.



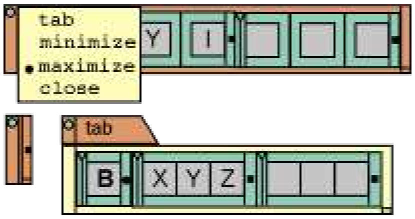
Maximized state is shown by a black square, a click on it minimize the tool group and this state is shown by a black dot. A double-click on the dot maximize the tool group.
A double-click on the grip (at left-hand side or at top) change the position of the tool group from the horizontal to the vertical, depending the initial position.
These functions can be as well reached through a contextual menu by a right-click on the grip.

When the tool group is minimized, a click on the black dot display the hidden contain. Once displayed, one hides it again by a click on the dot, or by a mouse out of the displayed frame.



Note: this is always the same procedure for hidden contains in this proposal, as well as the position of the displayed frame : below the tool group in a horizontal position and on the side for a vertical position.

Tool Box

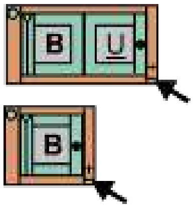


As shown, the tool box contains tool groups. It has a menu with four settings from the green button at top-left:

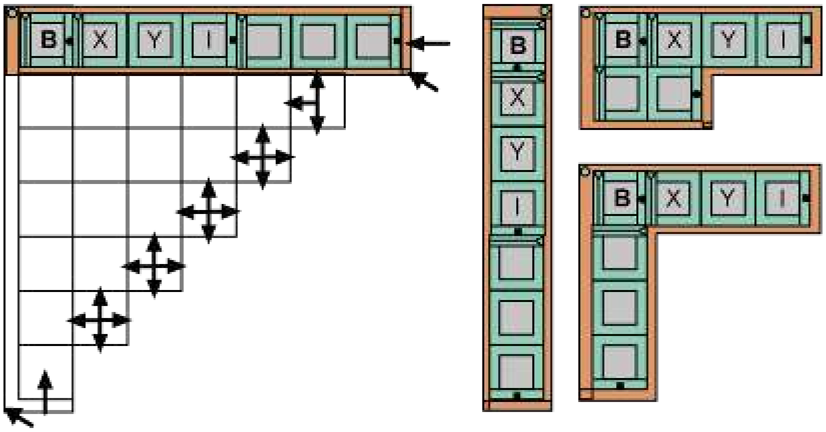
- tab
- minimize, signaled by a black dot
- maximize, also performed through the dot
- close

A double-click on the green button close the tool box, one opens it through the main menu.
One can change the horizontal or vertical position in the same way as for a tool group.
Set as a tab, one gets the tool group by a click on the tab; displaying and hiding follow the same procedure as previously. On the contrary, in the minimized state, this is nothing or all.

The surface of the tool box frame adjusts herself to the whole surface of the contained tool groups, and reversely.
For example by reducing the length of the tool box. In this case, each tool group minimizes itself tool by tool until the minimized state as seen previously. When the surface of the tool box is inferior to the whole surface of the tools groups inside, then the sign “+” appears at the right-hand side. One gets the hidden tools through this sign with the same procedure as above.



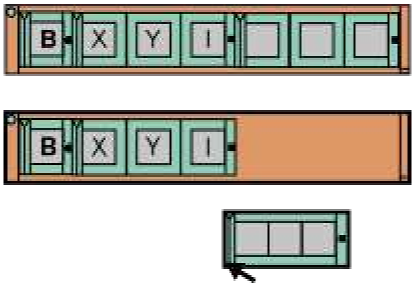
In fact, the tool box can be resized manually from the horizontal to the vertical, orthogonally. The minimized tool group or a tool group with a single function in the unit of each step.



Note that the shape of the tool box follows the surface delimited by the tool groups. It looks like a well known game, but in this way the space is optimized. As well, tool groups optimize their place inside the tool box frame. Reversely, maximizing, minimizing, remove a tool group or change the position of a tool group modifies the surface of the toolbox.

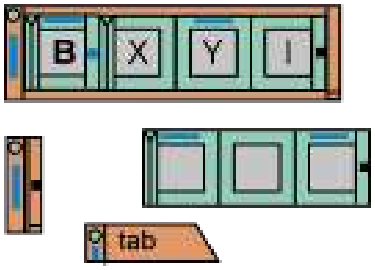
Tool Box & Tool Group

They share the contextual highlighting feature under two considerations : between them and relatively to a selection.



