# Towards a stable open source GIS: Status and future directions in GRASS development

presented at
Second Italian GRASS Users Meeting,
University of Trento, Feb. 1-2 2001

Markus Neteler

University of Hannover – Geographical Institute
Schneiderberg 50 – 30167 Hannover, Germany
neteler@geog.uni-hannover.de

July 2001

**Abstract**

The ongoing GRASS GIS project is following the open source development model with fully accessible source code which is released under GNU General Public License and regular releases. The basic development is undertaken by the "GRASS Development Team", special contributions are delivered by institutions or individual developers. If such contributions are interesting for a wider audience, they are integrated into the main GRASS system. In 2000 and first half of 2001 the development has been focusing on code stabilization and portability – GRASS shall be compliant to rather any UNIX derivate and MacOS X (Darwin) and, utilizing the CygWin tools, even on MS-Windows platforms. Important new features of GRASS 5.0.0 are introduced in this article. New ideas and changes will be introduced in the upcoming GRASS 5.1.0, the author briefly outlines basic ideas of proposed improvements. Due to recent changes the paper reflects latest development up to July 2001.

*Key words: GRASS, GIS, LAPACK/BLAS, portability, XDriver, XML*

## 1 Review – GRASS 5.0.0 Development in 2000

The GRASS (Geographic Resources Analysis Support System) Development Team is close to release the stable version 5.0.0 which provides major functionality improvements since years. After two years of development which was based on the inofficial sources received from U.S. Army CERL, the GRASS Development Team has nearly finalized this part of the project. At time of this writing GRASS 5.0.0pre1 has been published recently which can be considered as a reliable and portable GIS system. During the final test phase bugs and portability problems (GRASS 5 now runs on more than 10 different architectures) will be addressed to achieve the stable version. The following section will describe some aspects of the several thousand code changes which have been applied to GRASS

The Geomatics Workbook N. 2 (2001)
Politecnico di Milano, Italy (ISSN 1591-092X)

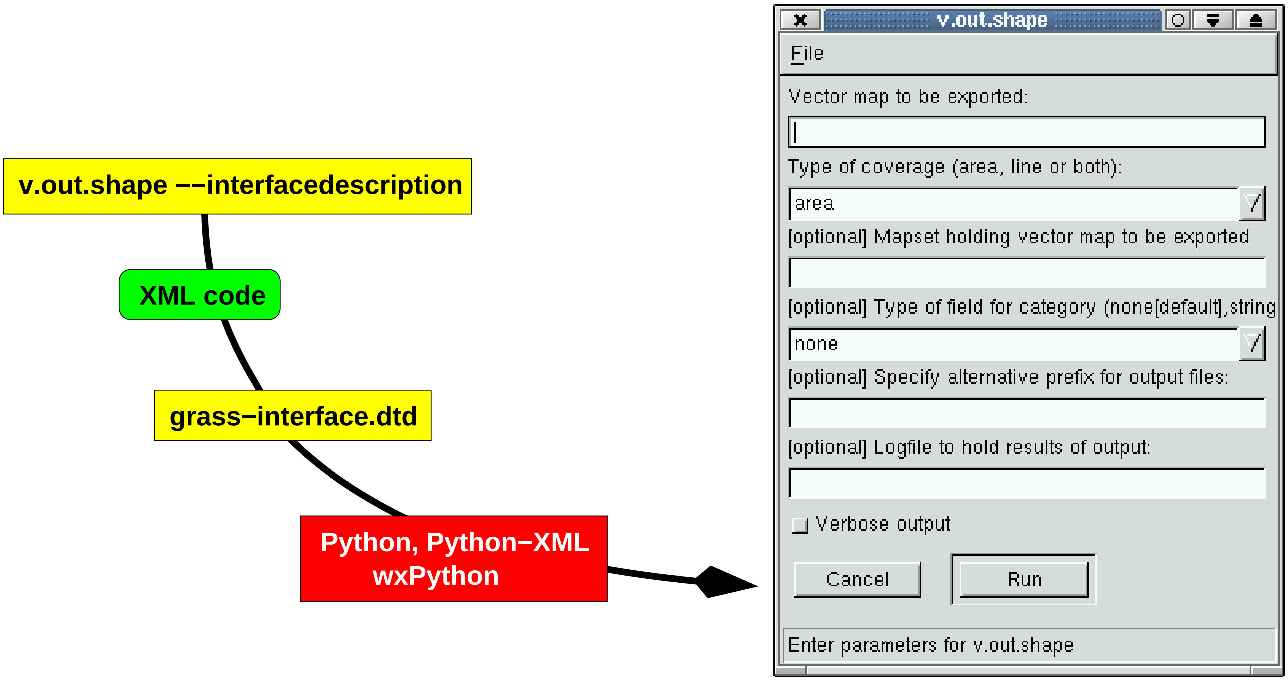
during the twelve month. Deficiencies in comparison to other proprietary GIS will be addressed in the forthcoming version GRASS 5.1.0.

