Web-Based Map Generalization Tools Put to the Test: A jABC Workflow

Henriette Sens

Potsdam University, Potsdam, D-14482, Germany
hsens@uni-potsdam.de

Abstract. Geometric generalization is a fundamental concept in the digital mapping process. An increasing amount of spatial data is provided on the web as well as a range of tools to process it. This jABC workflow is used for the automatic testing of web-based generalization services like mapshaper.org by executing its functionality, overlaying both datasets before and after the transformation and displaying them visually in a .tif file. Mostly Web Services and command line tools are used to build an environment where ESRI shapefiles can be uploaded, processed through a chosen generalization service and finally visualized in Irfanview.

Keywords: map generalization, geo-visualization, cartography, CSISS, GDAL, geo-processing.

1 Introduction: Workflow Scenario

A cartographic map - unlike other forms of cartographic or map-like products such as sketch maps, infographics or chorèmes - requires by definition some kind of generalization of the visual representation applied to the geographic data being used [25]. The basic concept is that elements on a map always represent real world phenomena at a scale smaller than their actual size. Therefore, they must be graphically transformed and fitted to the map scale while keeping and even emphasizing their typical and significant appearance. Not semantic but geometric generalization is the key aspect of the present workflow. Geometric generalization means that the size and degree of detail of map objects should de- or increase linear to scale change.

Having been considered "difficult for a computer to execute well" and a rather "human characteristic of judgement" [1] in the early 1990s, with the rise of digital data processing and online on-demand mapping services such as Google Maps, map generalization is now an integral part of the digital cartographic modeling pipeline. But can automated processing achieve what the human eye and aesthetic perception have done for centuries in the map making process?

Generalization tools are widely implemented in desktop GIS or spatial databases like PostGIS, but online free and ready-to-use services are rare. One example of those services is mapshaper.org, a flash-based web tool that uses line simplification algorithms to reduce the complexity of a given polygon by eliminating a calculated subset of outline vertices. Manual alteration and vertex

A.-L. Lamprecht et al. (Eds.): Process Design for Natural Scientists, CCIS 500, pp. 175–185, 2014.
DOI: 10.1007/978-3-662-45006-2_14

editing is also offered. So the aim of this project is to create a testing environment for different generalization tools.

It consists of the following principal steps:

1. Choose a generalization tool
2. Execute it and store the new data as SHP (ESRI shapefile)
3. Convert it to GML (Geography Markup Language) format
4. Overlay and intersect the two data layers
5. Convert the intersection layer into TIFF (Tagged Image File Format) format
6. Display the TIFF in a new Irfan View instance

The idea is to provide a process that includes the data processing as well as the visualization and gets the user to choose between several tools. Input data required is a polygon or polyline dataset in ESRIs shapefile format. It is then being altered by the chosen web service which again has a shapefile output. To be processed in the CSISS overlay web service [3], it needs be converted to GML data format. Finally, the result of the overlay is rasterized to a TIF file which is displayed in Irfan View, a small free-ware program that lets one open and to a small extent, alter pictures.

2 Service Analysis

The fundamental concept of the workflow is to execute several services programmatically - mostly via the Execute Command SIB that addresses command line tools.

– **Mapshaper - URL (Execute Command)**
  Up to the present date, only one generalization service could be found on the web which is the mapshaper.org. It is therefore the only selectable service and the workflow will be based upon it. The flash-based web-application mapshaper.org comes with a client interface, shown in Figure 1, that can be addressed through a web browser via the URL `http://mapshaper.com/test/MapShaper.swf`. The SIB Mapshaper - URL addresses the command prompt "mapshaper.cmd" which opens a firefox browser instance and loads the given URL.
  Naturally, this SIB needs a web browser and a flash plug-in as a requirement. The service is platform independent and its parameters (e.g. generalization algorithms or detail level) can only be altered manually through the browser GUI and not via jABC.
  The application uses ESRI-Shapefiles which is a proprietary vector file system that consists of the 3 basic files .SHP (stores geometric feature information), .SHX (linkage between .SHP and .DBF) and .DBF (stores feature attributes).
  The service contains three different line-simplification algorithms to choose from (Douglas-Peucker, Weight-Visvalingham and Visvalingham-Wyatts) and

Fig. 1. Web Service Mapshaper.org

apparently eliminates topological errors such as overlap and gaps between adjacent polygons. The level of generalization can also be altered. Additional vertex editing tools can be used as well. The outcome can be viewed on-the-fly.