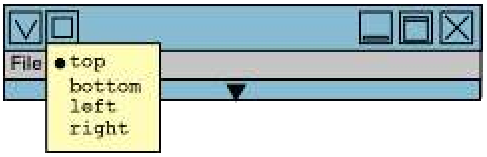
Selecting one or several tool groups sharpens their frame and the frame of the relevant tool boxes. This feature aims to show the structure and the organization of the tools. Though, that don't prevent the user to arrange the tool boxes and tool groups as he or she wants.
One can fill as well a tool box with all the tool groups and one can envisage that the user creates his own tool boxes.

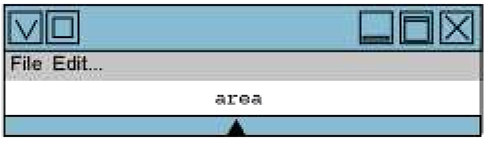
Relatively to a selection related to some functions, the relevant tool boxes and tools are highlighted as below.



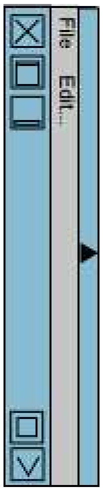
Whatever the state of the tool boxes or the tool groups, a highlight indicates the relevant tools for a selection, even if they are hidden. This feature helps the beginner but as well a trained user since one can separate the tool groups from the tool boxes and modify their organization, that helps to locate the tools over the screen. Mainly, relating a selection to the functions brings an informative level.

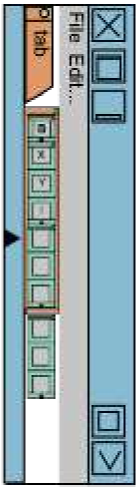
Main Window





As seen previously, one can change the position of the main window through a menu : top of the screen, at left, at right of the screen or at the bottom. These functions can also be set in the contextual menu of the window.



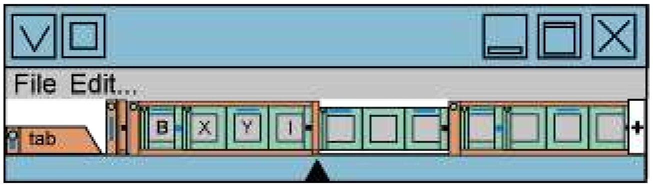


At left and right position, the main window is set vertically. As one can see, it's not just a clockwise or counterclockwise rotation. Everything is set vertically, the usuals buttons of the window are at top, the menu starts from the top, and the eventual tool boxes and tool groups in the area adopt the vertical position.
Of course one can envisage to set the text horizontally through the option panel.

The position of the main window has priority over the other elements of the interface (document window, tool boxes...). In other words, the overall screen is split in two zones, one for the main window and one for the other elements. By this way the user has always an access to the main menu, thus to all the functionalities.
Note: splitting the pitch in zones could solve the tricky situation of an extended desktop. In this case, the virtual pitch of the combined monitors is split in two zones, and then once again for each zone. By this way, one can have an intermediate position at right; that is to say, at right of each monitor.

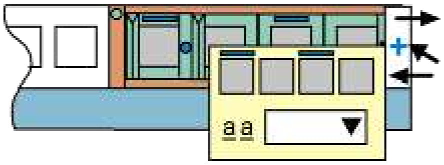
Area

Functions of the area are the same whatever the kind of window which presents it. One opens or closes the area by a click on his grip. One can put inside all the tool boxes or tool groups if one wants and empty the area by a right-click on the grip through a contextual menu or manually. Of course, tools inherit the vertical or horizontal position of the area.

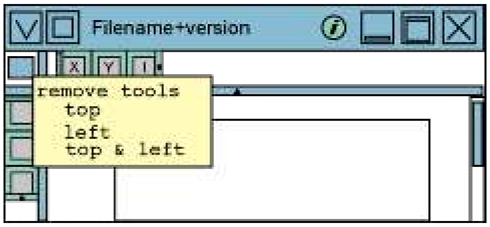


When the area is overloaded, the sign “+” appears. It is contextually highlighted as well as the grip of the area when it is closed. Accessing to the hidden tools can be done as follow :

- through the sign “+”, a click displays the list of the hidden tools
- a maintained click above the “+” scrolls the tools in one sens
- a maintained click below the “+” scrolls the tools in the opposite sens



Areas in a document window can be opened or closed separately. There is a menu at their intersection to empty them individually or altogether.

