Approaches for enhancing tile-based mapping with cartographic generalisation

Ralf Klammer, Dirk Burghardt

Institute for Cartography, Dresden University of Technologies, 01062 Dresden, Germany
E-mail: {ralf.klammer, dirk.burghardt}@tu-dresden.de

1. Motivation

As part of the progression of information age, internet users are increasingly able to participate actively in the appearance and content of the internet, what has been resulted in the term "Web 2.0" (O'Reilly 2005). As a result, some projects have been initiated prescribing the collection of spatial data with the help of a crowd, what is now known mainly as crowdsourcing (Martin et al. 2008). That evolved a kind of "neogeography" (Turner 2006) respectively "neocartography" (Gartner and Schmidt 2010) which expresses in form of "maps 2.0" (Crampton 2009). OpenStreetMap[1] is probably the most well known project in this context and is meanwhile well established. Now a large number of maps exist, that are derived solely from that user-generated (Goodchild 2007) "geomassdata" (Zipf 2009). But many of these maps can only be used with low requirements to the graphical quality (Zollinger 2008). One reason for this can be seen in a lack of applicability of cartographic generalisation. Probably the most widely used web mapping method is the representation in form of interactive, "free" scalable map displays, colloquial called "slippy map" (e.g.: OpenPisteMap[2], OpenCycleMap[3], WheelMap[4], OpenTopoMap[5], etc.). These are usually created completely automatic and displayed on the internet as tile-based map.

The processing of such huge datasets needs adequate spatial data structures for fast and efficient data processing respectively generalisation. This is regularly a tile-based structure for slippy maps. Cartographic generalisation has to fit to this structure, especially for automatic computation, update and on demand request of tiles. Changes in the corresponding data as well as the query of single tiles happen within an heterogeneous spatial frequency. Frequently obtained tiles get processed regularly and scarce obtained tiles are only processed on demand. In consequence, generalisation should just be applied to local changes and not to the whole dataset. Therefore, it is important to enhance the automatic computation of tiles, colloquial called "rendering", with concepts of on-the-fly generalisation, for keeping the tile-based mapping process flow completely automatic.

However, it must be considered that the claim, of keeping the process flow automatic, cannot be fully satisfied for all requirements of cartographic generalisation. For this reason, it is necessary to include also supporting concepts of data pre-processing, like MRDB, in the deliberations.

1 openstreetmap.org
2 openpistemap.org
3 www.opencyclemap.org
4 wheelmap.org
5 opentopomap.org

In summary, the general claim of this paper is to analyse the specific impacts of tile-based mapping on the applicability of automatic cartographic generalisation with an examination of possible architectures for enhancing tile-based mapping with cartographic generalisation subsequently. Thus this paper provides the basis for further research activities which will be focused on the development and integration of appropriate generalisation strategies in the context of tile-based mapping.

2. State of the art

Tile-based mapping is a very new technology in cartography, which is fundamentally based on the quadtree data structure (Samet 1990). Only very few publications exist on tile-based mapping currently. Most authors just describe and analyse the whole process flow of tile-based mapping systems (Sample and Ioup 2010; Jurk 2010; Smith 2008) or show how to implement such a complex system (Kunz 2012; Hearn 2011; Naumann 2010). As these are related to the processing of geospatial information, tile-based mapping is just one facet of their descriptions. The publication “Tile-Based Mapping Transitions in Cartography” (Peterson 2012) has the claim to give a first impression of the implications that occur with that new technology for cartography. In result, it is mainly also a description of the basic technology, at least with the focus on cartography. Aspects of digital cartographic generalisation have not been analysed so far.

This is also true for publications related to the cartographic generalisation of user-generated geoinformation. Zollinger (2008) did a common analysis of one specific OpenStreetMap-Visualisation, while the application of cartographic generalisation was just one aspect. He identified a variety of graphical lacks which are partially the result of missing generalisation (e.g.: Zollinger 2008: 20). In conclusion, only fundamental operations like semantic selection and symbolisation are applied to automatic tile-based mapping.