– **GDAL - Geospatial Data Abstraction Library (Execute Command)**
GDAL is a C++ open source translator library for raster and vector geospatial data formats and uses command line utilities for data translation and processing. It contains different subcomponents such as "OGR" or the utility program "GDAL_RASTERIZE". The first is a simple feature library that provides read and write access to a variety of vector file formats facilitating conversion between them [4]. The latter is used to burn vector data into existing raster files. The library needs to be downloaded and installed and can be addressed via command line. A batch file called "SDKShell.bat" comes with the installation and needs to be run first before executing the commands. It sets up the environment for using the command line tools. It runs on all modern Unix-based operation systems and most Windows versions.

– **Geobrain - Geospatial Web Services (CSISS) (Webservice)**
The Geobrain Geospatial Web Services are a collection of services to process and manipulate many geodata formats, e.g. satellite images, raster data, vector maps etc. Developed by the Center for Spatial Information Science and Systems of the George Mason University in Fairfax, Virginia, the services are SOAP based. In this workflow, the Service Raster Overlay is used to intersect both the raw and the generalized vector datasets.

– **IrfanView (Execute Command)**
The generated TIF file is opened in an instance of the small programm Irfanview. Irfanview needs to be installed, the programm is adressed under C:\Program Files (x86)\IrfanView\i_view32.exe.

3 Workflow Realization

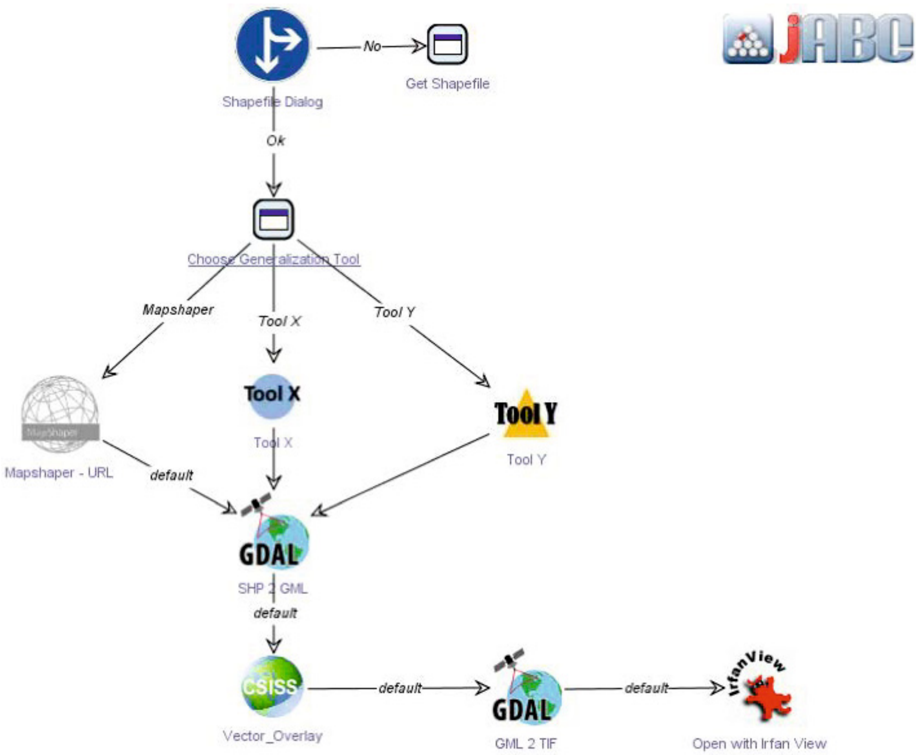

Fig. 2. jABC Workflow

Having the services characterized in the previous section, the realization of the actual workflow in jABC will now be described. It is a linear process with no submodels or loops. The only interaction with the user is necessary in the welcome dialog where the user has to choose the generalization tool (see Fig.2).