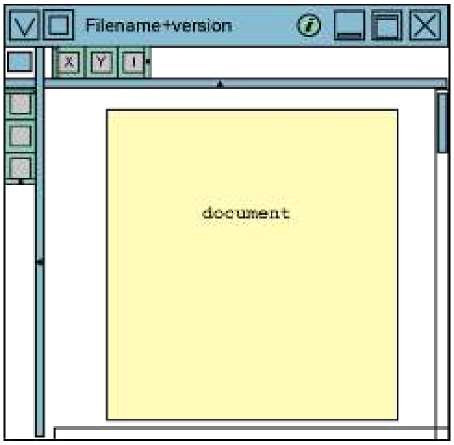


Though, the user uses them as his or her convenience, the interest of an area in the main window is different than in the document window. That is relevant when one has several documents to work, areas in the document windows permit to focus the right tools with the right documents. The others tools can be left aside or in the area of the main window, whether they are unused or relevant to all the documents.

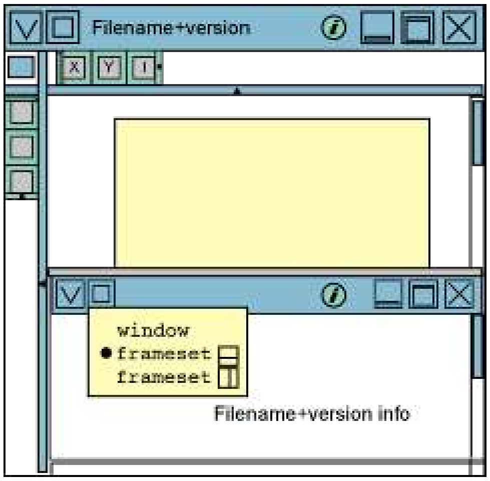
This implies that the dispatching of tool boxes and tool groups is relative to each document window, at least relatively to the contain of their areas. But one can also envisage to take in account the spatial dispatching for each document window.

<u>Document Window and Tool Window.</u>

This is the window of an opened file. The file name, his version and a dot info are displayed at his top. The color green or red indicate respectively if the file is locked or writable. If locked, the irrelevant tools are shaded.

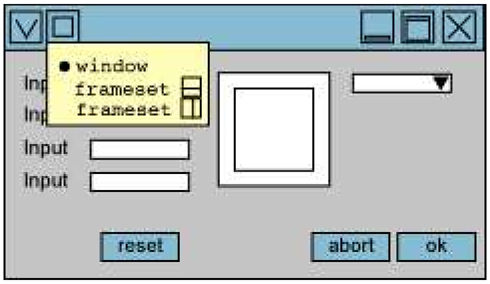


A click on the dot displays an information window inside the document window as a frameset. What kind of information has to be shown is not the present topic, though one could get here at least some statistics about the document.



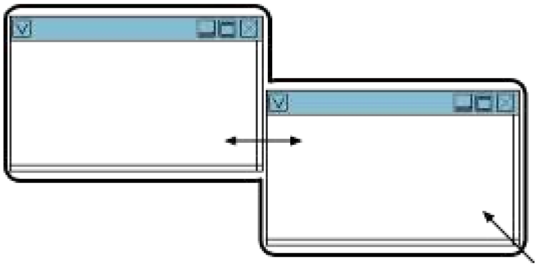
The information window is like a tool window, excepted by the fact that it is opened by default as a frameset. Though one can set it as a window and change the horizontal or vertical position of the frameset through his menu. As a window and if several documents are opened, his contain is relative to each document window in front.

On the contrary, a tool window has to be set as a frameset through the same menu. But in this case, though depending the functions, the frameset has no scroll bar.

