Innovative Tools for TerraHidro: An Intuitive GIS User Interface

Thiago L. M. Barreto[1], Sergio Rosim[1]

[1]Divisão de Observação da Terra e Geoinformática – Instituto Nacional de Pesquisas Espaciais (INPE)
São José dos Campos SP Brazil

{thiago.luiz,sergio.rosim}@inpe.br

Abstract.

This article shows the development and implementation of the TerraHidro system, interface version, in Linux, with innovations in relation to the traditional interfaces of geographic information systems. The objective is to reduce user interaction with the system, allowing several functions to be performed with just one interaction, with intermediate functions being performed automatically. Therefore, it was necessary to create a metadata structure to associate each data with its type. In addition, a tool was implemented to save the execution history, under user command. A text file will be created. It will work with a TerraHidro function script in the command line version.

1. Motivation

Traditionally, GIS (Geographic Information Systems) contains complex interfaces with the characteristic of demanding a large number of interactions from the user, who is normally an expert. Every new step requires an action from the operator informing their respective parameters and necessary file paths, which may be problematic for a continuous workflow environment. This is, the person needs to be fully dedicated for every new step avoiding the process to stop.

This paper presents the performed work to change the amount of operator interactions with the TerraHidro system interface, which is a GIS developed for distributed hydrological modeling tasks. It was achieved an important improvement on the user interface by reducing the operator dependency on the system, which is now able to execute several tasks in sequence by just configuring one interface instead of many. That indicates less time spent for setting up the interface and reduces the chance of introducing human errors. This was reached by implementing a generalized metadata schema mapped for the workflow and subject to further improvements. In addition, it is possible to save the processing steps and parameters into a history file for each executed interface, enabling the operator to have an exact batch script version of his work.

2. Related Works

It has been notable how Human Computer Interfaces (HCI) have recently evolved, especially those from mobile devices [2] where a certain level of adaptation was achieved between users and their environment (i.e. automatic adjustment of screen brightness based on surrounding light levels). Some other systems were automated in order to find groups based on their roles with a view to software engineering [4] and designing [3]. Thanks to studies with focus on the interface design and its layout components [5] we are able to understand better how users' performance interacts with respect to agronomy [6] (e.g. font size/color, element displacement) and software engineering driven development/tests [7, 8, 9].

When it comes to GIS interfaces we understand how complex this issue is since it normally demands highly skilled professionals for operating it as a result of the lack in design standards [1]. This happens due to complexity of interoperability [10, 11] in terms of: data (file formats, I/O), functional (CPU/GPU processing, memory), and device (web-based, desktop-based).

Recently, there were suggested different approaches for designing interfaces based on the result of quality attributes (hedonic or pragmatic) from users evaluation questionnaires [12, 13, 14, 15]. This not only supports more efficient software engineering prototyping cycles but also enables the evaluation between two or more distinct interfaces.

This paper aims to present a more intuitive GIS interface by an approach that drives the user to automatically set up the correct parameters and file paths based on previous processing stored metadata.

3. The TerraHidro Concepts

In general, systems have the property of approaching users with little computational knowledge, called laymen, of complex functionalities. Typically, this type of user is focused on producing results that are often repetitive. The characteristics of these GIS are the strong interaction between user and system. What is proposed in this work is the optimization of the time spent by the user by reducing the necessary interactions with TerraHidro GIS. For this, the TerraHidro interface was reprogrammed. A metadata schema was developed to initially contain the type of each data in the TerraHidro scope. Finally, the history of the functions executed, under user command, was also implemented, which is not common in systems with interfaces.

The user interface for TerraHidro has been designed and developed under TerraLib/TerraView framework, which is an open source library for Desktop customized GIS applications written and compiled in C/C++ under the GNU Lesser General Public License (LGPL). All new functionalities can be found as a shared library plugin compiled and installed into Terraview allowing new interfaces to be accessed through the main menu.

In this section, we present in detail the newly designed interfaces for hydrological modeling, the metadata implementation which allowed an important improvement on user experience, and finally the usability of storing every system call into batch scripts.