The concept of “on-the-fly-generalisation” is mainly used in the context of interactive cartographic applications, as it applies techniques of automatic generalisation in real time (Weibel and Burghardt 2008). Therefore it is predestined to enhance tile-based mapping with automatic generalisation. On principle, all currently existing algorithms for generalisation that run linear or logarithmic could be applied in this context (Glover and Mackaness 1999, Sarjakoski and Sarjakoski 2004). As most of them are not efficient enough for real time generalisation, there is still much need to develop fast respectively adapted algorithms. Additionally, on-the-fly-generalisation is not applicable to all arising generalisation tasks. Therefore some complementary publications combine on-the-fly with pre-processing. On the one hand spatial data structures can be used to support on-the-fly-generalisation (Oosterom 1989, Oosterom 1995). That has already been implemented by using hierarchical data structures (Burghardt et al. 2004, Mustafa et al. 2006) or by using reactive data structures (Petzold 2003). On the other hand the concept of multiple representation databases (Kilpeläinen 1992) was already used to realise on-the-fly-generalisation (Sederberg and Greenwood 1992, Neun et al. 2004, Bernier et al. 2005, Cheng et al. 2009). Comprehensive concepts, which refer especially on the impacts of tile-based mapping and the corresponding applicability of on-the-fly-generalisation are not existing so far.

Recent developments aim at the deployment of interoperable systems in the context of cartographic generalisation, with the intention of providing generalisation functionalities on the internet (Edwards et al. 2003, Neun and Burghardt 2005, Foerster 2010). This is based on Web Processing Services (WPS) standardised by the OGC (Open Geospatial Consortium). WebGen-WPS (Foerster et al. 2008) enables a generic generalisation of spatial data via the internet, independent of platform or processing language. As the main goal is to develop a generic framework for cartographic generalisation, this technology is assumed to pre-compute data.

3. Cartographic generalisation and tile-based mapping

The term "tile-based mapping" is neither in geographic information sciences nor in scientific cartography an established expression. Therefore this chapter starts by a term definition. The essential parts for implementing generalisation are identified simultaneously. Afterwards, special impacts for the applicability of generalisation to that technology are described.

3.1 Tile-Based Mapping

The term "*tile-based*" contains the properties of the basic visualisation method, signifying the display of an overall picture based on the combined representation of single tiles. A technique that has for a long time only been used to efficiently convey computer graphics (Sample and Ioup 2010). Accordingly, the combined visualisation of a digital map is called "*tile-based map*", or colloquial "slippy map." Since Google has implemented this type of digital map display in 2005, it was very well received by the users. Even without an appropriate technological background, it is by now familiar to nearly everyone who is using modern internet technologies. The biggest advantage is the ability to scale and shift an interactive digital map display. In result, the user can view a scalable and shiftable map. Especially this property can be understood as the origin of the mainly used term "slippy map."

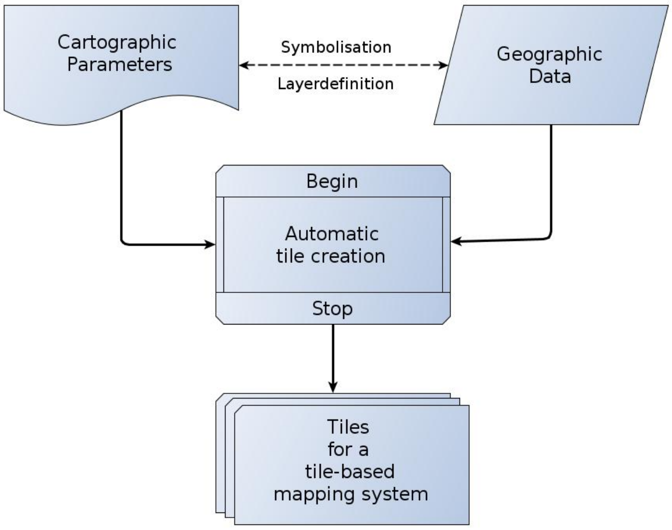

Figure 1: Schema for Tile-Based Mapping.

