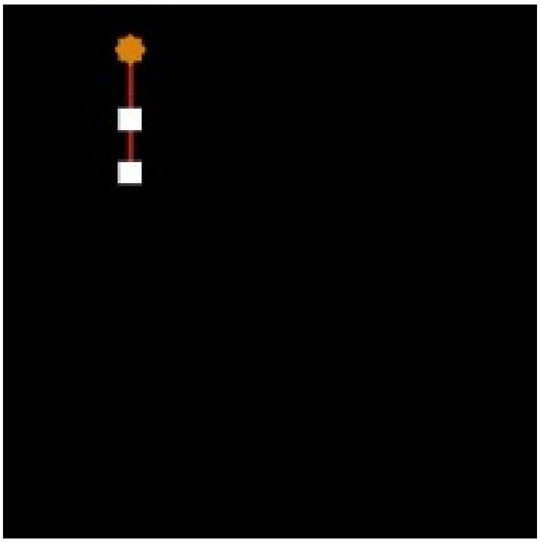
Idea: A Practical Way to Group Tabs

This idea is based on the concept I submitted to the Mozilla Design Challenge 09. I try to regroup fields of interest via a tree structure – without knowing the tab contents.

Because I am now focusing on creating tab groups according to certain rules, there should be no problem in exporting this idea to other projects. My goal is to create as few tab groups as possible, these groups being useful and well organized. To avoid confusion I'll explain my idea step by step.

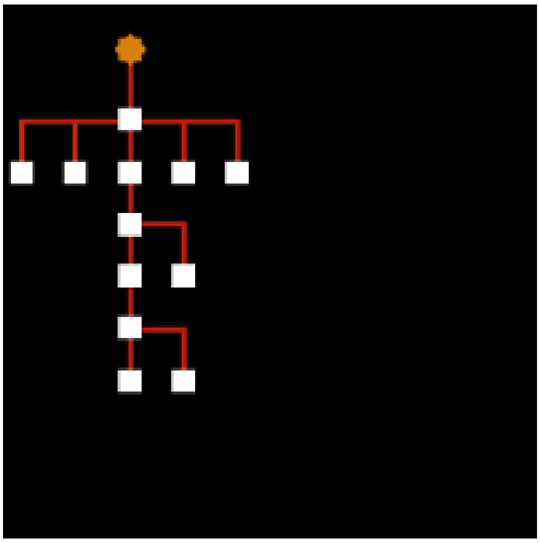
Open browser. A "Birth Tree" is created that opens the start page. As the name "Birth Tree" suggests, this is where trees are born – "tab trees". The first branch of the tree is the start page. There is an extra rule for the "Birth Tree": *"Each newly opened tab (superior tab) first belongs to the "Birth Tree", as long it has not opened another tab (sub tab). If another tab (sub tab) has been opened, both (superior and sub) tabs get bundled into a new Tab Group. This kind of extraction applies as long as the Tab Group is comprised of more than one website"*. "Superior tabs" are tabs that are created by clicking on the plus-symbol or pressing CTRL+T – "sub tabs" arise when clicking on a link (middle mouse click on a link, for example). This kind of birth process prevents a user from opening 50 tabs with CTRL+T and getting 50 tab groups. In the following explanations, I'll concentrate **exclusively** on tree structure and a tab-group that has gone through the birthing process (from now on all tabs are "sub tabs").

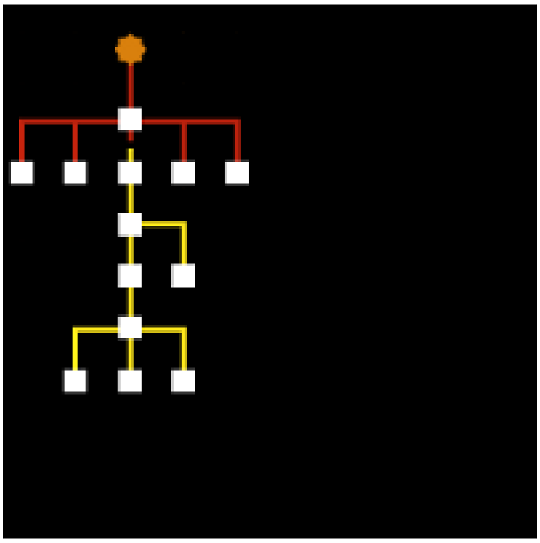
Until now our tab group would look like this:



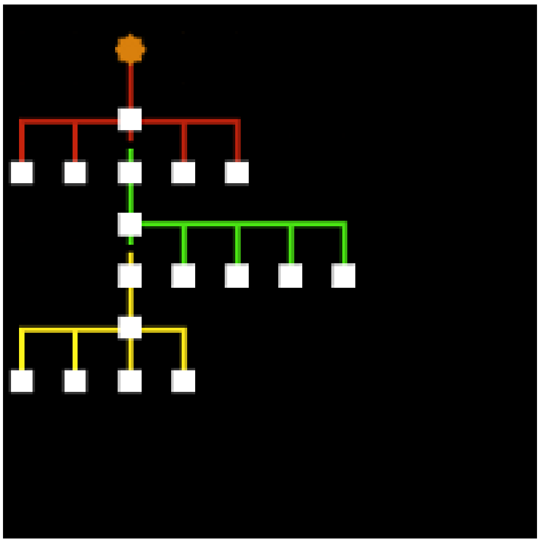
The highest point on the picture (the star) is the "starting point"; it is the founder of the group. The upper white box is the first website and the other, the tab that was created from it.

The “Birth Tree” rule would make no sense for “adult tabs”, that’s why I invented another rule for them: *“When you open more than two tabs from one website, you move higher up in the tree structure until this rule is broken again or until you reach the starting point, in order to branch off the tab tree and to create a new Tab Tree Group”*. This rule sounds somewhat complicated, but pictures should help understand it better.

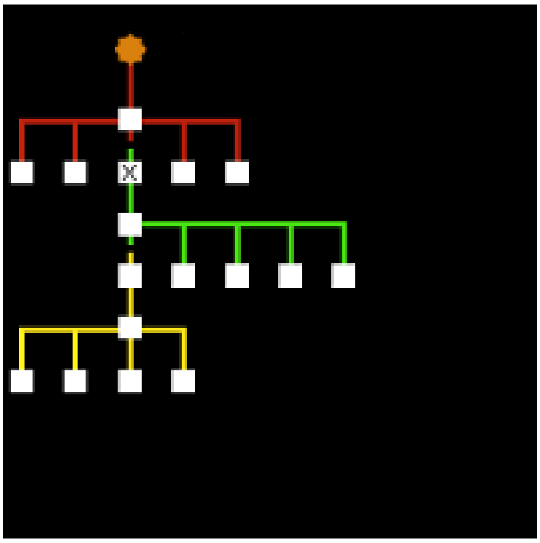


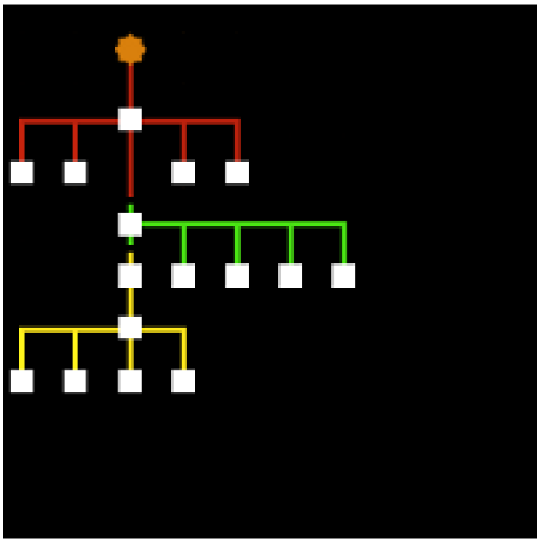


As you can see, the new group (yellow) gets integrated into the existing tree. The advantage of this procedure is that the allocation remains flexible. This fits the need of the user who can “jump” to higher tabs – from the point of view of the tree structure – and create new tabs from there. The following example will help you better understand what I mean. We assume that the user creates enough tabs at a higher branch point to build a new Tab Tree Group. This would affect the other groups (especially the second one) as follows:



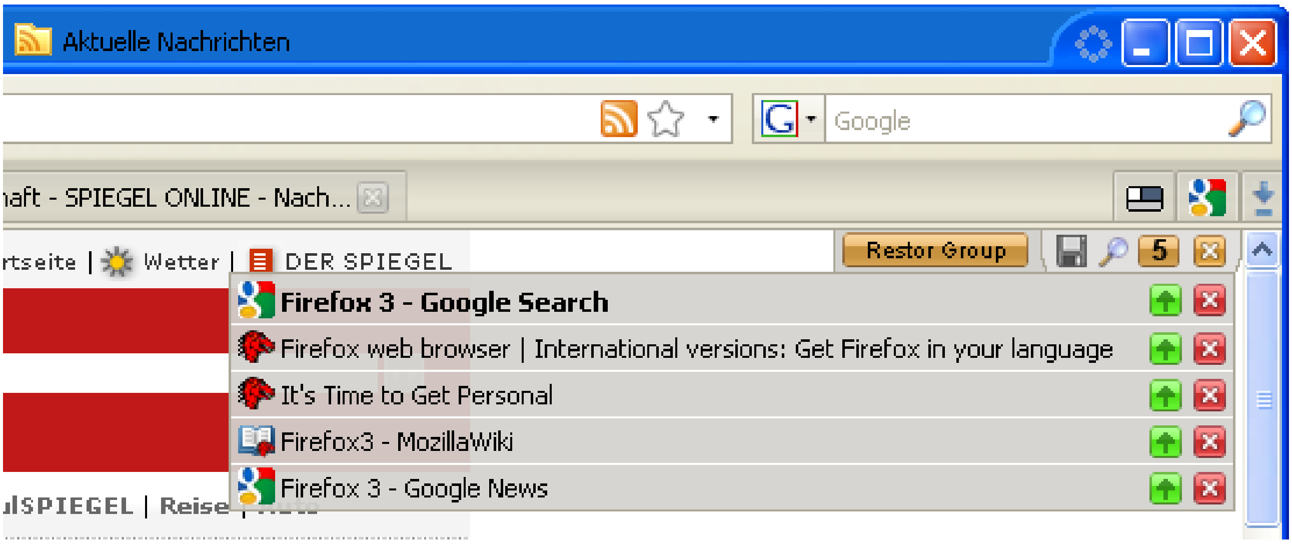
As we all know, you can not only create tabs, but you can close them too. So how would this kind of tree structure work when we close tabs, or better, how would it work if we closed tabs that are important for multiple Tab Tree Groups? If a tab gets closed, the lower structure will become attached to the upper one. The same principle applies to a group that no longer fulfills the Tab Tree Group criteria, it would be integrated into the next higher group. A special case is when a branching point/tab gets closed (marked with an 'x' on the pic.) and two groups become related to one tab: which group should be allowed to integrate it?





The answer to the question is that the tab further belongs to the group it was last belonging to. This prevents tabs from always jumping between different branch groups.

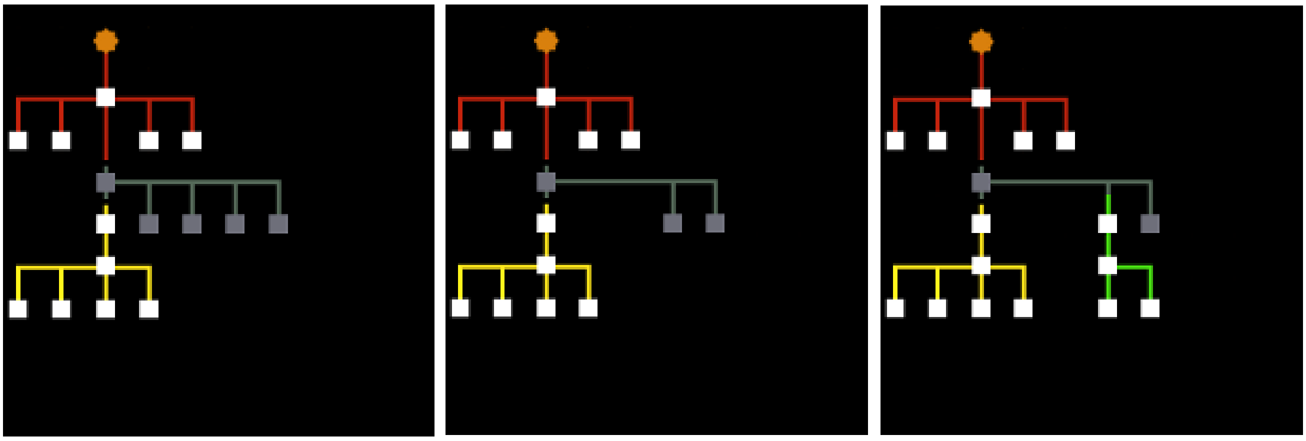
Until now, it was not important for the user to know whether a tab belonged to one or to another Tab Tree Group and if a group was created or closed, because all tabs were still open according to my concept. It’s only when you minimize a Tab Tree Group that the tabs they contain become important: we could say that they are stored in another place and no longer in the big “tab pool”. Take a look at the next picture that shows a Tab Tree Group with one click on it – actually, you see two Tab Tree Groups.