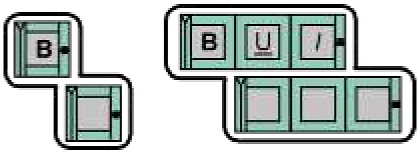


Nonetheless, the size of the frameset follows the size of the tool window, for example by minimizing the tool window or by rolling it up. As well, a click on the frameset separator reduces the tool window to his title bar, an other click expands it. One can equally resize the frameset manually. When the frameset is vertical, the title bar of the reduced tool window is simply vertical.
If the document window masks partially the tool window, a click on the latter fits the frameset to his dimension.

Multi Selection



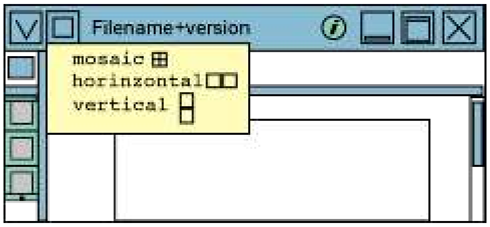
With the shift key pressed, one can select several windows and tool boxes or groups and then apply a function. Selected windows behave as a single window with frameset. The release of the shift key stops the multi selection. Below, an example with tool groups minimized or maximized altogether.



If a function does not apply to some objects of the multi selection, then they are released. Concerning the main window as a part of a multi selection, functions can't be applied on it.

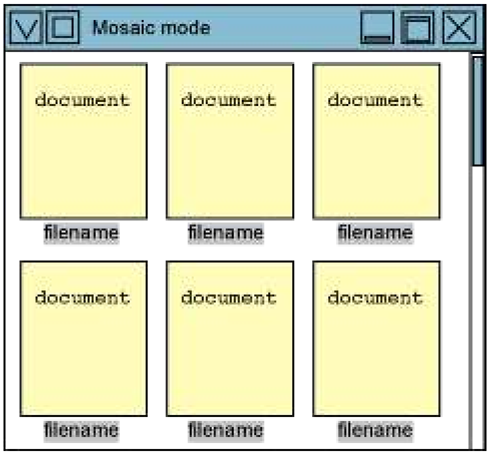
Arranging Windows

One can see here the benefit of the multi selection. The document window has a menu which provides some arrangements between document windows, that works as well with the tool windows :

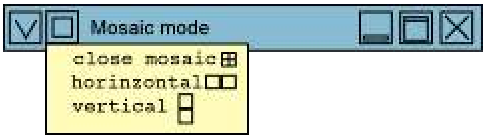


- mosaic
- horizontally or side by side
- vertically or above and below

Mosaic is an on / off mode. This is a window indeed which shows all the documents displayed in columns. A column is reserved for the tool windows, their inner frame only is displayed. That brings an overview to the user and an other way for managing the tools windows. One can envisage a tool bar in the mosaic window with some functions for the display, or for the links between the documents.



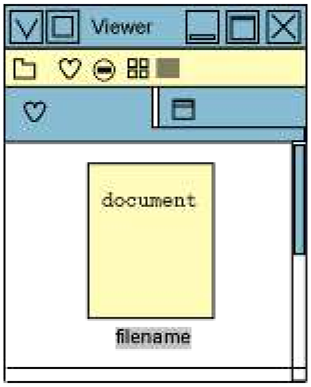
To bring in front one or more documents, one just has to select them and click on the selection. In so far as the selection is still active, one can apply a function on it. One returns the document windows in the mosaic through their menu (that includes some tool windows if they are selected).
Thus, one turns on the mosaic mode through the document window menu, but one turns off the mosaic mode through his own window menu.



One can arrange as well the mosaic with the other windows; set vertically, the thumbnails of the documents are presented in rows.

Arranging windows horizontally or vertically is applied to all windows, excepted the main window, if there wasn' t any mlti selection. Tool windows keep always their size.

<u>Viewer Window</u>



The viewer is an other way to bring an overview of all the documents, but it shows the documents from other applications too. Moreover, it's a means to pick some files as favorites and displayed in preview.
It is composed of two tabs, one for the favorites and one for the documents. There is a tool bar with at least three functions :

- open the home directory
- add the selected files as favorite
- remove the selected favorites from the tab

One can as well envisage some other functions for the display and for the sorting.
The viewer windows is opened from the main menu and by default it is narrowed to a single column. Thumbnails or previews are presented in column as the mosaic window if one enlarges the viewer window.

Menu Entries

Some precisions about the entries for windows and tool boxes or groups in the main menu. As usual one retrieves here each menus previously presented.

