GUI development for GRASS GIS

Martin Landa
Department of Geodesy and Cartography
Faculty of Civil Engineering, Czech Technical University in Prague
and
FBK-irst, Trento, Italy
landa.martin gmail.com

Keywords: GIS, GRASS, GUI, wxPython, development

Abstract

This article discusses GUI development for GRASS GIS. Sophisticated native GUI for GRASS is one of the key points (besides the new 2D/3D raster library, vector architecture improvements, etc.) for the future development of GRASS. In 2006 the GRASS development team decided to start working on the new generation of GUI instead of improving the current GUI based on Tcl/Tk.

Motivation and background

GRASS GIS has been under continuous development since 1982 especially by the U.S. Army Construction Engineering Research Laboratories (USA-CERL) a branch of the US Army Corp of Engineers, as a tool for land management and environmental planning by the military. In 1995 USA-CERL decided to leave GRASS project, the first GPL'ed GRASS GIS has been released in 1999 as GRASS 5. In the result GRASS GIS is actively developed for more than 25 years. GRASS as a piece of software with long tradition is closely linked to Unix environment.

Almost all GRASS modules are originally CLI-oriented (Command Line Interface). This can be advantage for developers or power users. The real capabilities of GRASS are basically connected to the power of CLI. The GRASS users request intuitive, simple graphical user interface. In GRASS world, GUI is an alternative way how to use the system, the main advantage for the users is powerful CLI. Anyway GUI is very important point for the most of GRASS users, requested especially by newcomers. Beside of that, there are tools like Georectifier, Digitization tool, etc. which are basically mouse-driven. These tools should be fully integrated into GUI structure.

The GRASS GUI History

The first native GUI for GRASS was originally developed by Jacques Bouchard in 1999 (based on Tcl/Tk) and became part of GRASS 5, so called TCLTKGRASS (fig. 1).

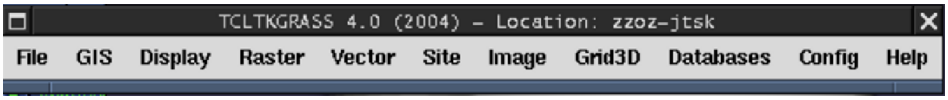

Fig. 1: TCLTKGRASS available in GRASS 5.0

Later has been developed still Tcl/Tk-based replacement of TCLTKGRASS called **d.m** ("Display Manager") by Michael Barton, Radim Blazek and others (fig. 2). This component allowed the users to run GRASS modules from menu and graphically manage the (old) GRASS monitors (based on X11 driver).