"*Tile-based mapping*" indicates the process of creating the required tiles for a corresponding web map (see Figure 1). That term outlines both all manual configuration steps as well as the automatic tile computation, colloquial named "rendering". The map editor defines the cartographic parameters containing rule-based information about the cartographic symbolisation and the virtual storage location of data sources. This is typically defined in a structured text file, e.g.: XML (Extensible Markup Language), comparable to the SLD- (Styled Layer Descriptor - OGC 2007) and SE-Standards (Symbology Encoding - OGC 2006). At this point manual "cartographic generalisation" (Grünreich 1985), like semantic selection or symbolisation, is feasible (see Figure 2).

These abstract manual operations imply a huge amount of mental processes performed by the cartographer, subsequently referred to as cognitive performance. Simultaneously the data sources can be prepared manually in dependence of requirement. That preparation necessitates manual or automatic geometric pre-generalisation (see Figure 2), perceivable as “model generalisation”. Afterwards the editor initialises the rendering software with the cartographic parameters and starts the rendering. The computation of a set of tiles is subsequently done iteratively and fully automated by the renderer (cf. Figure 1). At this point only automatic geometric generalisation is applicable, as no manual user intervention is intended to the standard workflow.

The entire display area has to be divided into a logical structure of tiles for each predefined scale (“zoomlevel”). The corresponding WMTS-Standard (Web Map Tile Service) for this decomposition was defined by the Open Geospatial Consortium (OGC 2010). It is based on the unofficial TMS-Standard (Tile Map Service), which was defined by the Open Source Geospatial Foundation (OSGeo 2006a). Parallel an unofficial standard for visualising tile-based maps via Web Map Services (WMS) was invented, the so called WMS Tiling Client Recommendation (WMS-C) (OSGeo 2006b).

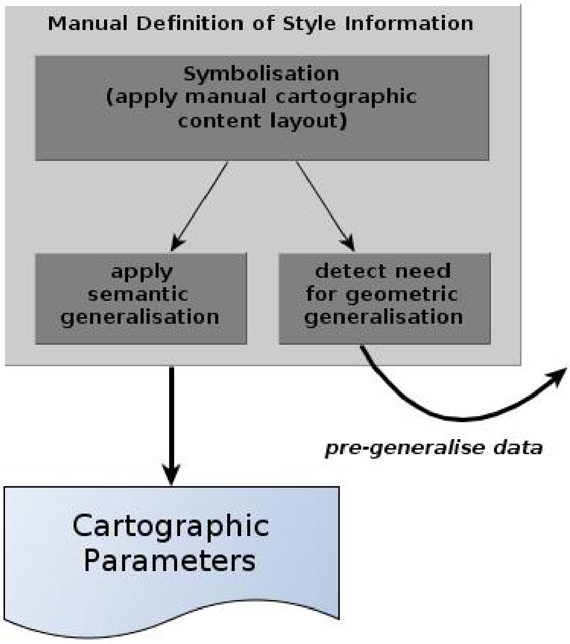

Figure 2: Schema for editing cartographic parameters.

Finally, the whole process chain of tile-based mapping and the following visualisation of a tile-based map is summarised to the term „*tile-based mapping system*.“ It contains all necessary aspects beginning with rendering tiles over storage management up to the final display via API (application programming interface).

3.2 Impacts on the applicability of generalisation

Describing the impacts of tile-based mapping to the applicability of generalisation, implies the need to distinguish between two basic differences for influencing generalisation. On the one hand there are impacts by the whole process flow of tile-based mapping systems. These are mainly aspects like rendering performance, tile storage or server technologies. These aspects have no impact on the applicability of generalisation but on the decision whether generalisation at all and which concept could be applied. On the other hand, there are impacts by the fundamental technology of tile-based mapping. Meaning, the computation of small tiles for a well defined area in a well defined scale to assemble them afterwards as an overall map display, as illustrated in Figure 3. The solitary rendering of map-tiles has direct impacts on the applicability of generalisation, precisely on the applicability of generalisation operators (e.g.: displacement, simplification, etc.). Researching these impacts will be subject of this sub chapter.

