Advances in Free Software Geographic Information Systems

Markus Neteler[1] and Venkatesh Raghavan[2]

[1]ITC-irst-MPBA/SSI Via Sommarive, 18 .3050 Povo (Trento), Italy. e-mail: neteler@itc.it

[2]Graduate School for Creative Cities, Osaka City University, Japan.

e-mail: raghavan@media.osaka-cu.ac.jp

ABSTRACT: This paper reports on recent advances in FOSS4G (Free and Open Source Software for Geoinformatics). Features of the new GRASS 6.2.0 release, a major software package among the FOSS4G product family, are summarised in this paper. The new capabilities of GRASS 6 include a new engine for spatial vector data processing as well as an improved user interface with internationalisation efforts and multi-byte font support.

1.INTRODUCTION

Spatial data are crucial for national and local development planning, disaster management and early warning, and environmental monitoring. Compared to other IT sectors, there are specific factors:

- high cost of data acquisition,
- high integration among components, thus a need for data interchange.

This has so far promoted the use of a narrow range of software products, mostly proprietary. However, the benefits of Internet-based technologies, satellite and aerial data sources, massive storage capacities and wireless data transfer are becoming fully operational. Over the last decade, Free and Open Source Software (FOSS) has been well grown and reached a major audience (Raghavan et al., 2003). The capacity building within the FOSS development community lead to numerous packages implementing Geoinformatics technologies such as Geographic Information Systems (GIS) combined with relational databases and statistical languages. A GIS is a system of hardware, software, and data that facilitates the development, enhancement, modelling, and display of multivariate (e.g. multi-layered) spatially referenced data. A GIS can model (e.g. synthetically recreate) a feature or phenomenon as a function of other features or phenomena, which may be related - where all features or phenomena are represented (characterised) by spatial and related tabular data. This article introduces the recent efforts in a major FOSS GIS project.

2. GRASS GIS: Free Software GIS

The Geographic Resources Analysis Support System, commonly referred to as GRASS, is a Geographic Information System (GIS) used for geospatial data management and analysis, image processing, graphics/maps production, spatial modelling, and visualisation. GRASS is a major player among the FOSS for Geomatics systems and currently used in academic and commercial settings around the world, as well as by many governmental agencies and environmental consulting companies (Neteler and Mitasova 2004). It is released under the GNU General Public License (GPL). The developed of this GIS as public domain software was started in the early 1980s by the U.S. Army Corps of Engineers' Construction Engineering Research Laboratory (USA/CERL) in Champaign, Illinois (Westervelt, 2004). New responsibility for the environment encoded into the National Environmental Policy Act of the late 1970s and the lack of a GIS that satisfied the needs were driving the own development program for writing a hybrid raster-vector GIS. As CERL and GRASS evolved through the late 1980s and early 1990s, CERL created the Open GRASS Foundation which evolved into the Open GIS Consortium (OGC, nowadays the Open Geospatial Consortium), which is aiming for interoperability at the data and user interface level. Later, in 1996 CERL was formally withdrawing GRASS development. After a period of transition, in late 1997 a group at Baylor University, Texas, created a new Web site for GRASS. In 1998, a worldwide development team was established and the coordination moved over to University of Hanover, Germany. Since 2001 GRASS development is coordinated at ITC-irst, Italy, where development focuses on the GRASS 6 version. The software is available at http://grass.itc.it and at more than 20 international mirror sites. Estimated are more than 25,000 downloads per month worldwide. Quality management is enforced by transparent development methods such as a centralised source code repository (in a CVS server) and the immediate peer-review of source code changes by email broadcast. (Mitasova and Neteler, 2004). The GRASS software itself is organised is in the UNIX spirit of modularity. Small tools for the various tasks in a GIS are composing the entire system. Modules can be easily combined to powerful applications depending on the user's needs.