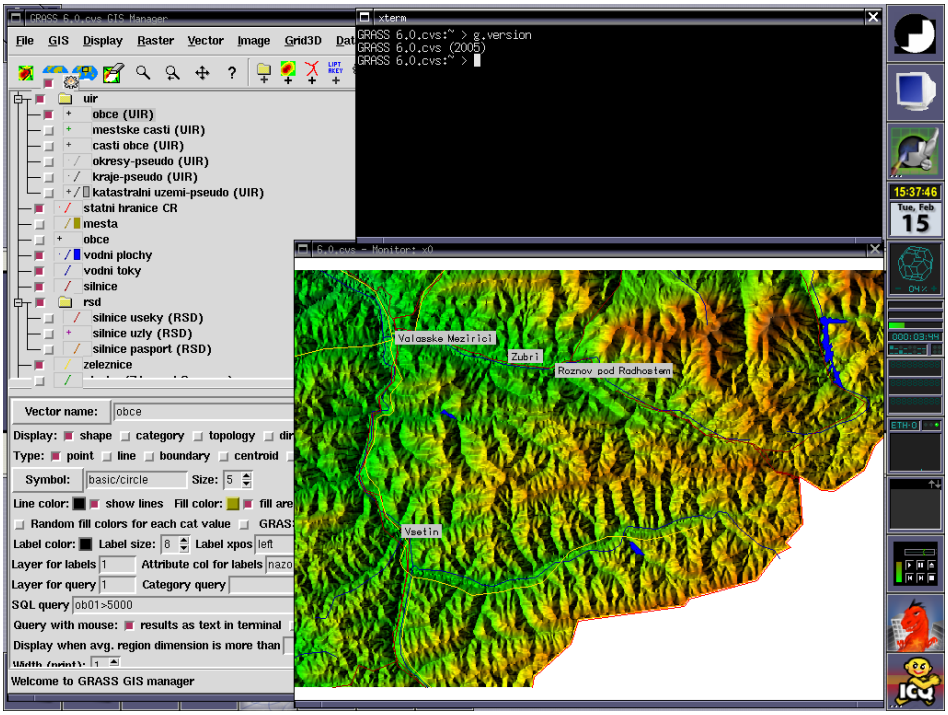

Fig. 2: Display Manager (d.m) in GRASS 6.0

In 2006 Michael Barton decided to develop **gis.m** ("GIS Manager") as a replacement of old d.m. In the result, gis.m has been released for GRASS 6.2 (fig. 3). Display architecture, module menu, graphical module dialogs (which are generated on-the-fly by the GRASS parser) have been completely rewritten, new output window and layer management introduced. Moreover new external tools have been integrated, e.g. georeferencing tool (replacement of X11 driver based on i.points module). Also new map display window with integrated basic tools like zooming, panning, data querying has been implemented (as a replacement of X11 driver-based GRASS monitors).

Similarly the digitization tool (module v.digit) has been originally CLI-based. GRASS 6

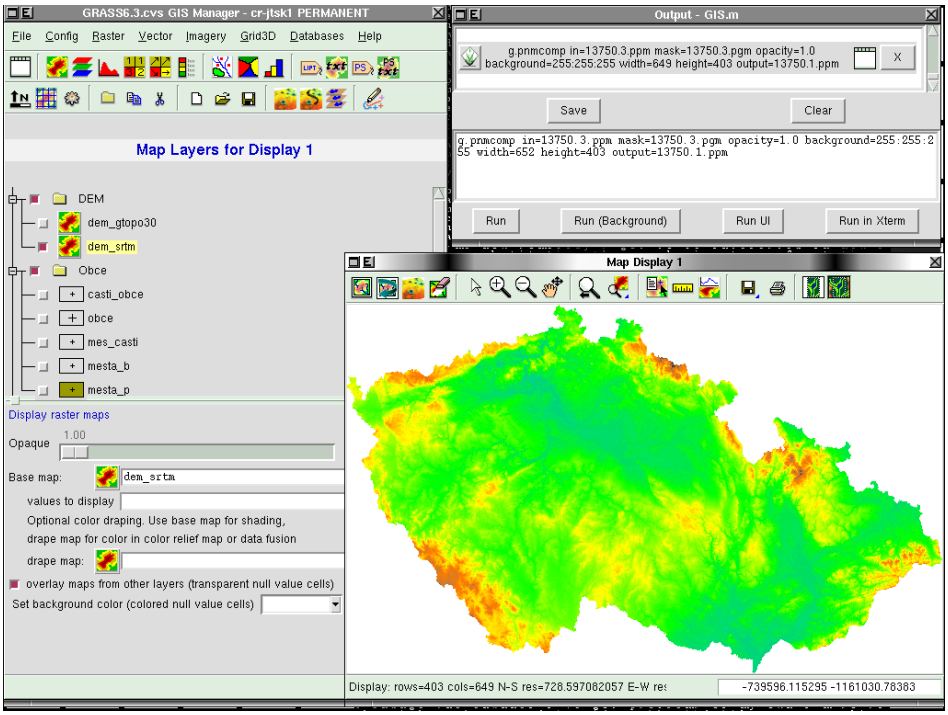

Fig. 3: GIS Manager (gis.m) in GRASS 6.3

brought GUI-based completely rewritten v.digit (fig. 4).