## 1.1 Improved spatial data exchange

The data import and export capabilities are essential for any GIS related work. Due to the multiple data formats used in the GIS world and a currently low standardization level GIS software needs to support various data formats. Comparing to GRASS 4.x the capabilities have been improved: GRASS now allows to import vector data in SHAPE, E00 and MapInfo format. As SHAPE is an non-topological vector format, a special topology engine was implemented to optionally correct data inconsistencies during import. The only current drawback is that the vector file has to be processed completely in memory which leads to speed reductions and eventual high memory consumption. This problem will be addressed in GRASS 5.1 where segmented vector processing will be implemented. For raster data import support a bridge to “GDAL” (Geospatial Data Abstraction Library) has been implemented. GDAL is a library which reads more than 20 GIS and image processing raster formats ranging from simple (geocoded) images to SAR data containing complex numbers. GRASS 5.0.0 provides some new export modules as well: SHAPE, E00 (with the limitation of projection information not yet being exported) and MapInfo.
A new external library "libgrassio" has been developed: This standalone GRASS library is suitable for non-GRASS applications which directly want to read from and write to GRASS databases. For example map browsers or geostatistical software are possible applications to use this function set. As being released under LGPL, even proprietary software products may integrate it.

## 1.2 Source code changes

An essential improvement for the entire development process has been the integration of GRASS 5.0.0 source code into a CVS-server (Concurrent Versioning System) since end of 1999. The CVS-server establishes a modern method to manage source code in a centralized manner. Every change is reviewed by other developers, new GRASS releases can be easily extracted. As CVS works fully automated, there are no longer limitation to access and modify the GRASS software except local technical reasons or network problems. During the year 2000 more than 3000 so-called CVS-commits have been uploaded: bugfixes, improvements, source code and documentation updates.
A focus of development has been to increase the platform portability of GRASS. Beside various UNIX compliant operating systems the recently published MacOS X (Darwin) system is now supported as well as MS-Windows utilizing the CygWin tools. Latter still lacks the common "InstallShield" but this is just a matter of time.
Due to GRASS’s history numerical functions have been spread throughout the code. These functions are now assembled in a new mathematical library “gmath” which additionally provides wrapper functions to support LAPACK/BLAS routines (http://www.netlib.org/lapack/). The dependency on one single library shall improve the numerical reliability. For future modules which require stable numerical algorithms the LAPACK/ BLAS functions provide sophisticated and well tested vector and matrix operations support. Even a parallelized version is available which performs optimized on computer clusters.
The projection software has been syncronized to the current main release of “PROJ4” software (http://www.remotesensing.org, originally developed by USGS, now this software is maintained by Frank Warmerdam). Future plans cover the integration as a shared library and utilization of the datum shift functions which are already implemented in PROJ4. At time datum shifts are unsupported in GRASS, this will be changed for GRASS 5.1.0.



Vector map to be exported:

Type of coverage (area, line or both): area

[optional] Mapset holding vector map to be exported

[optional] Type of field for category (none[default],string none

[optional] Specify alternative prefix for output files:

[optional] Logfile to hold results of output:

□ Verbose output

Enter parameters for v.out.shape

*Figure 1: Autogenerated menu entry based on XML/wxPython. The XML output contains text from the “–help” parameter. Command line call:*