Concerning the menu about windows, it is not required to decline each kind of windows, the functions listed apply to a selection or the window in front. Hence, let's say for the mosaic mode, functions like “m osaic on / off”; “ in mosaic”; out mos aic” are sufficient.

The menu about tools is a more complicate. The user could apply the following functions :

- restore the tool box, this apply as well for a tool group since one can close only a tool boxes.
- restore all
- restore default
- create a tool box, from scratch or save a modified tool box
- make a session of tools and select a session
- the list of the tool boxes displayed or closed

Structure of the tools

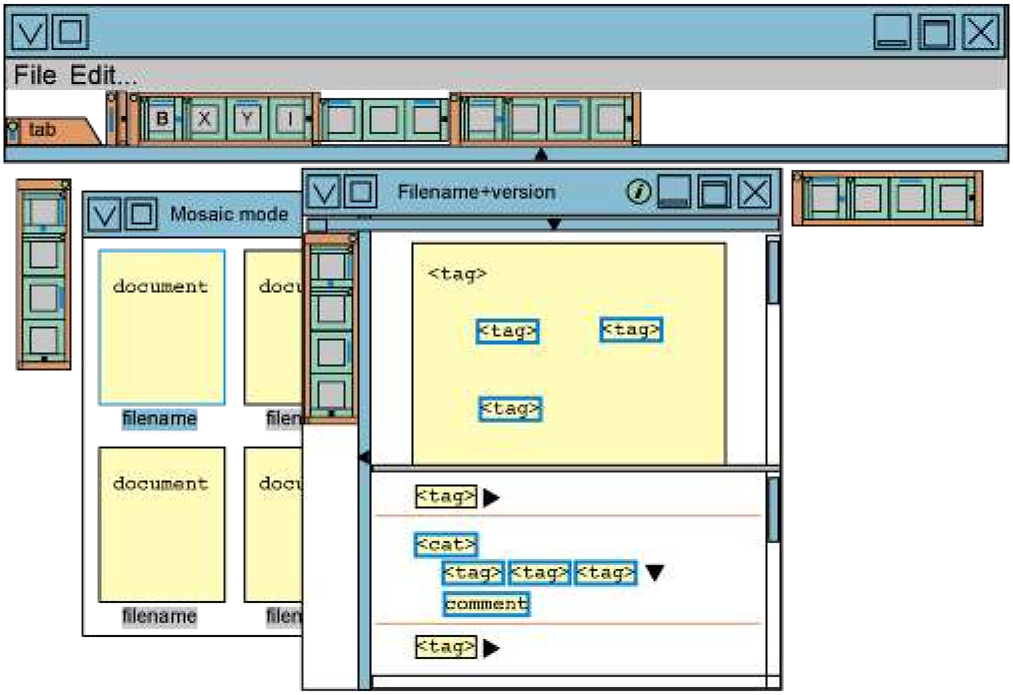
Tool boxes enable a personalization on the go, though, tool groups can not be personalized, their structure is predefined. But this is already the case in most office applications, hence, this is not a mystery to define the tool boxes. Nonetheless, office applications share a lot of functions but their importance is different for each application.
Thus, the real matter here is to determine if a function should be a single tool group or not. The best is to let the user decide: in the tool box menu entry, simply two functions, group and ungroup, like in a illustration application.
Grouping and ungrouping are temporarily only, unless the user create some others tool boxes with these “n ew” tool groups. The restore default function above restore the ungrouped tools. Temporarily grouping and ungrouping is the best solution, that avoids to turn the structure in a mess and to lose the user.

Since a document window can include a tool window, this latter should not be too large. Hence, functions which require a window but can be apply to several objects, colors or borders for example, will have their own tool window. Otherwise, functions which are specifics to an object, like setting a table, will be as tabs in a tool window.
Consequently, tool windows don' tclose automatically after validation, they update their contain accordingly to the front document window; by this way the user can also organize his job in batches. At least the choice could be proposed in the option panel.

KWord, KSpread and KPresenter.

One wants to present here for each application some proposals about some of their features and their apparency.