Figure 1: GRASS on iPAQ hand-held

2.1. Portability

While major GIS vendors focused on certain computer platforms only, the GRASS Development Team was heading into the opposite direction: full cross-platform portability. The same code base is supporting a range of operating systems (including 64bit processors). The long list comprises GNU/Linux (GRASS is now official part of several Linux distributions such as Mandrake, Fedora etc), Sun Solaris (SPARC/Intel), Silicon Graphics Irix, Mac OSX/Darwin, MS-Windows native and with Cygwin, HP-UX, DEC-Alpha, AIX, BSD, and even hand-held devices like iPAQ/Linux (Stankovic et al., 2002, see Fig. 1). Also the GRASS data format is portable, allowing a user to access a centralised database from various hardware. In the early days GRASS provided vast raster capabilities with some limited vector capabilities. Over the last few years the functionality has been heavily extended in various directions. The GRASS 5.0 series of releases in the 90th added floating point support for raster data. In the GRASS 5.3 series the built-in projection engine was replaced by the external improved PROJ4 library (http://proj.maptools.org), which added support for datum transformations. Re-projection of GPS and satellite-derived data into national coordinate systems usually require datum transformation.

2.2 Features of the new GRASS 6.2.0 release

Over the past few years the source code and the vector architecture have been reorganised. In October 2006, the new GRASS system was published at ITC-irst. Some of the improvements in GRASS 6 are detailed below

2.2.1 Data Import and Export

GRASS data are maintained in their own directory structure in so-called 'locations' and 'mapsets'. The idea behind this structure is to provide a multi-user environment with access control. Locations can be maintained on a centralised server. Extensive capabilities of data exchange are essential for the daily GIS work. GRASS profits from an external project, the GDAL/OGR library (http://www.gdal.org). This FOSS library is also used even by global data vendors as well as in some proprietary GIS applications. Many of the raster and vector formats supported for reading can be also be generated. GRASS is using a topological vector architecture. To support Simple Features vector data (conformal to Open Geospatial Consortium), these data sets are converted upon data import; conversely, export into Simple Features is also possible. Instead of importing data sets, vector maps can also linked into the GRASS database as virtual maps.

2.2.2 Visualisation and user interface

The integrated graphical user interface was re-designed for a more intuitive usage. Every module now generates its graphical user interface (GUI) on the fly. This keeps descriptions always up to date and permits for internationalisation of the system. NVIZ, the included visualisation package, was enhanced to display 3D vector data as well as 3D voxel volumes. Perspective views and fly-bys, even at a resolution larger than the monitor (off-screen rendering), can be generated. An alternate graphical user interface is provided through the Quantum GIS (QGIS, http://www.qgis.org) project. QGIS is a stand-alone geodata viewer with common look-andfeel. It also includes a digitising tool intended to replace the GRASS-internal digitiser. In the long run QGIS might become a standard graphical user interface with GRASS as workhorse. Fig. 2 shows QGIS with GRASS integration.