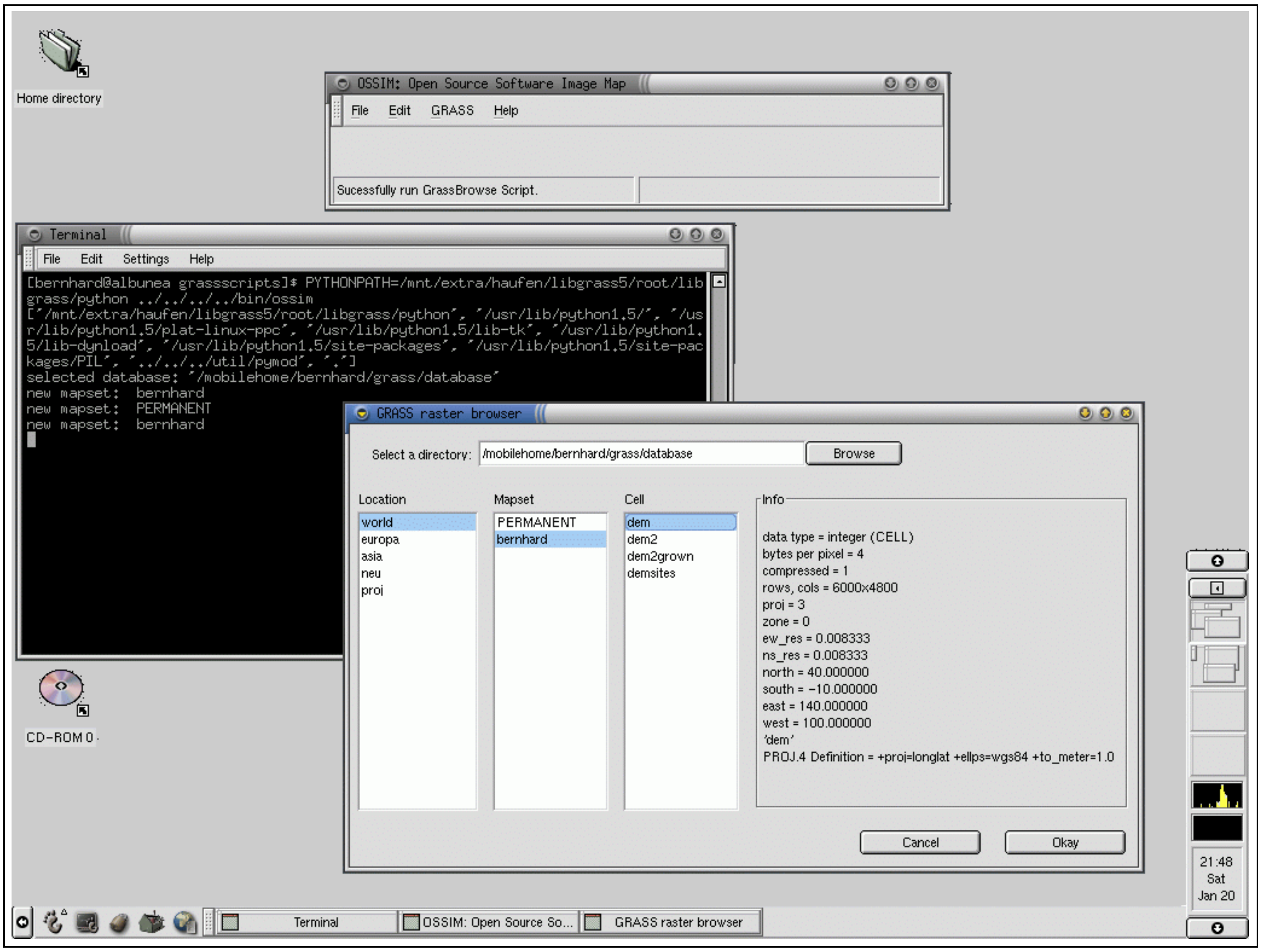
```
v.in.shape -interface-description | python src/gui/python/grassgui.py
```

## 1.3 Legal issues

The GRASS 5.0.0 package consists of about 1.5 million lines of C-source code which is released under GPL (GNU General Public License). GRASS is by size still belonging to the Top 10 largest open source projects worldwide (`http://www.codecatalog.com`). To fulfill the GPL restrictions, a major cleanup was undertaken to remove copyright-restricted routines which have been rarely used within the code. A special problem had to be addressed: For GRASS 5 floating point data the LZW-compression has been used which is copyrighted (and restricted) by UNISYS company. To avoid any problems, this compression method was completely removed and changed to the free “zlib” compression. For GRASS databases which were generated in LZW-compression the user has to convert them with a special script “r.lzw2z” which is available online.