3.1. Interfaces

Each interface was designed and implemented with respect to the general workflow diagram presented in Figure 1 as well as its respective relationship in classes containing their inputs and outputs. The process starts with a given Digital Elevation Model (DEM) of a scene being pre-processed for removing flat areas and pits (REMOVE_PITS).

Then, each pixel is computed with respect to its local direction flow inside a 8 neighbor (D8) window. This allows the system to compute the accumulation of the flow (D8) and to extract the river streams by applying a cut-off threshold classification (DRAINING). Finally, every individual river stream is calculated (SEGMENTS) in order to obtain its respective river basin (MINIBASINS).

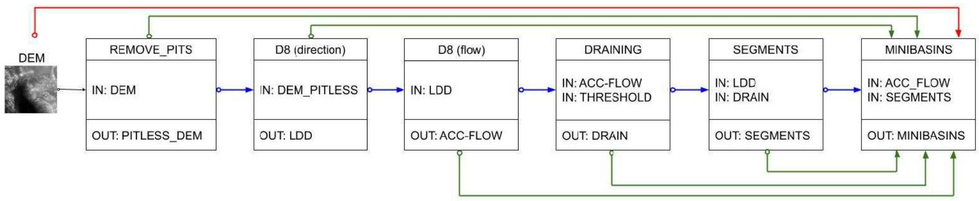

Figure 1. Workflow diagram of TerraHidro main functions.

This workflow can be executed in two different ways. Step by step, when the user needs to command the execution of each function, in the following sequence: REMOVE_PITS, D8, D8CA, DRAINING, SEGMENTS e MINIBASINS. It's shown in the Figure 1 by the blue line sequences. Another way is the direct execution of the desired function, MINIBASINS in this case, from the existing input data. In this case, the user will only have to interact once, as the intermediate functions are being performed automatically. It's presented by the red line.

In this figure, the user starts by giving the original altimetry grid as input data. The possibilities are shown Once activated, the MINIBASINS function will cause the system to execute REMOVE_PITS, D8, D8CA, DRAINING, and SEGMENTS functions automatically, generating their respective data files. The concept will be the same if the initial data is the altimetry grid already with the flat areas and local minima extracted, or the grid of local flows, or the grid of accumulated flows, or the drainage grid, or even the grid of segments of drainage. The other possibilities can be seen in the green lines at the same figure.

3.2. Metadata

The type of each data file is critical for this interface design to work. The data type will indicate which functions must be performed, according to the function performed by the user. For example, the user has an altimetry grid as input and aims to execute the DRAINING function. The data type informs if the grid is of original altimetry (DEM) or if it has already been processed to eliminate flat areas and local minima. Therefore, each raster file will have at least one metadata that is its type. All metadata information is stored in a .json data structure, described below.

In the literature it is possible to find good examples of metadata being used for geoinformation systems. Among them, we can mention those derived from ISO institutional standards or free code from the W3C (World Wide Web Consortium) [16,17], and which supported the adoption of the standard presented below. After numerous discussions with different stakeholders, we came up with a metadata model to be manipulated by the system in .json file format. Figure 2 presents the final data structure, as well as their respective fields and description. The .json data structure model was chosen due to its wide adherence among today's main development tools, and can be easily integrated with them, in addition to being an open source file standard.

Metadata	Description
"id":"000000"	A unique string key containing the file ID
"predecessors": ["000","111"],	A set of file IDs corresponding to all predecessors
"pullpath_filename": "path"	A string field containing the full path of the file
"process_type": "type"	A string field to carry on which step of workflow was responsible to generate that file specifically
"threshold": { "absolute":"value", "percentual":"value" }	A struct for storing which threshold was used for binarization of river streams
"extras": [ ]	A free space for user custom metadata storage

Figure 2. Metadata structure and description.

It was also discussed possible limitations of the methodology such as the number of attributes, generalization of parameters and inclusion/reading of attributes by the user. All of them were solved by implementing a metadata class called "extras" as described in Figure 3. Within this data structure, it is possible to define metadata different from what was initially planned.


```
"extras": [
        {
                "pointName":"point01",
                "pointDesc":"gps point close to river",
                "XCoord":"100",
                "YCoord":"200",
                "Lat":"-23",
                "Lon":"-12"
        },
        {
                "pointName":"point02",
                "pointDesc":"gps point close to town",
                "XCoord":"1000",
                "YCoord":"2000",
                "Lat":"-23.01",
                "Lon":"-12.01"
        }
  ]
```


Figure 3. A piece of a metadata file generated with the "extras" function to store data related to geographic coordinates.

