Enhanced Window Manager Support for Large Monitors

CS 465: Project Proposal
Steve Bezek (sbezek2@illinois.edu)
September 7th, 2008

Motivation for Project

This summer I interned at Google where each developer uses a 30" monitor. Despite the large monitor, my productivity was not much higher than my home set up consisting of a laptop and a 19" monitor.

At home I typically have two windows simultaneously opened, one maximized on each screen. If I want to work on a different window I just drag it to the screen I want, then click maximize. At Google I would have to carefully move the window to the correct position on the monitor, then carefully resize. This probably took about 10x longer than the simple operation at home.

In this project I am to make window management more efficient for large monitors. This issue will grow in importance in the future as monitors continue to get larger.

Audience for Project

The primary audience for this project will be software developers using large monitors. I focus on software developers because I know the audience well. Also my project will be using Linux and a far greater proportion of software developers use Linux than the general public.

Description of Project

My project will be enhanced large monitor support for the GNOME/Metacity. To do this I will add some aspects of tiling window managers into GNOME/Metacity. I choose GNOME/Metacity since it is the most popular desktop for Linux, and because I have used it at home and at internships.

It would appear that tiling window managers solve the problem. However current tiling window managers are hard to use. Many companies use GNOME by default and don't let employees switch window managers. Additionally by allowing both tiling and overlapping, users have more flexibility. Also since many people are familiar with GNOME, the learning curve for the enhancements will be very shallow as opposed to a learning a brand new window manager.

I will investigate, prototype, and build one or more of the following ideas:
1. Using keyboard shortcuts to automatically move and resize windows. This idea is similar to the program WinSplit Revolution for Windows. Using this idea you would press alt-numpad9 to make the current window take up the top-right quarter of the screen.

2. Another option is to divide a physical monitor into a number of different virtual monitors. Then when you move a window into one of the virtual monitor areas and click maximize, it will only maximize to the size of the virtual window.

3. When resizing windows, make the windows snap to a 75px grid. Rarely do people need more finely grained control and this option requires the least change in the behavior of the user.

4. Add an additional button to the left of the "Minimize", "Maximize", and "Close" buttons. When the button is clicked, it will bring up a dialog box of different screen layouts: columns, 2x3 grid, etc. The user will click on the layout, then click on the specific location within that layout and the window will automatically be moved and resized.
The picture above shows the location of the additional button. When the button is clicked it brings up a menu of 4 different window arrangements. The user then clicks within a specific rectangle within one of those arrangements and the current window is resized to that location.

5. Change the docking behavior so if a window is brought against a side of the monitor instead of just making it stick to the side, it would resize it to either the whole column or quarter of the monitor. I can't think of any good ways for this to work for having more than 4 visible windows.

6. When moving a window, a small box showing different window configurations pops up in the middle of the screen. To have the window resized to one of the configurations, drag and drop the window on that configuration.

**Underlying Metaphor**

The beauty of my system is that because it is built on GNOME, the desktop metaphor is still in effect. Users can still drag and overlap windows. The additional features will mainly add quick pre-sets for windows. My project would be analogous to preset buttons on radios: people can still tune the radio like they were used to and get what they want, but with presets they can get there much faster.

**Potential Challenges**

I don't have much experience working with open source Linux code. The GNOME/Metacity code may be hard to understand and poorly documented. If that is the case I would spend the majority of my time trying to understand the code instead of working on user interface principles.
Other Progress

I've downloaded Metacity and started looking through the source code. I will post my proposal on the Metacity mailing list (http://mail.gnome.org/archives/metacity-devel-list/) and solicit feedback from the maintainer to see if this is a direction they want to go, or if this code belongs in a different package.

After reading more about Metacity, it might be the case that my project would be better suited to a different window manager such as Enlightenment. Metacity is described as a "Boring window manager for the adult in you", and may avoid new ideas.