## 1.4 Graphical User Interface/XDRIVER development

The long time awaited rewriting of the XDRIVER has been finished: The GRASS monitor can now be resized and closed by mouse like other windows, the displayed maps are immediately auto-redrawn. Internally the XDRIVER was modified to use “sockets” instead of the slower “fifos”. These “sockets” allow much faster communication between the GRASS core system and the GRASS monitor. Accordingly the CELL driver, a recently developed PNG driver (to directly write monitor output into a PNG raster map) and a new HTMLMAP driver (to directly write vector output into a HTML image map) have been updated. To support modern GUI (Graphical User Interface) development, the module short descriptions, well known from the "help" parameter, can be printed in XML format using the new "–interface-description" parameter (see fig. 1). This feature is required for the forthcoming automated GUI-builder which will create menus on-the-fly. The XML definitions are stored in the new GRASS Bridge DTD, originally developed for the OSSIM project (Open Source Software Image



Sucessfully run GrassBrowse Script.

```
[bernhard@albunea grassscripts]$ PYTHONPATH=/mnt/extra/haufen/libgrass5/root/lib
grass/python ../../../../bin/ossim
['/mnt/extra/haufen/libgrass5/root/libgrass/python', '/usr/lib/python1.5/', '/us
r/lib/python1.5/plat-linux-ppc', '/usr/lib/python1.5/lib-tk', '/usr/lib/python1.
5/lib-dynload', '/usr/lib/python1.5/site-packages', '/usr/lib/python1.5/site-pac
kages/PIL', '../../../util/pymod', '.']
selected database: '/mobilehome/bernhard/grass/database'
new mapset:  bernhard
new mapset:  PERMANENT
new mapset:  bernhard
```

Select a directory: /mobilehome/bernhard/grass/database

Location: world, europa, asia, neu, proj

Mapset: PERMANENT, bernhard

Cell: dem, dem2, dem2grown, demsites

```
data type = integer (CELL)
bytes per pixel = 4
compressed = 1
rows, cols = 6000x4800
proj = 3
zone = 0
ew_res = 0.008333
ns_res = 0.008333
north = 40.000000
south = -10.000000
east = 140.000000
west = 100.000000
'dem'
PROJ.4 Definition = +proj=longlat +ellps=wgs84 +to_meter=1.0
```

*Figure 2: GRASS map browser prototype for OSSIM project: XML/wxPython*

Map, http://www.ossim.org/). At time of this writing all command line modules support the new feature. Eventually it will be extended to generate the module manuals dynamically which are currently available in HTML format.

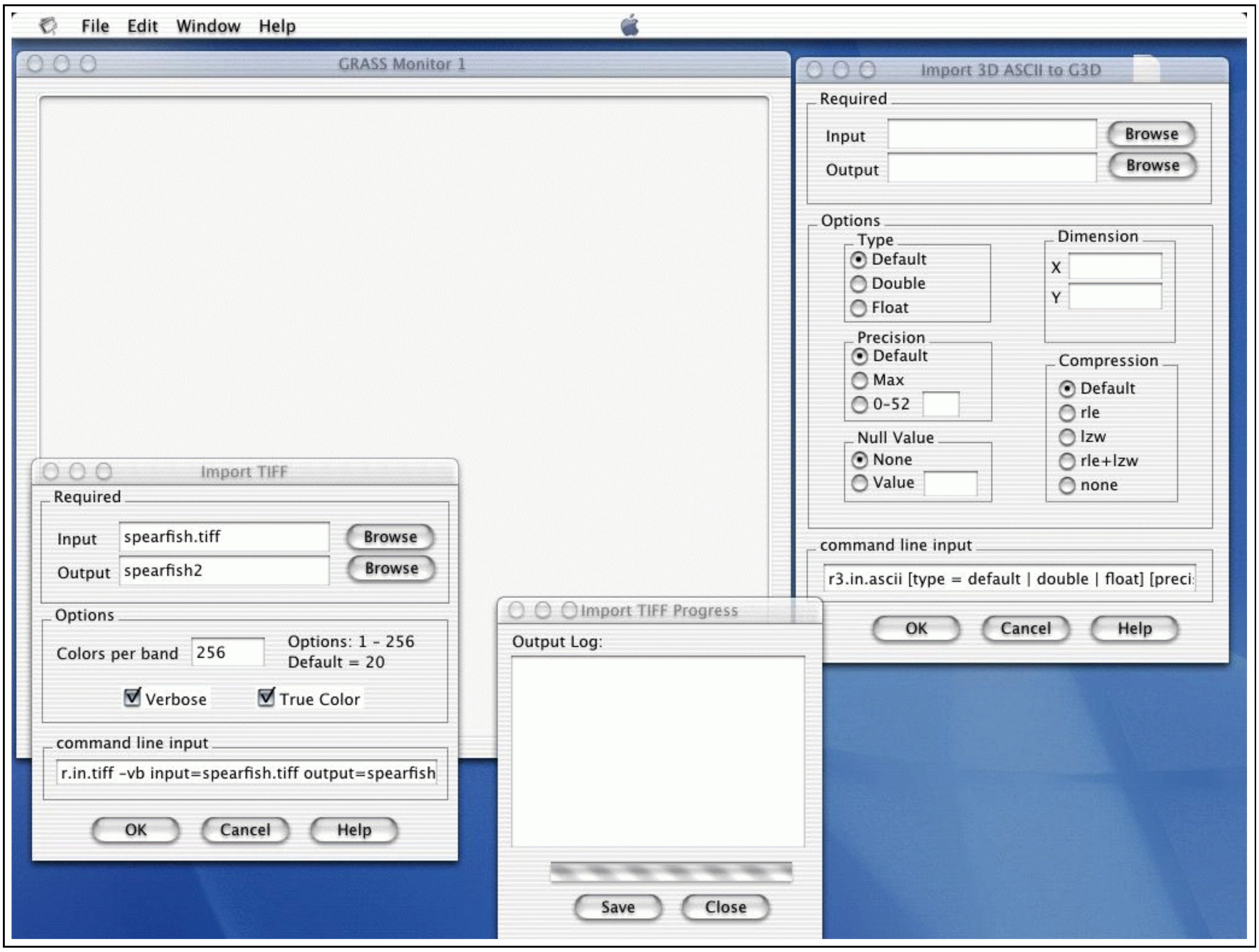
A prototype GRASS map browser developed by Intevation GmbH can be seen in fig. 2. The portability even allows to develop platform independent user interfaces: Fig. 3 shows a prototype on MacOS X which was developed with “Cocoa” programming environment.