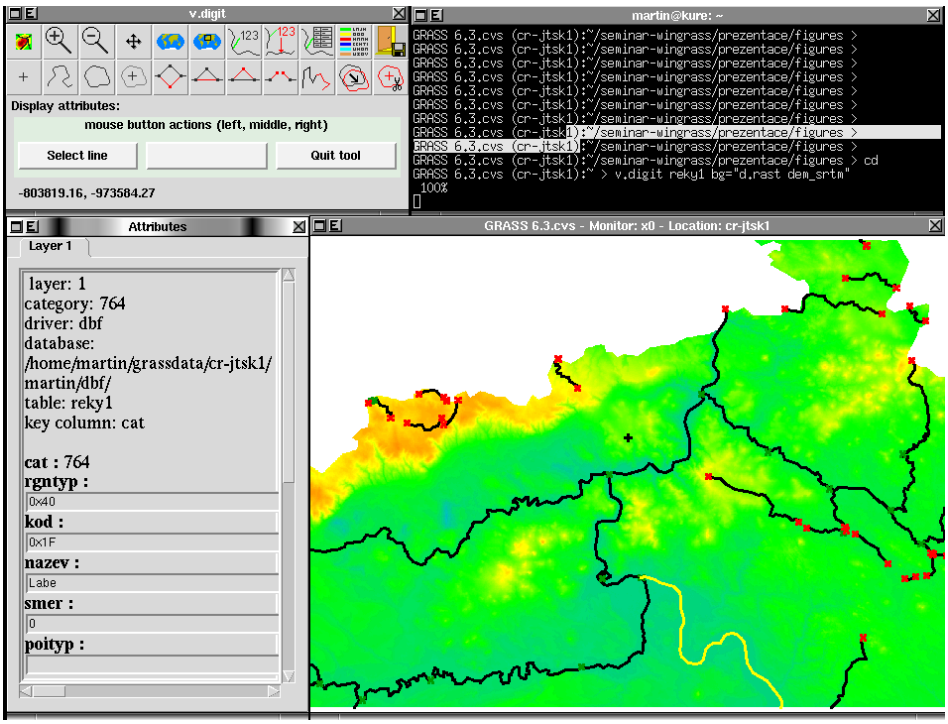

Fig 4.: Digitization tool v.digit from CLI-driven to GUI-driven user interface

The last major part represents NVIZ, a tool for 3D visualization also written in Tcl/Tk

(fig. 5).

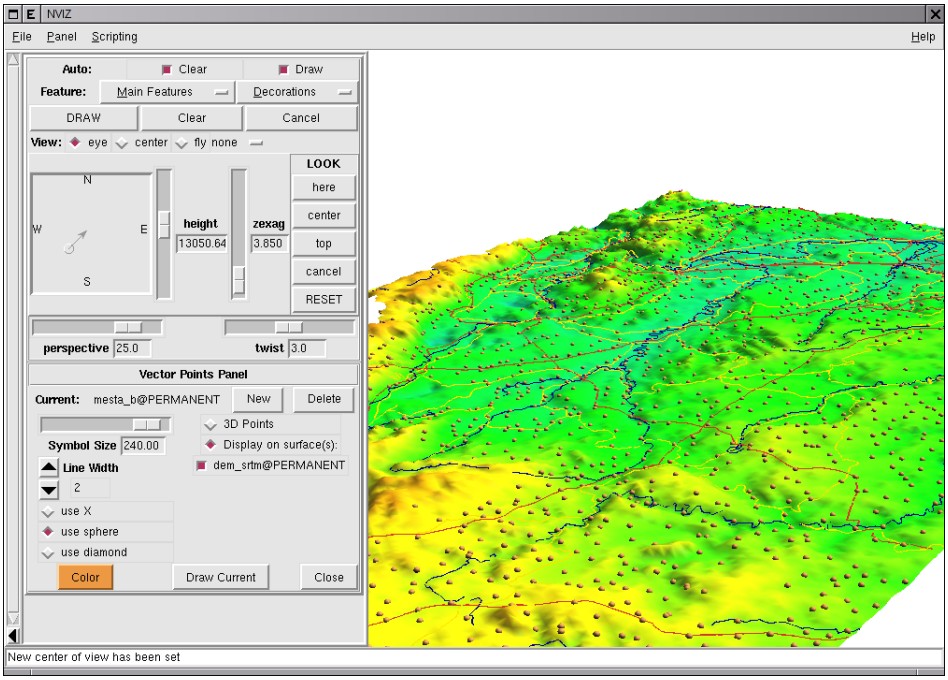

Fig 5.: NVIZ as a tool for 3D visualization

Discussion about future development

The native GUI for GRASS was originally developed using Tcl programming language and Tk graphical toolkit (including first version called TCLTKGRASS or the latest GIS Manager, NVIZ or user interface for v.digit module). Besides the native GUI also a few alternative GUIs for GRASS are available, mainly JGRASS written in Java or QTGRASS (QT library).

During the time, the limitations of Tcl/Tk toolkit appeared to be fundamental for the future development. This issue has been several times discussed in the GRASS developer mailing list. At the end has been decided to leave Tcl/Tk and to design new native GRASS GUI from the scratch using another graphical toolkit.

There was question which graphical toolkit to choose. The major players are QT, GTK and wxWidgets. The main requirements:

- portability,
- native look and feel,
- OpenGL widget available,
- powerful Python binding,
- performance and flexibility.