Aktuelle Nachrichten
Google
aft - SPIEGEL ONLINE - Nach...
rtseite | Wetter | DER SPIEGEL
Restor Group
5
lSPIEGEL | Reise

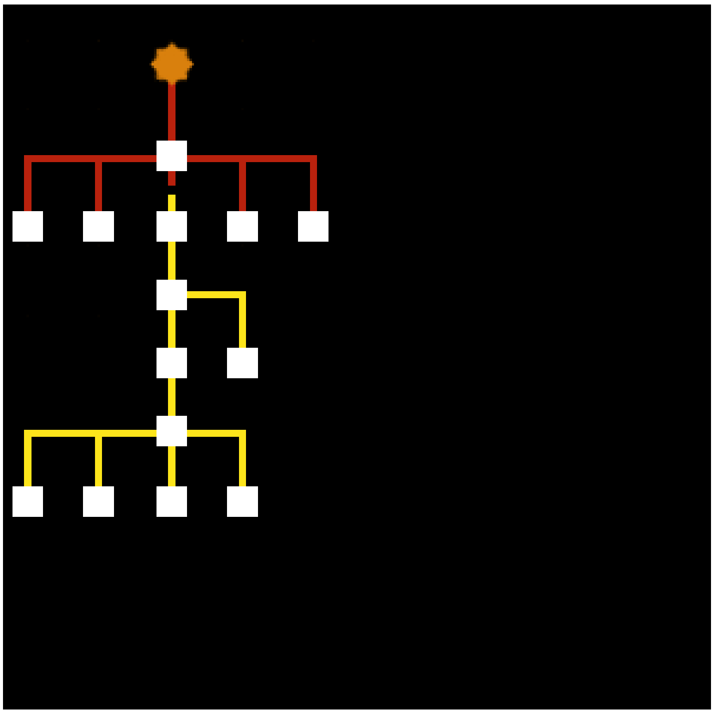
Firefox 3 - Google Search
Firefox web browser | International versions: Get Firefox in your language
It's Time to Get Personal
Firefox3 - MozillaWiki
Firefox 3 - Google News

It would be very unpleasant if you were to close a few tabs within a minimized Tab Tree Group and the whole Group would suddenly disappear – because it would no longer comply with the rule – and would become linked to another group. To prevent such an occurrence, I suggest freezing the minimized tab structure. The user could then close individual tabs without having the minimized Tab Tree Group integrated somewhere else. If the user chose to reactivate only one single tab from a Tab Tree Group, this tab would “de-freeze” and the standard rules would apply again.



When you save a Tab Tree Group, you also save the underlying tree structure. Because the user can add further tabs to a Group that has already been saved, I suggest to integrate it directly below the starting point (star). In this way, the existing group structures would not be affected.

Positive sides of this method:
It enables the user to open many websites and to bundle them into a few well-structured groups that clearly show the user how he has arrived there. I'll give you a quick example: let us suppose that the following picture represents a Google search. The first website would be the result of the search. Different search results would follow as open tabs. One of these tabs would be a Wikipedia entry from which we open further links. As long as we don't open more than two tabs starting from the same website, the whole structure belongs to the "Google Group" (red). As soon as the user opens more than two tabs starting from the same website, a new Tab Tree Group appears (yellow). The new Tab Tree Group holds all tree tabs up to the Wikipedia entry (also included). It is not so relevant to the user to know how the second Tab Tree Group was created. It is more important for him to see in the second Tab Tree Group where he has come from, i.e. from Wikipedia.



Negative sides of this method:
In most instances, the user will not understand why a new Tab Tree Group was created – I don't see it as a real problem though, since we have reached our target of bundling tabs. Only a field test can show how practical my method really is.

I would like to add that I find automatic tap grouping solutions far better than those that are left to the user's discretion. Another basic rule should be to keep the number of clicks to a minimum and mouse movements as short as possible.

That's about it! I'd be really happy to get your feedback and to know what strengths/weaknesses you see in the suggested method.

Regards
Paradiesstaub

http://justoneidea.wordpress.com