## 1.5 New (Geo)-Statistics Interfaces

Two new interfaces to support external statistical software have been developed:

- R statistical data language interface (Bivand 2000, Bivand & Neteler 2000)
- gstat interface (Edzar J. Pebesma 1998)

R consists of a core statistical system (R statistical data language) supposed to be the GNU clone of the proprietary S-Plus. It comes along with many extensions which add further statistical functionality like geostatistics, cluster analysis, classifications, time series etc. to the core system. These extensions are maintained individually by their authors. In terms of geostatistics R recently offers various geostatistical extensions to support variogram analysis and different interpolation methods.



Import TIFF

Required
Input spearfish.tiff
Output spearfish2

Options
Colors per band 256
Options: 1 - 256
Default = 20
☑ Verbose ☑ True Color

command line input
r.in.tiff -vb input=spearfish.tiff output=spearfish

Import 3D ASCII to G3D

Required
Input
Output

Options
Type: ● Default ○ Double ○ Float
Dimension: X Y
Precision: ● Default ○ Max ○ 0-52
Compression: ● Default ○ rle ○ lzw ○ rle+lzw ○ none
Null Value: ● None ○ Value

command line input
r3.in.ascii [type = default | double | float] [preci

*Figure 3: GRASS prototype GUI on MacOS X using Cocoa GUI builder*

The GRASS-R-interface allows to exchange raster, vector and sites data from and to R for further analysis. The 3D raster voxel is forthcoming. As the existing memory limitation has been removed in R (in past the demanded memory had to be reserved at startup), only the physical memory limits the amount of data for analysis. Using R diagrams and maps can be generated and stored in Postscript format. The R-XGOBI interface is an excellent extension for data visualization and exploration. In terms of remote sensing multi-dimensional datasets can be visualized through the “grand tour” feature.

The gstats software is a pure geostatistical tool. Using the gnuplot visualization tool it can plot sample variograms and variogram functions. Like R it is ported to most computer platforms. Gstat covers the modelling, prediction and simulation of geostatistical data in one, two or three dimensions with a focus on the various kriging methods. Similar to R it may be programmed with it’s own programming language.

The GRASS-gstats interface has been published recently (from gstats version 2.3.3 onwards, 2001). It is built-in into the gstats package and allows to read and write GRASS spatial data. Point data are read from site lists or raster maps, output (prediction or simulation results) is written to either raster maps or site lists.

Both R and gstat have to be started within GRASS as the environment variables defined by GRASS are interpreted. As known from GRASS raster modules, both interfaces are region sensitive. This concept allows to easily select and analyse sub-regions.