At the end wxWidgets toolkit has been chosen, the major reasons:

- Python binding (know as wxPython).

- Uses native platform SDK (native look and feel), it means that a program compiled on Windows will have the look and feel of a Windows program, and when compiled on a Linux machine, it will get the look and feel of a Linux program.
- Free OpenGL widget (in contrast to QT) which is requested for the future development (NVIZ replacement).

For real development has been chosen Python binding of wxWidgets which is known as wxPython. The main reason is more or less simple, Python as "easy-to-learn", object oriented, currently "very popular" language should enable more people actively contribute on the development in contrast to wxWidgets which works for C++ (or Tcl/Tk used for the old GUI). Moreover wxPython is currently actively developed and maintained. The decision is strategic for the future development.

Development of wxPython-based GUI

Development of GUI using wxPython started in the end of 2006.

Basic information

The newly developed GUI follows more or less design of the latest Tcl/Tk-based GUI which is available in GRASS 6. The GUI is composed by *Layer Manager* (fig. 6) which enables the user to run different GRASS modules (generated on-the-fly based on XML output of the GRASS parser) from menu, layer management for map display windows, integrated command-line, etc.

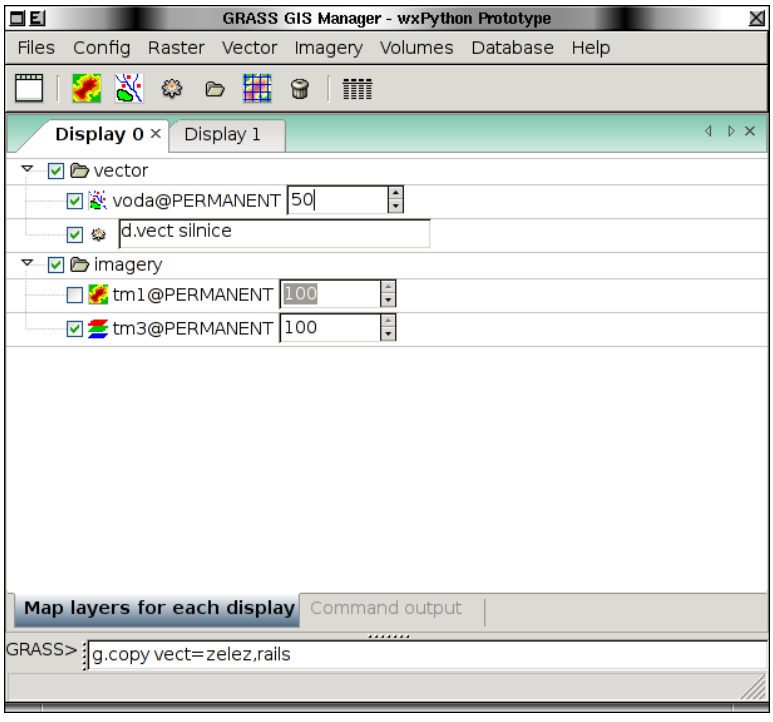

Fig. 6: wxPython-based Layer Manager

The second main component is *Map display window* which integrates basic tools for zooming, panning, data querying, decorations (north arrows, barscale, etc.). This component replaces the old X11-based GRASS monitors. The user is allowed to start various map display windows during one session. Layer Manager registers the all started map display windows using different tabs.