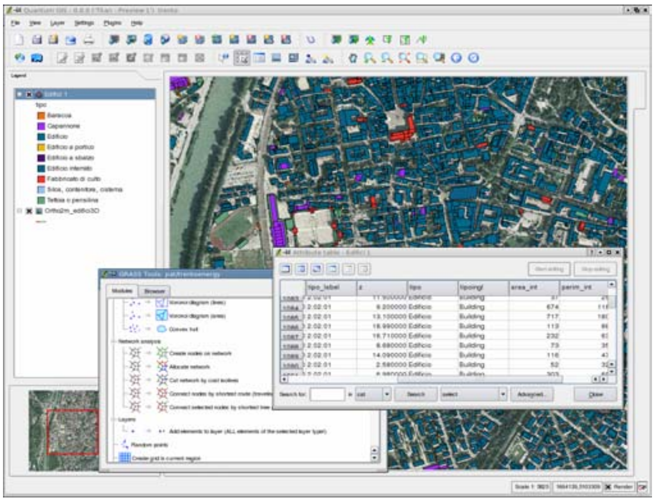

Figure 2: Quantum GIS with GRASS toolbox

Figure 3: Internationalised user iinterface: Vietnamese

2.2.3 Internationalisation and multi-byte font support

The framework to support translated GRASS user messages has been migrated from the GRASS 5.0.3 where it was implemented by the FOSS4G initiative (Masumoto et al., 2004). This approach was used and implemented slightly modified in GRASS 6. While in GRASS 5.0.3 all modules have a message catalogue file on a module basis, it was simplified to use only three catalogues, one catalogue for library messages, one for module messages and one for graphical user interface messages which are not covered by the other catalogues. The message handling is fully integrated into the new Makefile system which helps non-programmers to start new translation projects as messages can be automatically extracted and handed over to translators. Japanese and other Asian message translations have been partially migrated from GRASS 5 to the new system (see Fig. 3 for Vietnamese). The new Tcl/Tk based graphical user interface is autogenerated upon runtime. Due to this improvement developers do not have to separately program the graphical user interface any more. Also translations of the standard help texts will appear immediately in the new graphical user interface. The parameters/flags explanation section of the HTML documentation is also rendered from the standard help texts within a virtual session during compile-time of the software. Currently translations are ongoing for 16 languages. The efforts are coordinated in a dedicated mailing list(http://grass.itc.it/devel/i18n.php).Also multi-byte FreeType Font support has been migrated from the FOSS4G initiative efforts in 5.0.3. It allows to display FreeType/TrueType Fonts in the GRASS display system including multi-byte fonts for Asian characters (see Fig. 4).

2.2.4 New vector engine

In the last two years the vector engine has been completely overhauled. Geometry and attribute management are clearly separated, giving advantage to more flexibility in the data storage. The vector geometry was extended to manage 2D and 3D topological vector data. The new internal vector data format is portable across 32bit and 64bit platforms. A new spatial indexing system as well as category indexing accelerates data access. The support of topology enables the user to perform data cleanup to ensure data consistency. Support for vector map overlays, intersections and extraction of features is implemented. A new digitising tool permits to create or update vector features with attributes on screen. The attribute management includes full and flexible integration of database management systems (DBMS; currently supported are PostgreSQL, MySQL, DBF, SQLite and ODBC). SQL statements are used to manage attributes. Graphical updating of vector attributes has been implemented as well.

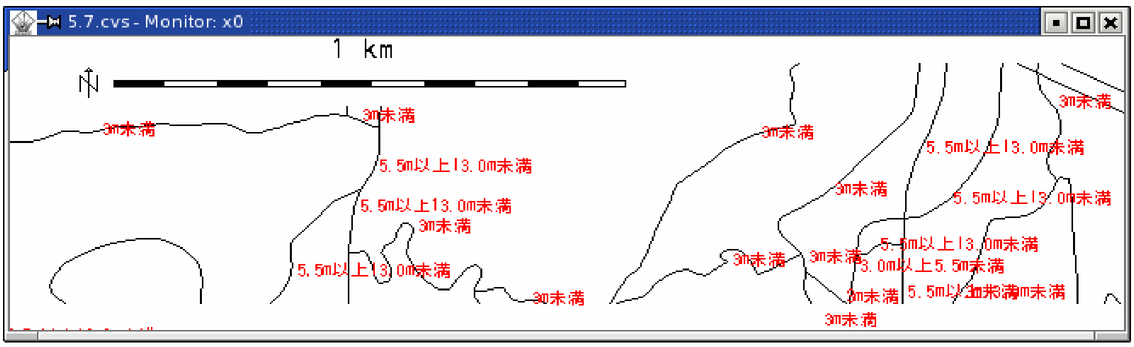

Figure4:Multi-byte font support for Asia fonts: Excerpt of "ojiya" sample data set

2.2.5 Vector network analysis

The new integrated Directed Graph Library provides support for vector network analysis. Available algorithms are shortest path (see Fig. 5), travelling salesman (round trip), allocation of sources (subnetworks), minimum Steiner trees (star-like connections), and isodistances (from centres). Costs may be assigned both to nodes and arcs. Both directions of a vector line can be used which permits to define a forward and a backward direction and to store their attributes into the related attribute table.

2.2.6 Voxel volume support

GRASS 6 offers support for 3D raster voxel volumes. Basic features currently cover import and export of 3D data, 3D map algebra and 3D volumes interpolation (IDW algorithm; RST regularised splines with tension algorithm). The visualisation of volumes has been integrated into NVIZ, the rendering tool of GRASS. The user can generate isosurfaces (which is the equivalent to isolines in 2D), profiles, and volume renderings, then rotate and animate the view. Volume visualisation can be combined with raster and vector data. As external volume viewer the VTK based Paraview.org software is supported.