The workflow begins with a reminder to have a shapefile ready for the generalization (see Fig. 3) and if not, to provide one (ShowMessageDialog, see Fig. 4). What follows is a dialog window (Show.Branching.Dialog) in which the user is asked to choose a generalization tool to perform the workflow with.

Tools X and Y are represented by prototype SIBs which can later be replaced by other actual tools. The only executable tool is Mapshaper.

The generalization web service is then being executed through the Execute Command SIB "Mapshaper - URL". It simply addresses the command file "mapshaper.cmd" in a local folder which opens a Firefox instance with the mapshaper.org - URL (see Fig. 6).

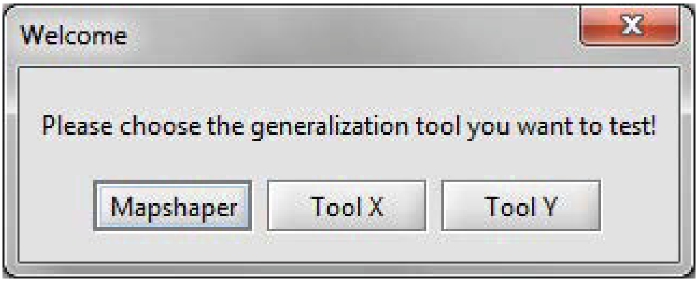

Fig. 3. Branching dialog: choose a generalization tool

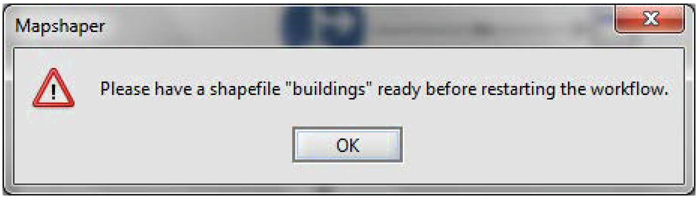

Fig. 4. Message dialog: please provide a shapefile

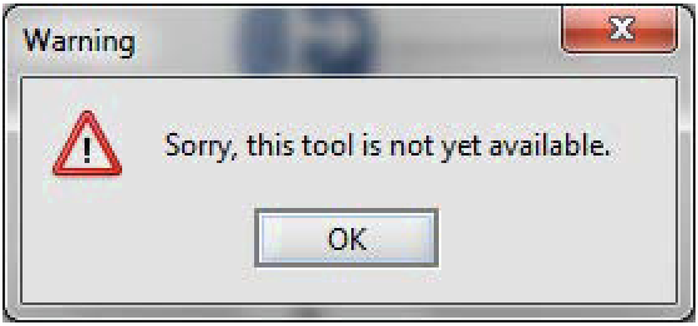

Fig. 5. Message dialog: tool not yet available

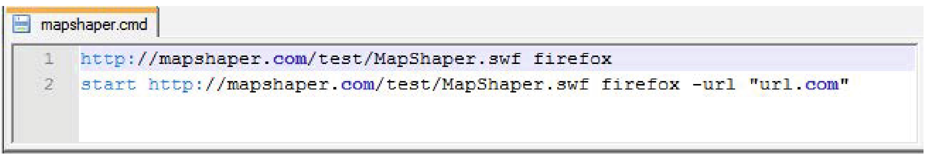

Fig. 6. Commandline: opens firefox browser with specific URL

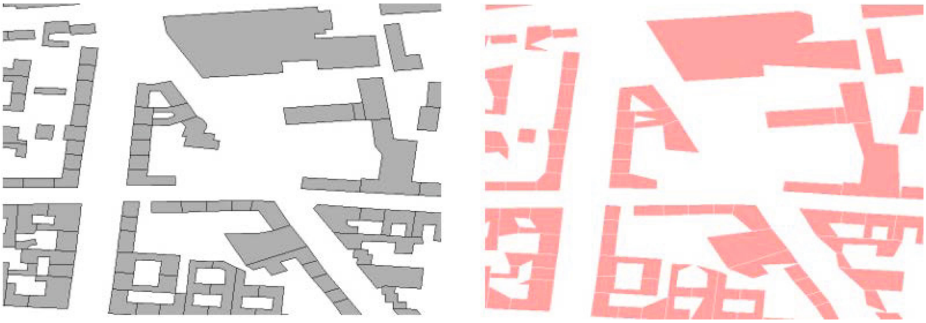

Fig. 7. Buildings before (left) and after generalization (right)

The result is a manipulated shapefile. In the example, a polygon Open Street Map building shapefile is used. The Douglas-Peucker algorithms and a 5%-simplification level is applied. Now, the interactive tool can be used and the output shapefile must be saved. It can be seen how outline vertices are reduced and polygons partly collapsed (see Fig. 7).

Unfortunately, there is no connection between this step which is still performed in the mapshaper.org web environment and the rest of the workflow, so it needs to be restarted manually at that point in the process.

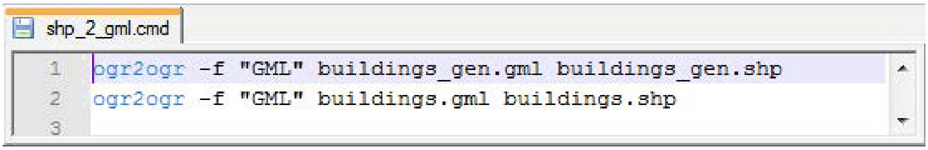

Fig. 8. Commandline: converts shapfiles to GML

In the next step, the shapefiles are converted to GML format. This is done by the command line translator OGR 2 OGR (see Fig. 8). The new files are saved in the same folder.