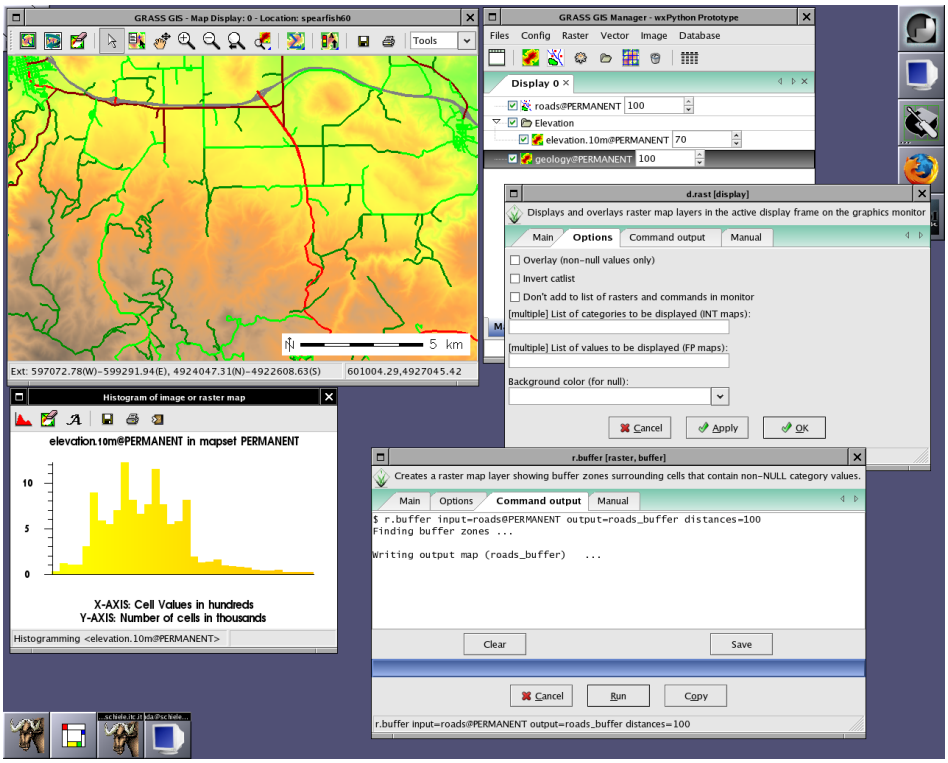

Fig. 7: Layer Manager and several Map display windows in one session

Each map display window has different toolbars, by default only "Main" toolbar is enabled. Currently is available special toolbar for Digitization tool, under active development is also Georectifier tool.

The future development

One of the key components of GUI – Digitization tool – is currently under active development. This tool is fully integrated into GUI structure and should replace current Tcl/Tk-based v.digit module. Almost all features available in v.digit has been implemented in the new Digitization tool. Some additional features are planned to be implemented, e.g. marking isolines (available in old CLI-oriented v.digit in GRASS 5.0, never re-implemented in v.digit), snapping to vertex, background vector map layer objects, etc. This requires also modification of vector library especially to enable partial (based on the bounding box) building of topology for modified vector map layer.

Another important component – Georectifier tool – is currently being developed by Michael Barton. Many other components or sub-components have to implemented or improved including interactive command-line, module search engine (to find given GRASS module according