The first main restriction of tile-based mapping is the treatment of geometries that belong to more than one tile or are located at the tile-border (see Figure 3). That might arise bridging errors when the geometry of an object is modified differently in neighboured tiles. For example when simplifying linear geometries with a Douglas-Peucker-Algorithm (Douglas and Peucker 1973), different results can occur for each tile. Adequate solutions could be using a buffered area or by rendering a super-tile.

Buffering the tile computation is a solution that is already used in practical implementations. A tile is not rendered directly on its final size (e.g.: 256x256 pixel) but the most renderer use a buffer (e.g.: 128 pixel). Mainly the buffer size is adjustable. After rendering the tile, the rendered area will be trimmed to the final tile size. This induces some general questions: How huge should such a buffer area be, to ensure that no bridging areas or other problems occur? Should the buffer size be fixed or set situation dependant? The answers are not trivial and have to be found empirical.

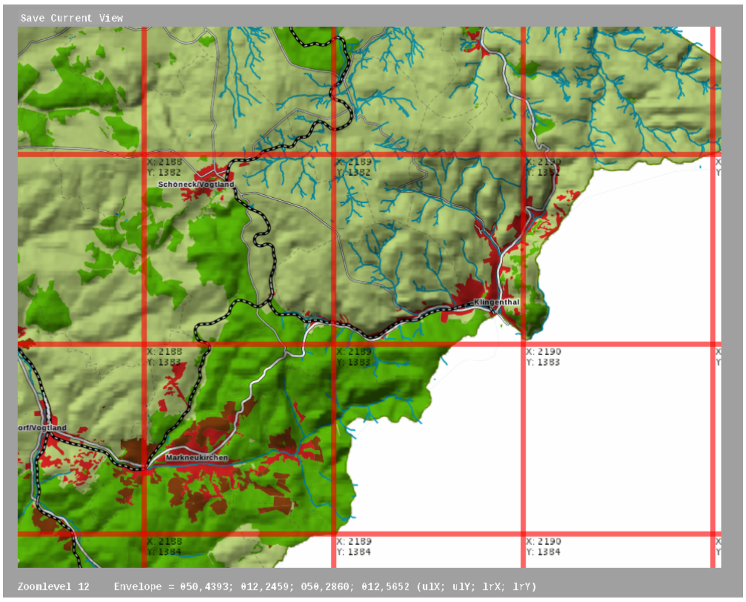

Figure 3. Tiles of a tile-based map – (data: OpenStreetMap).

Rendering super-tiles means, rendering a huge tile (e.g.: 2560x2560 pixel) in a first step and segment this to sub-tiles of aimed size (e.g.: 100 tiles à 256x256 pixel) afterwards. Kunz (2012) describes this solution to avoid problems in label placement. This approach significantly delimits the performance of the renderer as it needs additional storage and implies the question on how huge the super-tile should be. An additional research question is: Isn't the problem relocated to the border of the super-tile? This can be prevented by not using all segmented tiles (e.g.: use only central segments) and creation of overlapping super-tiles.

The second major restriction is the consideration of spatial context (Mustière and Moulin 2002). Spatial context describes the relation of an object to its surroundings. On the one hand it describes if objects are part of a group or inside a particular area (hierarchical relations). This is especially important for generalisation operators like aggregation. On the other hand spatial context specifies local topological relations between objects (non hierarchical relations), which are fundamental for operators like displacement. Some of these topological relations have to be kept and some can be modified. Observing spatial context is a challenging task in the context of tile-based mapping.

First, the renderer has to be able to handle spatial context. For example, detect if an object is part of a significant group, lies inside a particular area or if it conflicts a neighboured object. Existing rendering software solutions are not capable of that (see chapter 4.1).