It is considered a file that is imported into the TerraView geographic viewer, which hosts the TerraHidro plugin. By the first time a TerraHidro function is executed, the user will receive a warning alert that the input file has its unspecified. The user must then associate the appropriate type, ensuring that any data used by TerraHidro has its data type.

3.3. Interface to Batch Script (history of executions)

The third feature we presented on our GIS is the possibility for the user to store the execution history in a .txt file. He defines the starting and ending point of the file and the generated file will contain the functions performed during the execution of TerraHidro interfaces, already in the TerraHidro command lines format.

For instance, considering the workflow executed from SEGMENTS, as shown in the Figure 4, step and having already run PITLESS_DEM as shown in bold in Figure 4, will produce the following batch script accordingly. Note that for each executed step, the filenames are being defined automatically in order to be convenient.

```
./th d8 /home/inpe/data/s13_w040_1arc_v3_PITLESSDEM.tif
/home/inpe/data/s13_w040_1arc_v3_PITLESSDEM_LDD.tif
./th d8ca /home/inpe/data/s13_w040_1arc_v3_PITLESSDEM_LDD.tif
/home/inpe/data/s13_w040_1arc_v3_PITLESSDEM_LDD_ACC-FLOW.tif
./th d8drainage /home/inpe/data/s13_w040_1arc_v3_PITLESSDEM_LDD_ACC-FLOW.tif
/home/inpe/data/s13_w040_1arc_v3_PITLESSDEM_LDD_ACC-FLOW_DRAIN_100_00.tif
12329.30
./th segments /home/inpe/data/s13_w040_1arc_v3_PITLESSDEM_LDD.tif
/home/inpe/data/s13_w040_1arc_v3_PITLESSDEM_LDD_ACC-FLOW_DRAIN_100_00.tif
/home/inpe/data/s13_w040_1arc_v3_PITLESSDEM_LDD_ACC-
FLOW_DRAIN_100_00_SEGMENTS.tif
```

Figure 4. Resulting batch script from storing the execution history based on Figure 1.

4. Application exemple

This example shows the execution of the DRAINING function from the original altimetry grid. Before, this grid was imported into TerraHidro, receiving the DEM data type as metadata. The user just had to choose this function by selecting the altimetry grid and pressing the "execute" button. Figure 5 shows the original grid, the interface used and the drainage result, in vector format.

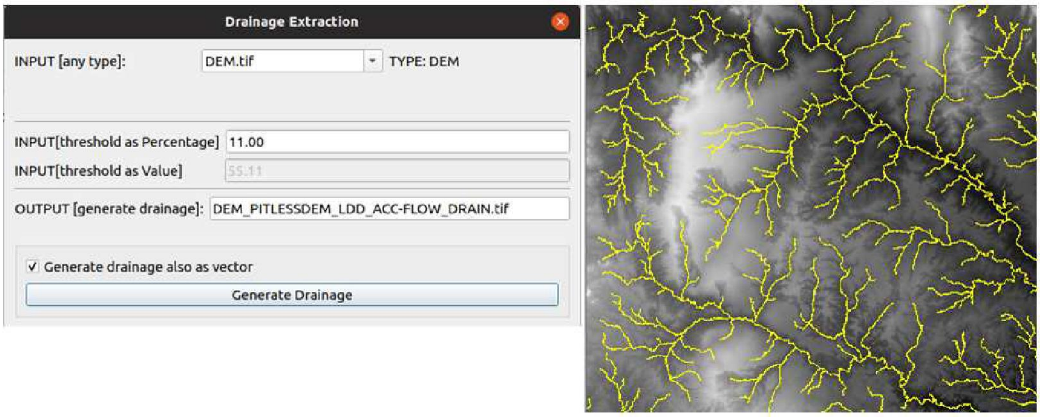

Figure 5: interface for drainage extraction and altimetry grid and drainage extracted.