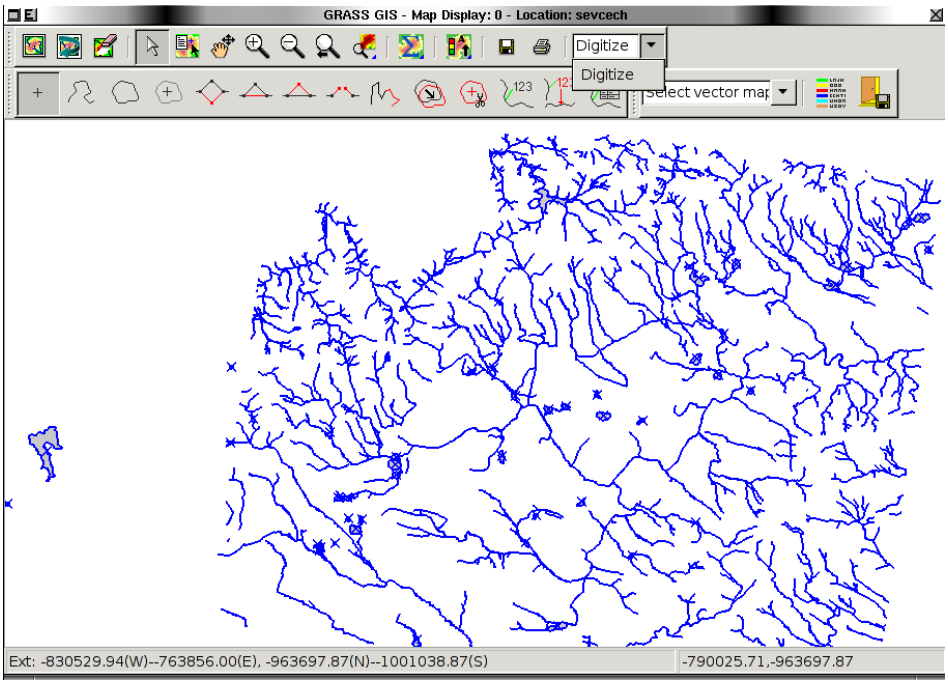

Fig. 8: Map display window with integrated different toolbars

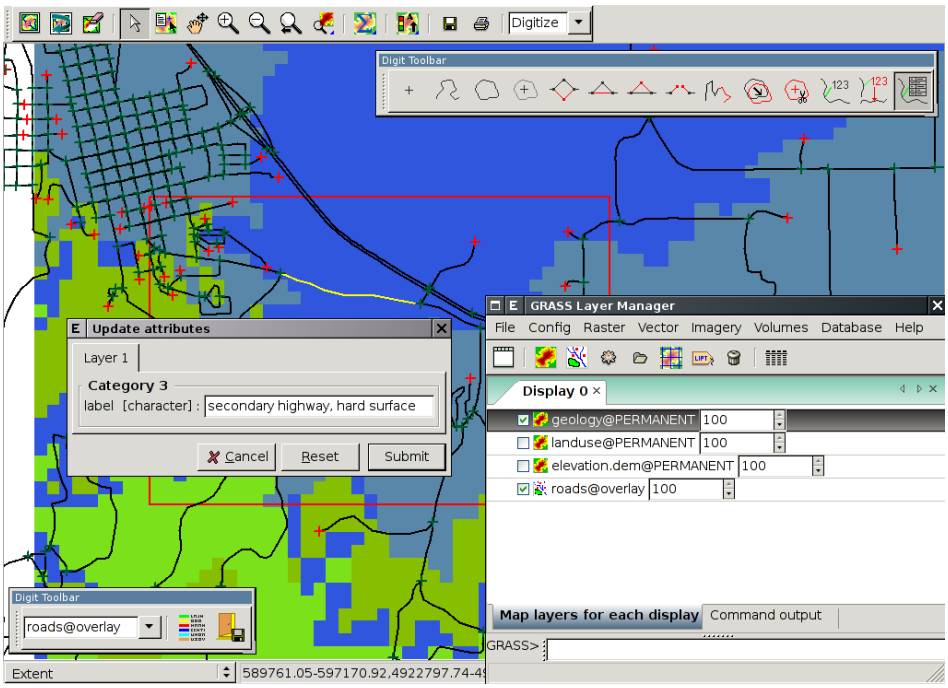

Fig. 9: Digitization tool in action / Uno

key words for a given user's task), message handling, etc.

Very complex and integral part is NVIZ re-implementation in wxPython. Functionality of NVIZ (3D visualization tool written in Tcl/Tk) is planned to integrated into the current components of wxPython GRASS GUI. Map display window currently enables only 2D visu-