Second, some questions in relation to the tile borders have to be clarified. Should objects located in neighboured tiles be ignored or observed? Should objects located in neighboured tiles be influenced respectively modified? And how far does the influence on neighboured objects reach? These questions cannot be answered globally. Since different generalisation operations have to observe different spatial context relations, it depends on the applied operation. An adequate approach, for integrating generalisation, should be able to process every generalisation operator equally.

4. Possible architectures

This chapter describes possible architectures to enhance tile-based mapping with automatic cartographic generalisation. We start by specifying an approach for the implementation of on-the-fly-generalisation. Afterwards the implementation of MRDB as a supporting tool is described.

4.1 On-the-fly-generalisation

The term "on-the-fly-generalisation" describes mainly the computation of temporary generalised datasets, only for the visualisation without additionally storing them (Oosterom 1995). On tile-based mapping this can be done while rendering data but also while querying data from a database.

As described in chapter 3.2, the renderer has to be able to handle spatial context in order to apply automatic generalisation. Therefore it is necessary to have a closer look at the process flow of automatic tile rendering (see Figure 4). Tile rendering is an automatic, iterative process, where all features of one layer, all layers of one tile and all tiles of one tile-based map are processed successively.

Generalisation operators have to consider topological neighboured objects both of the same (e.g.: aggregation) as well as of different layers (e.g.: displacement). Existing renderer implementations can perform simple topological query to features of one layer. For example, mapnik[6] is able to avoid overlapping of point or text placement. Avoiding overlap of complex geometries, like lines or polygons, is currently not implemented. Additionally, context relations across different layers and semantic information are not considerable. For example, to detect the overlapping of streets and buildings.

In consequence, the process of automatic rendering has to be modified. The corresponding question is, how an adequate architecture should look like. A modified renderer would get a bunch of information on how, when and which objects have to be generalised. These information could be defined conceptional in the cartographic parameters. Accordingly, requirements on the cognitive performance of the cartographer will arise, as this is an additional abstract manual process. The renderer should observe

6 mapnik.org

topological relations independently and process the resulting operations automatically. Required algorithms could be: (1) implemented to the rendering software, (2) called by the renderer from an external library or (3) provided by web generalisation services. It is adversely, that for case (1) the required algorithms have to be written in the same programming language as the implementation of the renderer. The same is true for external libraries. Using web generalisation services is a good generic solution but reduces rendering performance, as the data processing via internet extends processing time.

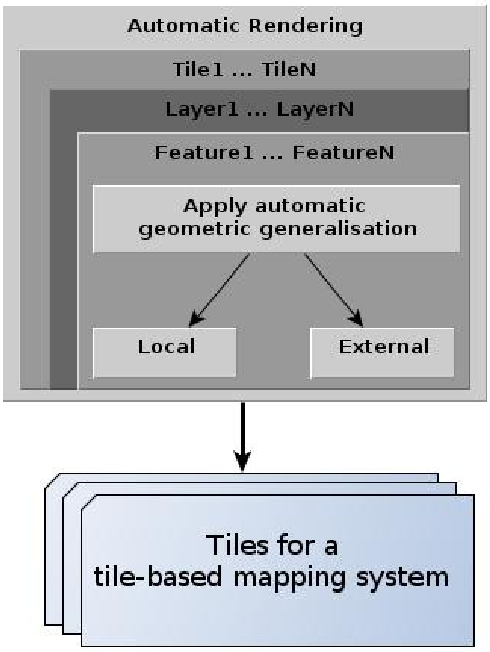

Figure 4. Schema for automatic tile rendering.

It is obvious that the implementation of this architecture is very complex. An option to solve simple generalisation tasks, is the generalisation of objects while querying them from a database. Especially PostGIS[7], the spatial addition for PostgreSQL[8], offers the possibility to implement functions, that could also do generalisation tasks. Two elementary functions for simplifying geometries are already included (ST_Simplify()[9] and ST_SimplifyPreserveTopology()[10]). But these are just quite simple implementations of the well known Douglas-Peucker-Algorithm. The potential lies in the development of complex, sql-implemented functions and integration to the data queries of automatic processing. PostGIS offers a wide range of functions for observing spatial relationships and measurements as well as geometry processing functions. These can be used to implement generalisation operations and handle spatial context of objects.