# 2 Future Directions - Development in 2001/2002

After release of GRASS 5.0.0 (stable) the version number scheme will be changed to the Linux Kernel numbering scheme: GRASS 5.*even*.x will reflect stable versions, while GRASS 5.*odd*.x denotes development versions. The "beta" numbering used in past won't appear any more.
The 5.1.0 development can be split into two main phases, a restricted "Phase 1" and a second open "Phase 2". Various milestones will be addressed during the development. As the 5.1.0 development will be started with a major code reorganization, a restricted phase is necessary to minimize the amount of unpredictable bugs during the implementation phase.

## 2.1 GRASS 5.1.0 restricted "Phase 1"

The first task will include the development of a new directory layout. This is required for the implementation of a new Makefile system based on "automake" which replaces the current GRASS Gmakefiles. A second aim deriving from the new code layout and a code separation is to build the GRASS libraries as shared libraries. This will reduce the memory requirements for each module and speed up loading time. As noted above a separation of the GRASS database I/O routines from GISLIB will be implemented. These routines (currently limited to raster I/O routines) are already extracted into an external "libgrassio" package.
As a third major task a new improved GRASS vector library will be integrated. To overcome current GRASS limitations, the new library supports 2.5D/3D vector data, will have integrated database management system (DBMS) support to manage multiple vector attributes and will be fully 32/64bit compliant. The new attribute management requires a built-in DBMS interface. The new core vector library has already been separately developed as well as a basic DBMS driver. To speed up vector topology build processes on large vector files, a new vector segmentation algorithm will be implemented. The next milestone includes the update of all vector modules. Finally, after adding the raster and sites modules to 5.1.0, a modular packaging concept will be established. Such packages allow topic oriented web downloading e.g. of a "GRASS image processing package" or the "GRASS DEM analysis package" additionally to the "GRASS core package". At this stage of implementation GRASS 5.1 will become fully usable.

## 2.2 GRASS 5.1.0 open "Phase 2"

Already separately developed, but not yet implemented into 5.x is a fully graphical startup screen with session manager, which allows to build locations on the fly from external GIS data, along with location enlargement on-the-fly and graphical location definition. However, the text based startup will be kept in future for low-level access.
To improve and speed up the sites management, in GRASS 5.1.0 the sites list will be managed within a DBMS. This allows to support "NULL data" in multiple attributes sites lists (not possible in the current ASCII lists).
Another task is the integration of FreeType font engine to support rather any font. GRASS 5.0.0 still lacks Postscript or TrueType fonts due to limited support (at time restriction to Hershey fonts). The forthcoming FreeType font engine allows to display even Asian or other non-latin fonts.
To improve the acceptance of GRASS especially for new users an internationalization (multiple languages support) will be undertaken. Using the GNU "gettext" system, all messages will be maintained in tables. For each language a column will contain the translated messages. Language selection is controlled by country codes.

# 3 Conclusion

The GNU licensing of GRASS continuously attracts an increasing number of developers. The chosen license model prevents the software authors from misuse of their developments due to the license restrictions at the same time of full access. All participants benefit from Open Source concept. Sharing ideas and knowledge with an underlying common GIS concept improves flexibility and speeds up own developments. The current limitations of GRASS 5.0.0 will be addressed in GRASS 5.1.0.

# 4 References

Bivand, R. S. (2000) – Using the R statistical data analysis language on GRASS 5.0 GIS database files. Computers & Geosciences, 26, pp. 1043-1052

Bivand, R., M. Neteler (2000) – Open Source geocomputation: using the R data analysis language integrated with GRASS GIS and PostgreSQL data base systems. Proc. 5th conference on GeoComputation (CDROM), 23-25 August 2000, University of Greenwich, U.K.
http://reclus.nhh.no/gc00/gc009.htm

Dongarra, J.J. (1999) – LAPACK numerical library,
http://www.netlib.org/lapack/

Pebesma, E.J., C.G. Wesseling (1998) – Gstat: a program for geostatistical modelling, prediction and simulation. Computers & Geosciences Vol. 24, No. 1, pp. 17-31
http://www.gstat.org

Imagelinks Inc. (2000) – Open Source Software Image Map (OSSIM),
http://www.ossim.org/

Neteler, M. (ed.) (2000) – GRASS 5.0 Programmer's Manual. Geographic Resources Analysis Support System. University of Hannover.
http://www.geog.uni-hannover.de/grass/grassdevel.html