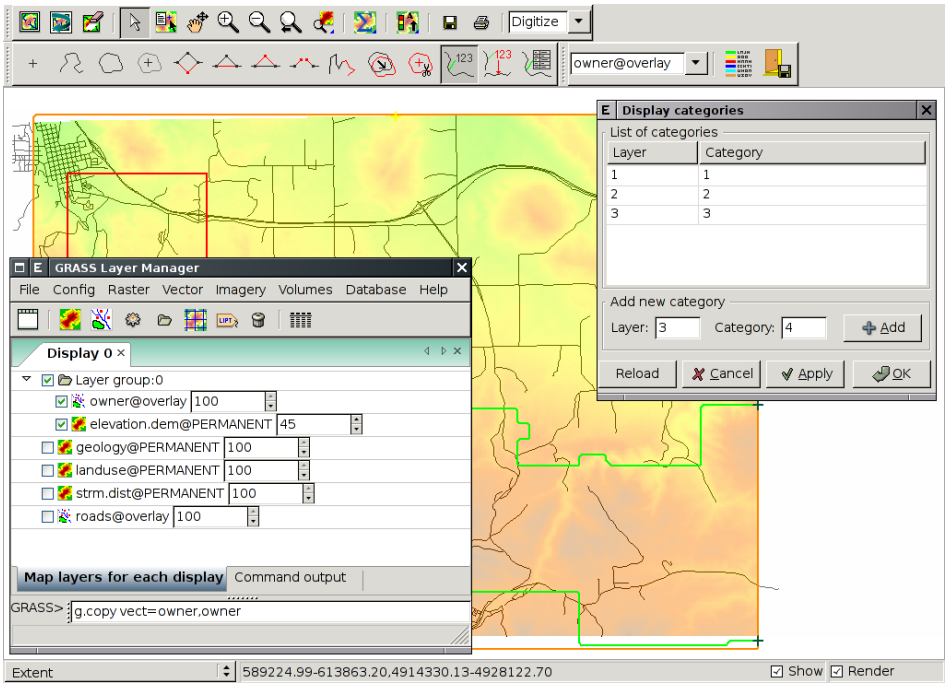

Fig. 10: Digitization tool in action / Due

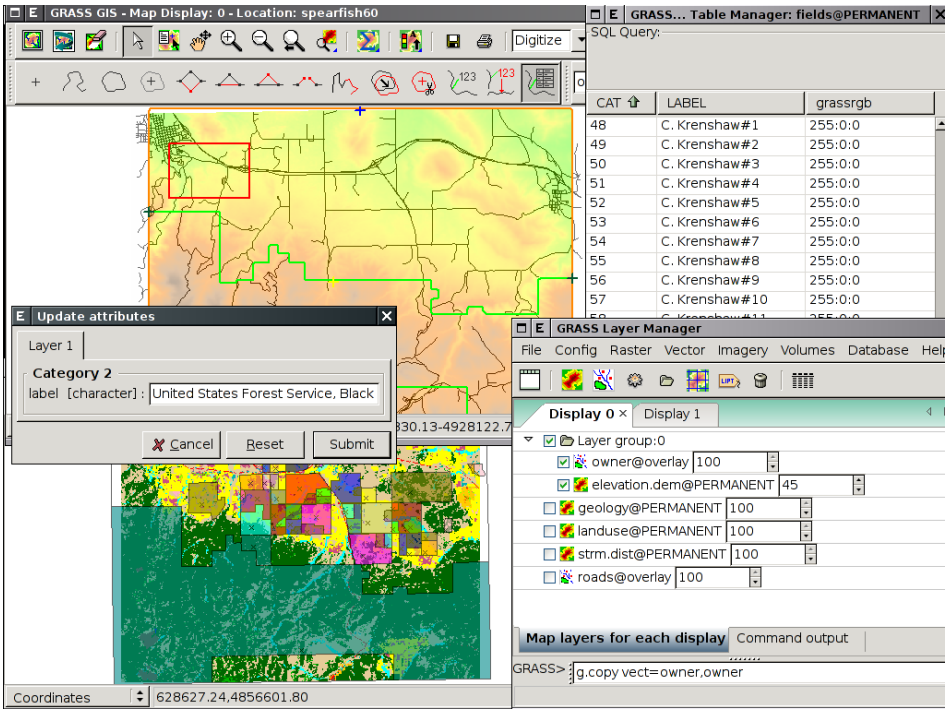

Fig. 11: Digitization tool in action / Tre

alization, in the future it will be extended towards 3D visualization skills. This fundamental feature will be implemented using OpenGL widget which is available in wxPython.

Conclusion

The decision to replace currently used Tcl/Tk with wxPython is crucial for the future GRASS GUI development. The experience with wxPython is mostly positive. wxPython is one of the most active and maintained bindings for wxWidgets toolkit. Moreover it supports more or less all features which are needed for the current development (including fundamental OpenGL widget) of GUI for GRASS.

WxPython-based GUI for GRASS is actively developed by several people, it seems that Python as a "easy-to-learn" programming language enables more people to actively contribute on the development process.

The GRASS development team is currently preparing new development branch for GRASS 7, wxPython-based GUI will be part of GRASS 7 as the default graphical user interface. Current speed of development is promising for the future.

Sophisticated GUI is crucial for GRASS user politics especially connected to the newcomers or non-power users who essentially request GUI. The GUI need to reflect the specific needs of GRASS users, need to be intuitive, simple and in the certain point "minimal".

References

1. Markus Neteler and Helena Mitasova: Open Source GIS: A GRASS GIS Approach.Third Edition., Springer, New York, ISBN 978-0-387-35767-6, http://www.grassbook.org
2. Noel Rappin and Robin Dunn: wxPython in Action, Manning Publications, ISBN 978-1932394627
3. Julian Smart, Kevin Hock and Stefan Csomor: Cross-Platform GUI Programming with wxWidgets, Prentice Hall PTR, ISBN 978-0131473812
4. GRASS GIS – http://grass.itc.it
5. wxPython – http://wxpython.org
6. wxWidgets – http://www.wxwidgets.org
7. PyOpenGL project – http://PyOpenGL.sourceforge.net