4.2 Multiple Representation Database System

The second proposed architecture is based on the concept of pre-computing objects and saving them as multiple forms of representation in a database system (Kilpeläinen 1992). Preparing different levels of generalised objects for each required scale is not

7 postgis.refractions.net
8 www.postgresql.org
9 http://postgis.org/docs/ST_Simplify.html
10 http://postgis.org/docs/ST_SimplifyPreserveTopology.html

effected by tile-based generalisation constraints. There is no need for developing new strategies or technologies because manual or automatic classical procedures can be adopted. But this concept can be used as an aiding tool, to support generalisation operators that are difficulty applicable to on-the-fly generalisation.

A corresponding process flow (see Figure 5) would be: (1) load original data into a geoinformation system (2) generalise data via WPS and (3) save results in a MRDB. Redundancy can be acceptable when it generates performance enhancements. But it is advantageous to avoid redundancy in the database by saving data in additional tables without meta data and reference them to the original object by an unique identification number (ID). This can be carried out for different generalisation operations, always in relation to the pursued scale. Subsequently, the queries of the additional tables have to be included to the cartographic parameters. This can be done in the original layer, as sql-join in the select-statement, or as an additional layer. In result, the renderer can process generalised objects automatically when it is necessary. Some corresponding results will be demonstrated in the following chapter.

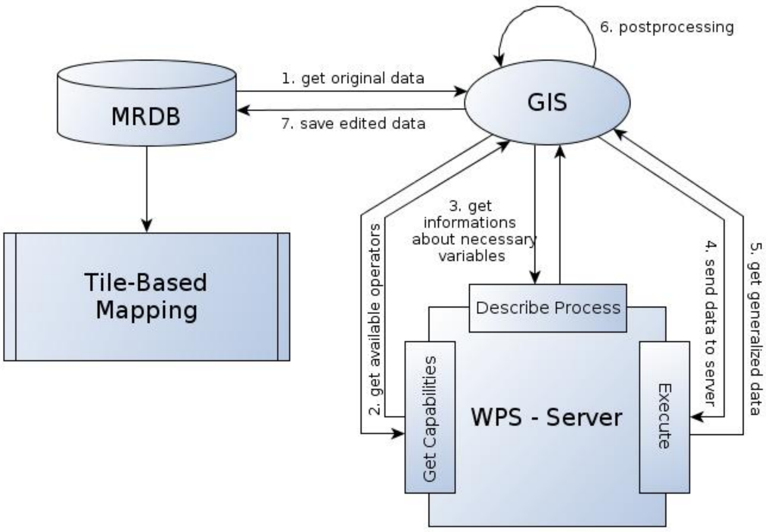

Figure 5. MRDB - proposed process flow.

5. Exemplary implementation of proposed architectures

In order to prove the proposed architectures as well as a general usability of WebGen-WPS, the proposed architectures were put into practice. Hence, this chapter offers a condensed description of the practical implementations.

Only open source software and free, user-generated spatial data was utilised for this proof-of-concept. Mapnik was chosen for rendering, as it is the most common and universal open source software for rendering tiles. Alternative rendering software can also be used, but is either especially designed for the OpenStreetMap project (e.g.: Kosmos[11], Pyrender[12]) or is not (by default) applicable to tile-based mapping (e.g.: Osmarender[13]). The data was taken from the central OpenStreetMap database, which provides free available user-generated topographical geoinformation. That data can be imported to a spatial extended PostgreSQL database (PostGIS) with the help of osm2pgsql[14].