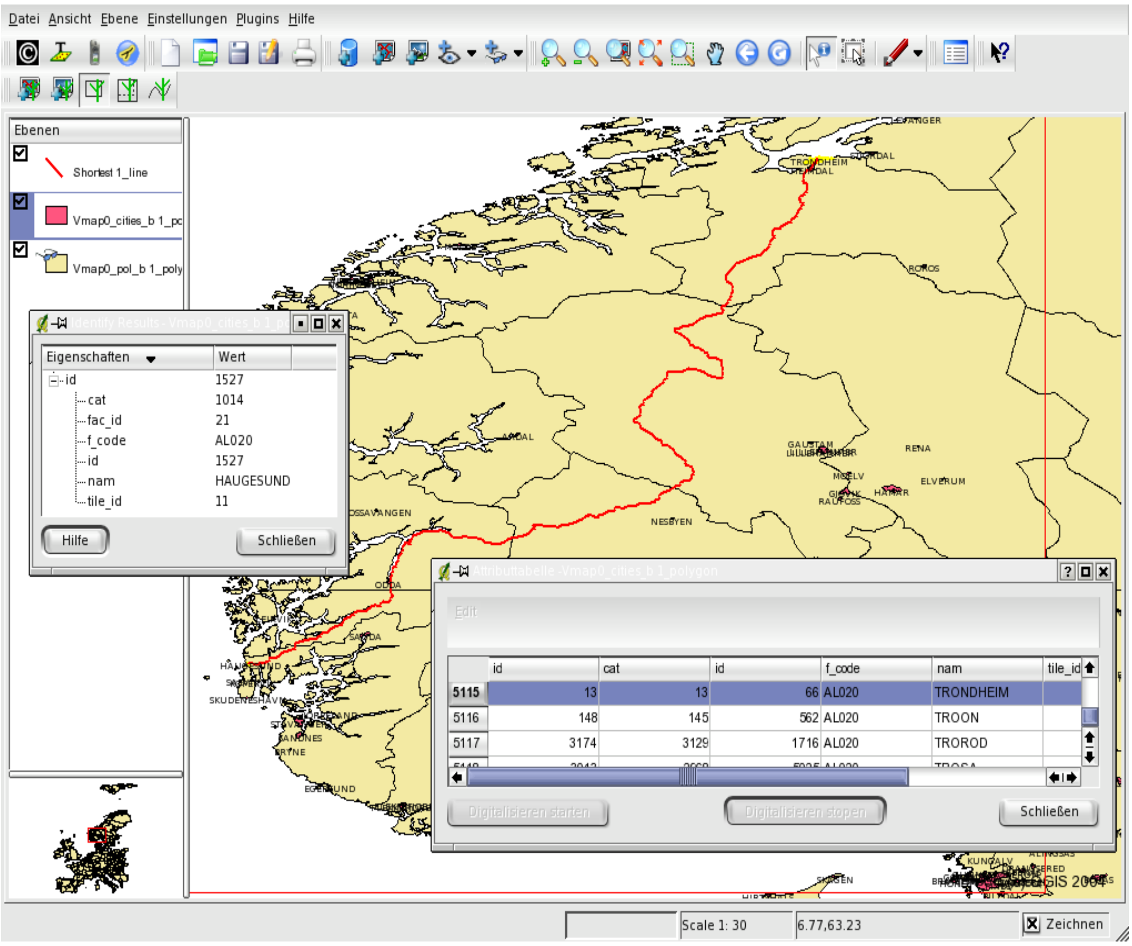

Figure 5: Shortest path example for vector network analysis (seen in QGIS geodata viewer)

2.3 Geostatistics

Since GRASS does not provide high level geostatistical functionality, a couple of interfaces is available. These interfaces link to external software packages such as R statistical language, MATLAB, Octave, and gstat. Within GRASS data can be pre-processed, then taken over to the external package and results stored back. GRASS itself provides raster time series processing where each cell value is a statistical function of the values of the corresponding cells in all the input raster maps.