At this point comes the CSISS Vector Overlay SIB into play. Its service performs the overlay between two GML files by the operator OR which in the GIS world corresponds to the intersect command union. That means, two or more spatial layer are overlapped, both areas are kept and the spatially intersecting parts are determined. Polygons are separated at the intersection lines. Attributes are kept and written to the newly created polygons to later detect which parts have attributes of both input layers. Intermediate output is another GML file, in this case named buildings_union.gml.

Another conversion task is executed in the next step. Using the GDAL_ RASTERIZE command, the source GML file is rasterized into a target TIF file which

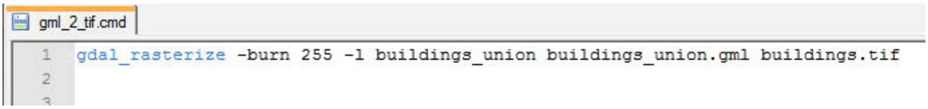

Fig. 9. Commandline: rasterizes GML to TIF

comes with a world file .TFW for spatial reference. The command line contains the following parameters [4] (see Fig. 9):

– burn: puts a fixed value into a band for all objects, in this case it is the value for red, 255.
– l: indicates the layers from the datasource that will be used for input features, both the GML and TIF file.

Fig. 10. Final output TIF as seen in Irfanview

Finally, the image is opened and displayed in the program Irfan View (see Fig. 10).

4 Conclusion

Webtools for visualization are numerous and the results already achieve high design standards. Examples like "prettymaps" (http://prettymaps.stamen.com/), "Open Street Map" (http://www.openstreetmap.org/) or "Polymaps" (http://www.polymaps.org/) give good results. Geodata is also increasingly available on the web, free and ready-to-use as one can easily connect to a geographic database and retrieve information upon request [2]. This is what most web services can offer and interchange standards are also set.

The only missing link from data acquisition to visualization in the pipeline of web based on-demand map production is the actual data processing. Transformation should alter the data geometry and/or semantics. That goes beyond the adaptive zoom function many visualization services offer. At present, free tools and services for geoprocessing and even generalization are rare, but might increasingly be developed as demand is rising. The development should benefit from progression in the other two fields.

