Improved GUI concepts for hackers
13 January 2005

Anselm R. Garbe

garbeam at gmail dot com

Abstract

This paper discusses new trends of improved GUI concepts with hackers and keyboard users in mind. It presents the Window Manager Improved (WMII)[1] for the X Window System[2] as solution for most requirements of improved GUI concepts beside some other unconventional window managers. It also discusses some interesting relations to the Acme[3] user interface for programmers of the plan9[4] operating system.

1. Introduction

Classical widespread graphical user interfaces (GUI) have been hardly improved for years, especially for hackers. The mouse became the most frequently used input device for GUI interaction in the last decades, though its handling has resulted to be inefficient for most applications compared to keyboard handling. Basically beginner users benefit from the mouse handling of today's graphical user interfaces like MS Windows[5], Apple Mac OS X[6], the K Desktop Environment (KDE)[7] or the Gnome Desktop Environment[8], whereas hackers have met with no response until now. Thus the basic window and widget handling of classical GUIs has become very mouse oriented apart from more intelligent rudiments, which don't require special input devices except the keyboard.

---

1 http://wmi.modprobe.de
2 http://www.x.org
3 Acme: A User Interface for Programmers by Rob Pike, *A system with a more radical approach to programming and editing.* http://plan9.bell-labs.com/sys/doc/acme/acme.pdf
4 http://cm.bell-labs.com/plan9dist/
5 http://www.microsoft.com
6 http://www.apple.com
7 http://www.kde.org
8 http://www.gnome.org

2. Terminology

The term 'hacker' is meant for shell enthusiasts, Unix geeks, software developers, system administrators, security specialists, computer scientists or whatever people with deep knowledge about computers in general. The opposite term 'beginner users' aims at users with no deep knowledge about computers in general.

3. Indispositions of hackers

Hackers don't want GUIs which are only usable with the mouse. They don't settle for shortcuts until the mouse gets optional. They don't want wasting space on their desktop for icons, trash, menus, colorization overkill or tons of loose windows lying around. They don't want resize and move their terminal windows (and other windows in general) permanently. Hackers don't want desktop environments which took up minutes for starting. Last but not least they don't want desktop environments which feed at least the half of physically memory.

4. Affections of hackers

Hackers want GUIs which are *usable* without the mouse. They love clean desktops with nice wallpapers and translucent terminals. Hackers want desktop environments which start within a second and take only a hundredth of physically memory. Hackers want desktop environments which behaves like their favorit editor or shell. To say it in short words: hackers want window managers which are improved.

5. Background

Apart from classical GUIs, which seem to fulfill all aspects of the shown indispositions, there exists several window managers for the X Window System which come into account to support the affections. Most famous window managers of them are Ion[9] and LarsWM[10].

---

9 http://modeemi.cs.tut.fi/ tuomov/ion/

5.1. Ion

Ion is developed by Tuomo Valkonen and pretty mature. Its predecessor PWM, which was also developed by Tuomo Valkonen, is the first known 'tabbed window manager' to me, which arranges multiple clients in one[11] frame. The concept has been integrated into the Ion window manager. But the main improvement of Ion is the non-overlapping - so called 'tiled'[12] - arrangement of frames over the desktop. Such non-overlapping arrangement is particularly appropriate for terminal windows. This results in a clean and tidy desktop.

The only drawback of Ion has been for a long time that its non-overlapping concept was too restrictive for floating arrangements of windows. In the meantime Tuomo Valkonen developed a new version of Ion which supports virtual workspaces for floating window arrangement similiar to PWM behaved.

5.2. LarsWM

When Ion didn't support the arrangement of floating windows, Lars Bernhardsson forked[13] the 9wm window manager, originally developed by David Hogan, into the LarsWM.

LarsWM, which is a really leight-weight but pretty functional one, has been able to close the gap between a floating window arrangement and a non-overlapping window arrangement. LarsWM segmented all virtual workspaces into two layers - first layer for floating window management and second layer for so called 'tiled' window management.

But LarsWM has the drawback, that it don't support the arrangement of multiple clients in one frame. As workaround LarsWM provides an additional partitioning of the 'tiled' layer into two - so called - 'tracks'[14]. The left track contains only one window (the active client) per time. The right track contains all other windows in an non-overlapping and vertically tiled order.

To work with clients in the right track they were swapped with the active client in the left track.

LarsWM organizes windows very space-saving, because it abandons title bars and thick borders. As an alternative it provides a status bar.

5.3. Similarities

Both, Ion and LarsWM, are primarily designed with keyboard users in mind and provide a very efficient window handling. Both are highly customizable and pretty stable.

6. Window Manager Improved

In 2001 I began to specify a vision of my own window manager for the X Window System which was based on the consideration of advantages and drawbacks I noticed in Ion and LarsWM. Two years later, in October 2003 I began the development of the Window Manager Improved (WMI), which is based on the ideas I got over the years.