2.4 Beyond clicking: Scriptability and programming

Due to its open source nature, GRASS provides a unique opportunity to improve and extend GIS capabilities by new code development. The complete GRASS source code is available on the GRASS Web site. The code base is written in ANSI C programming language. To make the development of GIS tools more efficient, GRASS provides a large GIS library with documented application programming interface (C and C++ API). Two basic levels of programming are supported. Average users will use 'script programming' to simplify repeating processes, while advanced users can extend existing code or even develop new modules. The modular concept of GRASS provides huge potential for development. Most modules are also usable from command line which allows the user for integration into UNIX shell, PERL, PHP or Python scripts. The C API is exposed to other programming languages through a SWIG interface (currently PERL and Python). While GRASS doesn't yet provide support for parallel computing, it is possible to use it in a simple scripted approach on openMosix clusters (Neteler et al., 2004). Especially in image processing, when thousands of data sets have to be analysed, GRASS gains high performance.

2.5 World Wide mapping

On top of this, Web based spatial services are developed. In particular, opportunities for integrating Spatial Data Infrastructure (SDI) into Internet based services have grown. Interoperability and standardisation is granted by following standards of the Open Geospatial Consortium (OGC) for geodata and related information technologies. While internally data storage and processing may differ from proposed OGC standards, all recent FOSS for Geomatics systems provide data exchange interfaces to industrial standards. The open source UMN/Mapserver can be easily integrated with GRASS as it directly reads GRASS raster and vector data through the GDAL/OGR interoperability library directly from a GRASS database. Alternatively, GRASS vector data can be stored into a PostGIS spatial database which is then linked to UMN/Mapserver. An ongoing project is the 'GRASS Server' turning it into a GIS backbone. An application is the support of wireless hand-held devices (field mapping etc.) which request data subsets or give a push to complex and data intense algorithms on the server side and then just receive the results.

2.6 Outlook: liberty

The FOSS concept offers a license scheme which comprises freedom of usage, modification, redistribution of original and modified software. For GIS the unlimited access to the source code is of particular interest as the underlying algorithms are often complex with great influence to results of spatial analysis and modelling. While an average user may not be able to trace errors within a complex source code, there is a number of specialists willing to test, analyse and fix the code. This framework is embedded into an Internet based development model with high frequency of new releases. The different backgrounds and expertise of the developers contribute to the synergistic effects. Overall such a development model leads to faster and more cost effective software production and stable and robust products.

Acknowledgement

Part of the results presented in this paper evolved from the collaborative work carried out during MN's tenure in summer 2004 as visiting researcher at Media Center Osaka City University (OCU), Japan. MN would like to express his sincere gratitude to the President, Osaka City University for his generous invitation. He would also like to thank colleagues at OCU, especially Dr. Venkatesh Raghavan, Dr. Shinji Masumoto and Dr. Katsuichi Kita for their cooperation and useful discussions.

References

Masumoto, S., Raghavan, V., Nonogaki, S., Nemoto, T., Mori, T., Niwa, M., Hagiwara, A., and and, N. H.(2004)- Present status of GRASS internationalization project, Proc. Free/Libre and Open Source Software for Geoinformatics. GIS-GRASS Users Conference, Bangkok, Thailand.

Mitasova, H. and Neteler, M.(2004) - GRASS as Open Source - Free Software GIS: accomplishments and perspectives. Guest editorial. Transactions in GIS, 8 (2): pp145-154.

Neteler, M., Grasso, D., Michelazzi, I., Miori, L., Merler, S., and Furlanello, C.(2004) -New image processing in GRASS, Proc.Free/Libre and Open source software for Geoinformatics. GIS-GRASS Users Conference 2004.

Neteler, M. and Mitasova, H.(2004)Open Source GIS: A GRASS GIS Approach. Number 773 in SECS. Kluwer Academic Publishers/Springer, Boston, 2nd edition.

Raghavan, V., Kita, K., Iwao, K., and Neteler, M.(2003) Open source GIS -GRASS for developing spatial data infrastructure - Present status and future potential. Journal of Information Science and Technology Association, 4 (53): pp 216-224

Stankovic, J., Neteler, M., and Flor, R.(2002)Experimental mobile wireless GRASS based GIS for handheld computers running GNU/Linux, Proc. Open Source Free Software GIS-GRASS users conference *2002*, Trento, Italy.

Westervelt, J(2004)GRASS roots, Proc. Free/Libre and Open Source Software for Geoinformatics: GIS-GRASS Users Conference, Bangkok, Thailand.