As a general point of critisism can be stated that the workflow as presented was realized with just one generalization service, mapshaper.org. If more services were available, a real comparison could be performed at the end of the workflow. Different output datasets of the same input data could be contrasted with one another. The discrepancy of altered and/or collapsed polygon areas could be calculated and displayed. Since the principal concept of the designed workflow has been the comparison between different generalization tools on the web, implementing just one web service does not comply with the initial goal.

As for the concrete form of the current workflow implementation, the workflow could certainly embody more user interaction. Not only choosing a tool to be tested, but also setting its generalization parameters or selecting an algorithm and the input data would enhance the workflows usability. Unfortunately, the nature of the tool mapshaper.org does not allow programmatic access to its parameters in jABC, as it is a highly interactive flash environment. Maybe other tools will be easier to address and maybe some form of interface standards will be developed in the future.

Likewise, it would also be nice to include more interaction dialogues when converting the files. Input files and output file names and directories should be chosen by using the ShowFileChooser SIB. As another idea, a loop could be embedded between the viewing of the outcome in Irfanview and the generalization tool so that in case outputs are not satisfying, the process could be iterated.

Note that the CSISS SIB Vector Overlay is currently not working correctly which might be due to server errors or false implementation. The output TIF also needs alteration in terms of color management and displayed data. There should be an preceding selection command that filters the GML and assigns certain colors according to a SQL-statement.

To conclude the evaluation, we take a closer look at the Mapshaper-Tool. One important question is if the service delivered results that could be used in the further process. The outcome is a shapefile, a proprietary vector format set by the GIS program company ESRI. That format alone can be handled through conversion by other services which is seen as an advantage. But it also comes with great disadvantages: The outcome data of mapshaper.org misses spatial projection. That means no coordinate system is assigned even though the input data (shapefile buildings from Open Street Map) has one. That will cause problems a few steps later with the Vector Overlay, when two data layers should be spatially intersected. The SIB cannot handle unprojected data. Open Street Map data sometimes comes geometrically and/or topologically incorrect. For example, outline rings are not closed and polygons have zero area values. Fixing those data errors is not originally planned to be part of the workflow but should be included since it is crucial to have manageable data to continue the process. To work with the data in the present workflow, those errors were corrected separately in a desktop GIS. Maybe the data should be checked beforehand.

This article is part of a larger evaluation [10], which aimed at illustrating the power of simplicity-oriented development [17] by validating the claim that process modeling can indeed be handed over to the domain experts by providing them with a graphical modeling framework [24] that covers low-level details in a service-oriented fashion [19], integrates high-level modeling in the overall development process in a way that user-level models become directly executable [18,15], and supports ad-hoc adaptations and evolution [14,16].

The project described in this article can be characterized as follows:

- Scientific domain: geoinformatics
- Number of models: 1
- Number of hierarchy levels: 1
- Total number of SIBs: 9
- SIB libraries used (cf. [13]): common-sibs (8), csiss-sibs (1)
- Service technologies used: SOAP web services, web applications, command line tools

The geoinformatics part of this volume contains eight other articles on workflow applications in this domain [8,21,7,26,6,23,22,5]. Further geoinformatics workflow projects with the jABC have recently been started. Ongoing work is also exploring how to apply semantics-based (semi-) automatic workflow composition techniques (as provided by, e.g., [20]) to support the workflow design process, as described in [11,12,9] for the bioinformatics domain.