5. Conclusions

The objective of this work was the development of the TerraHidro system, interface version, which could optimize the user's time in their interactions with this system. In addition to the step-by-step processing, the user can skip some steps that are automatically executed by the system, which, recognizing the initial data by its type, knows which functions must be executed. For that, a metadata structure was created to support the type of each data imported or created in the system, also allowing the user to create new metadata values. Another differential of this interface is the possibility of saving the execution log in the format of TerraHidro's own command lines.

6. Acknowledgments

The authors would like to thank the Brazilian National Research Council (CNPq) for the research funding spent fellowship of first author.

References

[1] GOODCHILD, Michael F. Spatial thinking and the GIS user interface. **Procedia-Social and Behavioral Sciences**, v. 21, p. 3-9, 2011.

[2] HUSSAIN, Jamil et al. Model-based adaptive user interface based on context and user experience evaluation. **Journal on Multimodal User Interfaces**, v. 12, n. 1, p. 1-16, 2018.

[3] ZUBKOVA, T.; TAGIROVA, L. Intelligent user interface design of application programs. In: **Journal of Physics: Conference Series**. IOP Publishing, 2019. p. 012026.

[4] MARTINEZ, Jabier et al. Variability management and assessment for user interface design. In: **Human Centered Software Product Lines**. Springer, Cham, 2017. p. 81-106.

[5] SHAO, Junkai et al. How dynamic information layout in GIS interface affects users' search performance: integrating visual motion cognition into map information design. **Behaviour & Information Technology**, p. 1-18, 2022.

[6] WANG, Siyi et al. Interface Design of GIS System Based on Visual Complexity. In: **International Conference on Applied Human Factors and Ergonomics**. Springer, Cham, 2019. p. 713-722.

[7] SILVA, Thiago Rocha; WINCKLER, Marco; TRÆTTEBERG, Hallvard. Extending Behavior-Driven Development for Assessing User Interface Design Artifacts. In: **The 31st International Conference on Software Engineering & Knowledge Engineering (SEKE 2019)**. 2019.

[8] SAMRGANDI, Najwa. User Interface Design & Evaluation of Mobile Applications. **International Journal of Computer Science & Network Security**, v. 21, n. 1, p. 55-63, 2021.

[9] KURIA, Ezekiel; KIMANI, Stephen; MINDILA, Agnes. A framework for Web GIS development: a review. **International Journal of Computer Applications**, v. 178, n. 16, p. 6-10, 2019.

[10] KUPER, Paul Vincent. Design of a Generic Mobile GIS for Professional Users. **ISPRS International Journal of Geo-Information**, v. 7, n. 11, p. 422, 2018.

[11] AGRAWAL, Sonam; GUPTA, R. D. Web GIS and its architecture: a review. **Arabian Journal of Geosciences**, v. 10, n. 23, p. 1-13, 2017.

[12] KAMMERHOFER, David; SCHOLZ, Johannes. An approach to decompose and evaluate a complex GIS-application design to a simple, lightweight, user-centered app-based design using user experience evaluation. **ISPRS International Journal of Geo-Information**, v. 9, n. 9, p. 505, 2020.

[13] PYKÄLÄINEN, Tuukka et al. Making location data visual–Simplifying an online municipality-GIS presentation editor through user interface design. 2019.

[14] CHARFI, Selem; EZZEDINE, Houcine; KOLSKI, Christophe. RITA: a useR Interface evaluaTion frAmework. **J. Univers. Comput. Sci.**, v. 21, n. 4, p. 526-560, 2015.

[15] ROT, Artur; KUTERA, Robert; GRYNCEWICZ, Wieslawa. Design and Assessment of User Interface Optimized for Elderly People. A Case Study of Actgo-Gate Platform. In: **ICT4AgeingWell**. 2017. p. 157-163.

[16] DO PRADO, Bruno Rodrigues et al. Padrões para metadados geográficos digitais: modelo ISO 19115: 2003 e modelo FGDC. Revista Brasileira de Cartografia, v. 62, 2010.

[17] ALBERTONI, Riccardo; DE MARTINO, Monica; QUARATI, Alfonso. Documenting context-based quality assessment of controlled vocabularies. IEEE Transactions on Emerging Topics in Computing, v. 9, n. 1, p. 144-160, 2018.