KWord and Annotation

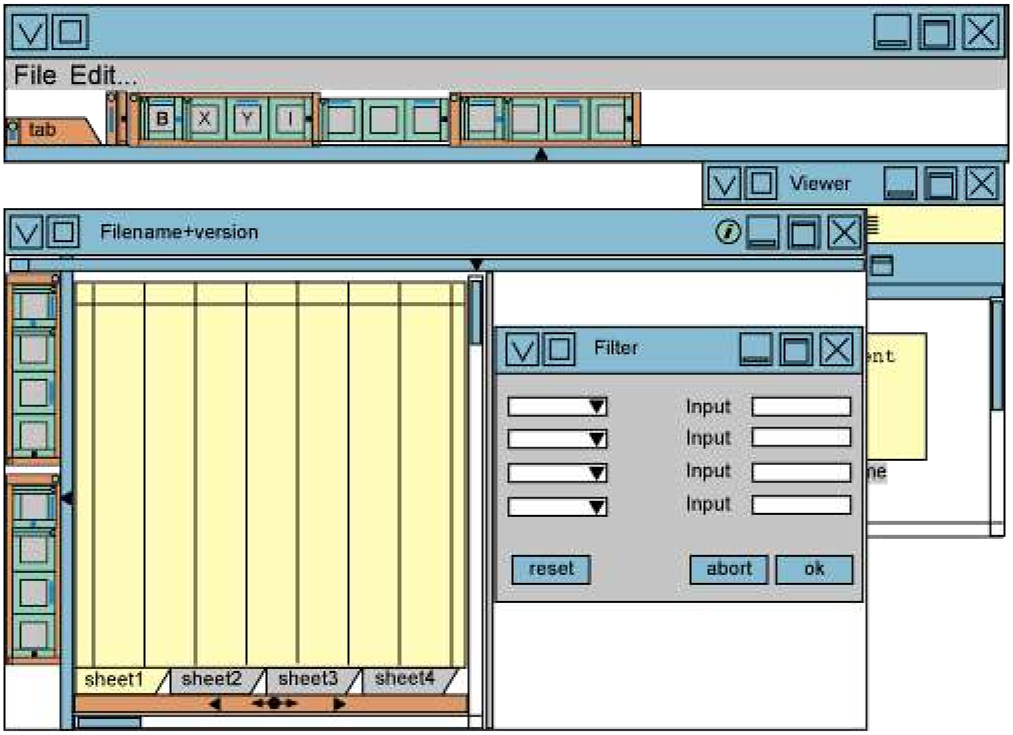


Like the information window, the annotation window is opened inside the document window. Though the principle of annotation is simple : a tag and a comment, or a comment alone; it is working at the informational level. The tag itself can have a meaning, by the use of some colors for example, it is not just a simple mark related to a comment. Moreover, tags can have a relation between them.

Hence, between the annoted document and the annotation window, the relation has to be shown. In the picture above, the relation is highlighted. One can also envisage that the user ties the tags together and make by this way a category.

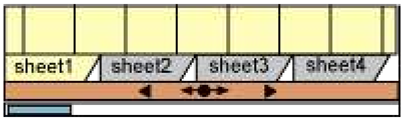
In the annotation window tags and tied tags include their comment which can be expanded or collapsed. There is a lot to do in this domain, not just about the interface, but about the functionalities of the application.

KSpread and Data



Since spreadsheets are usually wide, as one can see above, the frameset of the document window is preferably vertical. One can put a tool window inside, as well one could show here all the different commented cells; a map of the link between cells, sheets and spreadsheets or make a graphic

If some sheets are hidden, because the window size has been reduced or else, one proposes here a different way to get them.

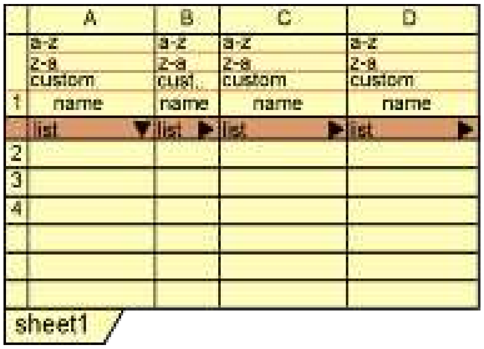


Between the horizontal scroll bar and the sheets, a bar with three functions:

- sheet by sheet at left
- sheet by sheet at right
- defile left or right.