References

1. Arlinghaus, S.: Practical handbook of digital mapping: terms and concepts. CRC Press, Florida (1994)
2. Burghardt, D., Neun, M., Weibel, R.: Generalization Services on the Web Classification and an Initial Prototype Implementation. Cartography and Geographic Information Science 32(4), 257–268 (2005)

3. Center for Spatial Information Science and Systems (CSISS). CSISS Geospatial Web Services. George Mason University, Fairfax (2011)
4. GDAL Development Team. GDAL - Geospatial Data Abstraction Library, Version 1.9.1. Open Source Geospatial Foundation (2012)
5. Hibbe, M.: Spotlocator Project Documentation. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 154–163. Springer, Heidelberg (2014)
6. Holler, R.: GraffDok: A Graffiti Documentation Application. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 235–247. Springer, Heidelberg (2014)
7. Kind, J.: Creation of Topographic Maps. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 225–234. Springer, Heidelberg (2014)
8. Kuntzsch, C.: Visualization of Data Transfer Paths. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 140–148. Springer, Heidelberg (2014)
9. Lamprecht, A.-L.: User-Level Workflow Design. LNCS, vol. 8311. Springer, Heidelberg (2013)
10. Lamprecht, A.-L., Margaria, T.: Scientific Workflows and XMDD. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 1–13. Springer, Heidelberg (2014)
11. Lamprecht, A.-L., Margaria, T., Steffen, B.: Bio-jETI: a framework for semantics-based service composition. BMC Bioinformatics 10(suppl. 10), S8 (2009)
12. Lamprecht, A.-L., Naujokat, S., Margaria, T., Steffen, B.: Semantics-based composition of EMBOSS services. Journal of Biomedical Semantics 2(suppl. 1), S5 (2011)
13. Lamprecht, A.-L., Wickert, A.: The Course's SIB Libraries. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 30–44. Springer, Heidelberg (2014)
14. Margaria, T., Steffen, B.: Agile IT: Thinking in User-Centric Models. In: Margaria, T., Steffen, B. (eds.) Leveraging Applications of Formal Methods, Verification and Validation. CCIS, vol. 17, pp. 490–502. Springer, Heidelberg (2009)
15. Margaria, T., Steffen, B.: Business Process Modelling in the jABC: The One-Thing-Approach. In: Cardoso, J., van der Aalst, W. (eds.) Handbook of Research on Business Process Modeling. IGI Global (2009)
16. Margaria, T., Steffen, B.: Continuous Model-Driven Engineering. IEEE Computer 42(10), 106–109 (2009)
17. Margaria, T., Steffen, B.: Simplicity as a Driver for Agile Innovation. Computer 43(6), 90–92 (2010)
18. Margaria, T., Steffen, B.: Service-Orientation: Conquering Complexity with XMDD. In: Hinchey, M., Coyle, L. (eds.) Conquering Complexity, pp. 217–236. Springer, London (2012)
19. Margaria, T., Steffen, B., Reitenspieß, M.: Service-Oriented Design: The Roots. In: Benatallah, B., Casati, F., Traverso, P. (eds.) ICSOC 2005. LNCS, vol. 3826, pp. 450–464. Springer, Heidelberg (2005)
20. Naujokat, S., Lamprecht, A.-L., Steffen, B.: Loose programming with PROPHETS. In: de Lara, J., Zisman, A. (eds.) Fundamental Approaches to Software Engineering. LNCS, vol. 7212, pp. 94–98. Springer, Heidelberg (2012)
21. Noack, F.: CREADED: Coloured-Relief Application for Digital Elevation Data. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 182–195. Springer, Heidelberg (2014)

22. Respondeck, T.: A Workflow for Computing Potential Areas for Wind Turbines. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 196–211. Springer, Heidelberg (2014)
23. Scheele, L.: Location Analysis for Placing Artificial Reefs. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 212–224. Springer, Heidelberg (2014)
24. Steffen, B., Margaria, T., Nagel, R., Jörges, S., Kubczak, C.: Model-driven development with the jABC. In: Bin, E., Ziv, A., Ur, S. (eds.) HVC 2006. LNCS, vol. 4383, pp. 92–108. Springer, Heidelberg (2007)
25. Stern, B., Hurni, L., Werner, M., Wiesmann, S.: Generalisation of Map Data (2011), http://www.gitta.info/Generalisati/en/text/Generalisati.pdf
26. Teske, D.: Geocoder Accuracy Ranking. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 159–170. Springer, Heidelberg (2014)