The WMI is an attempt to combine the advantages of Ion and LarsWM into one window manager. Additionally the basic user interaction interface is pretty much oriented on the two modes of the widespread *vi* [15] editor, which I have been using for years.

---

10 http://www.fnurt.net/larswm/

11 There exist other concepts which draw a smaller title bar in horizontal expansion like BeOS does. But these concepts don't arrange multiple clients in *one* frame.

12 The first tiled window manager under X11 known to me was the rtl tiled window manager by CMU, also known as Siemen's window manager, but I don't know if the famous company or a surname is meant.

13 As I know, he forked 9wm independendly from knowing about the existance of Ion.

14 A track is a sublayer which has the same height as the screen, but only some proportion of the screen's width.

---

15 Visit http://www.vim.org to get the most improved version of vi.

6.1. Requirements

The WMI achieves following requirements which are compared with Ion and LarsWM:

	Ion	LarsWM	WMI
multihead support	Yes	Yes	Yes
virtual workspaces	Yes	Yes	Yes
track support	No	Yes	Yes
tiled arrangement	Yes	Yes	Yes
floating arrangement	Yes	Yes	Yes
tabbed windows	Yes	No	Yes
status bar	No	Yes	Yes
builtin pager	No	poor	Yes
menu support	Yes	poor	Yes
dockapp support	Yes	No	Yes
borderless support	No	Yes	Yes
fully customizable	Yes	Yes	Yes
interactive config	Yes	No	Yes
session management	Yes	No	Yes
leight-weight	Yes	Yes	Yes
keyboard usage	Yes	Yes	Yes
mouse usage	uc*	uc*	Yes
object oriented	No	No	Yes
specific dependencies	Lua	No	No
license	LGPL	9wm	MIT
programming language	C	C	C++

(* unconventional)

As you can see, WMI supports all advantages of Ion and LarsWM.

6.2. Window concept

To supply overlapping and non-overlapping window arrangement and multiple clients within one frame or group (tabbing feature), the window concept of the WMI seems to be very complex. Surprisingly this is not the case, because the WMI window management is divided into so called *frames* which arrange maximized clients - and into an arbitrary number of groups, which arrange floating clients.

Frames are non-overlapping containers - a mixture of LarsWM tracks and Ion frames. Each client which is attached to a frame is maximized. If frames are arranged as neighbors (splitted), the track behavior of LarsWM can be emulated, because the WMI provides methods to swap clients between neighbored frames.

Groups are virtual containers which group an arbritrary number of floating clients on a subworkspace which is similiar to the LarsWM layer concept. There is no limit in the amount of groups, which is a difference to LarsWM layers, which handles only two default layers.

The window concept can be handled either with the mouse or the keyboard or vice versa.

The non-overlapping arragement of windows or frames, which is one main aspect of improved GUI concepts, decreases drastically the mouse usage for window adjustment and thus increases drastically the efficiency of user interaction.

6.3. Vi oriented interaction concept

For efficiency, consistency and space-saving reasons the WMI provides an interactive keyboard driven menu, which can be used to invoke 'actions'[16] (e.g. growing the frame, launching a terminal) flawessly in a manner, as the *command mode* of vi behaves. This menu is called 'input mode' in the WMI and supports automatic command completion like you know it from the shell. It also supports, like ion does, autocompletion for executing external programs (e.g. firefox). The size of the menu is fixed as one-line input bar, which has the same size as the status bar. This facilitates to see all clients while invoking 'actions'. To save maximum screen space the status bar can be hidden optionally.

6.4. Actions and configuration

Actions are the internal commands which provide the interface for the internal functionality of the WMI to the user. Each internal action can be bind to a shortcut arbitrarily for a snappy invocation. Key bindings and session configuration can be made interactively, all other specific configuration is done via editing configuration files.

Internal actions can be extended with aliases for external programs or with alias for sequential invocation of defined actions. Aliases are names for so called 'extern' or 'chain' actions, which behave pretty similiar to internal actions and can be also bind to arbitrary shortcuts. The differentiation is for consistency reasons.

6.5. Other concepts

To provide status information, a remote tool for setting status text of the WMI has been developed. This is pretty similiar to the LarsWM tool 'larsremote'.

Last but not least for dock app or gkrellm support a slot has been invented, which behaves very similiar to the NEXTish WindowMaker[17] dock. This mixes up a tidy desktop with nice status tools.

7. Solution

As a combination of all advantages of Ion and LarsWM the WMI implements the largest portion of requirements to

---

16 In WMI internal commands are called 'actions'.

17 http://www.windowmaker.org

satisfy all affections of a hacker to a felicitous window manager and GUI. The WMI is a window manager with a mixture of an editor, a shell and intelligent window management concepts.

Currently the WMI is under development and still not as stable as Ion or LarsWM, but the day is forseeable when WMI will become a widely used window manager for hackers, because it provides the most advantages they are looking for.

8. Credits

Following people provided useful feedback or several grammar fixes to this paper:

- Frank Boehme
- Tuncer M zayamut Ayaz
- et al.