The latter function is used like this: once the pointer is over the dot, it begins a hand; at left or at right, the more one moves the hand along the bar (until the two others buttons), the more the defilement is fast. This function could be as well related to a mouse wheel. Of course, one can also envisage two others functions to go directly at the end or at the beginning.

Here below, a other way to present the filtering functions in a sheet. Above the header line, the usual functions, and below the header line, a bar for each column.

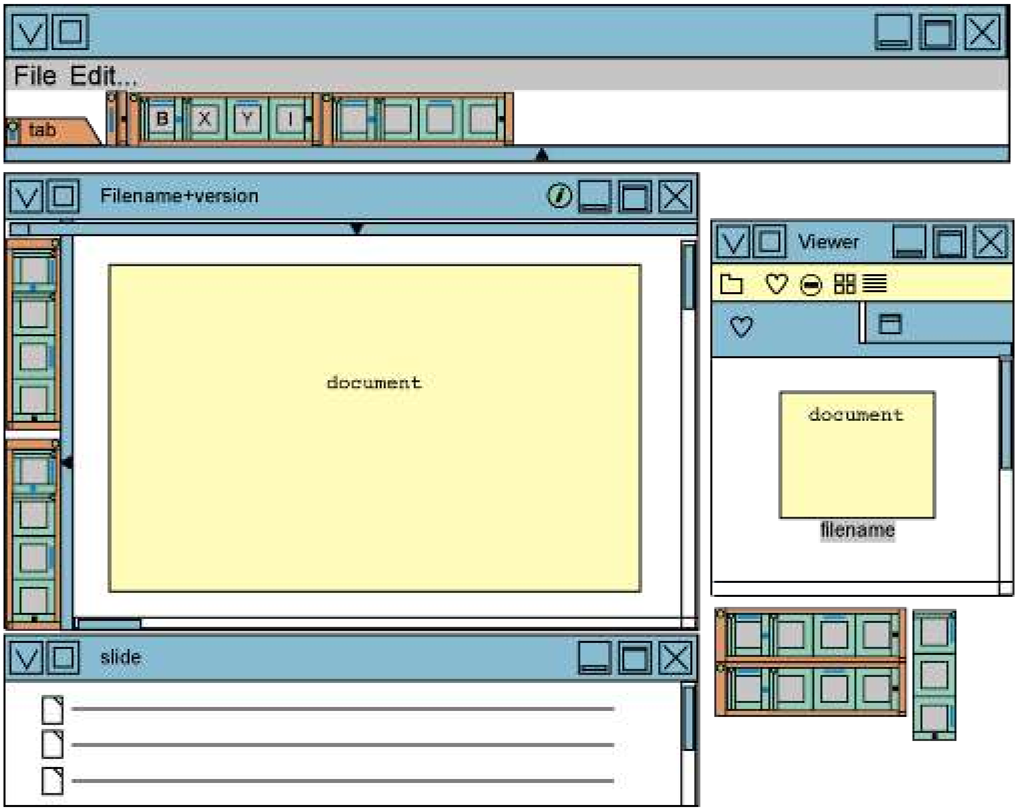


	A	B	C	D
	a-z z-a custom	a-z z-a cust.	a-z z-a custom	a-z z-a custom
1	name	name	name	name
	list ▼	list ▶	list ▶	list ▶
2				
3				
4				

sheet1

A click on the arrow list the data in the column, a click again restore the data in the column. By using a bar, one can envisage some functionalities through a contextual menu. For example, display the listed data beside their column, in the frameset, or as well, get an access to some functions.

<u>KPresenter and Slides</u>



Certainly the application for which the viewer window is very useful. That implies the ability of drag and drop, from the favorite tab to the document window. By this way one can compose easily his slides; as well KPresenter could have his own favorites or libraries of graphic. Below the document window, the tool window for managing the different slides and their links. As with the annotation window, one could show the relation between the objects in the document and the programmed behaviors listed in the tool window.

Conclusion

Since the user is an unknown and that one doesn't know his informational context, and since the screen monitor is indeed the main interface implying a determined space, one has proposed here an UI which lets the user organize the tools and windows as he or she wants. Moreover one has tried to relate between them the functionalities of the interface in an informative way, and as well a way to relate the different objects of a document; objects which are always working at the informational level for the user.