11 wiki.openstreetmap.org/wiki/Kosmos
12 wiki.openstreetmap.org/wiki/Pyrender
13 wiki.openstreetmap.org/wiki/Osmarender
14 wiki.openstreetmap.org/wiki/Osm2pgsql

There are also alternative free available database management systems with the option to add spatial extensions, like MySQL[15] or SQLite[16]. The data import could also be done with similar open source software, like osmosis[17], imposm[18] or the like. Data preparation and pre-processing can be done with OpenJump[19]. That open source GIS was chosen because it offers the possibility to communicate with PostGIS and WebGen-WPS by additional plug-ins. WebGen-WPS was used to generically apply geometric generalisation operations. Finally, the on-the-fly architecture was implemented in python[20], especially because mapnik has python-bindings. But also because of the WPS-communication by PycURL[21] and the simple possibility of creating graphical user interfaces with GTK+[22].

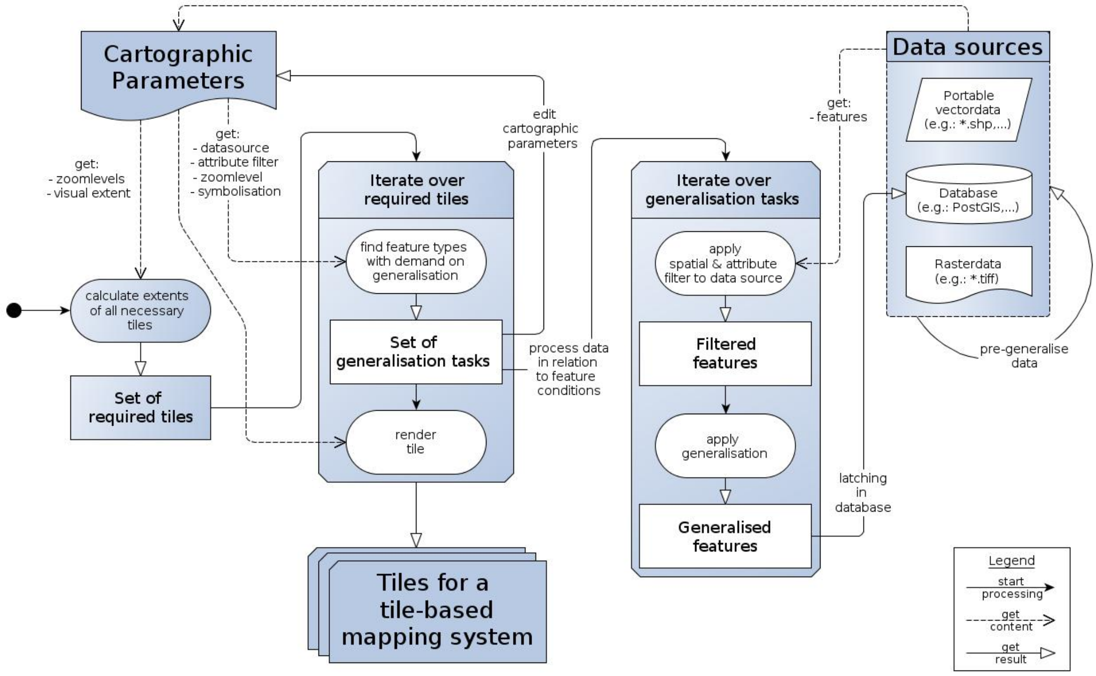

Figure 6. Process flow of exemplary implementation

Figure 6 visualises the basic elements of the realised implementation in a simplified form. The MRDB-architecture was realised as illustrated in Figure 5, therefore it is just indicated by “pre-generalise data”. The on-the-fly architecture can be realised by using the predefined cartographic parameters of the tile-based rendering whereby features can be analysed and manipulated in relation to their symbolisation. The symbolisation of each feature type contains information about its demand on generalisation. These information are extracted along with the feature conditions, to filter the original data source and generalise the filtered features. Finally, the generalised features are latched in a database and the information about the memory location get included to the cartographic parameters. These steps are done for each tile, whereby a tile-based on-the-fly generalisation can be realised.

15 www.mysql.com
16 www.sqlite.org
17 wiki.openstreetmap.org/wiki/Osmosis
18 imposm.org/docs/imposm/latest
19 www.openjump.org
20 www.python.org/
21 pycurl.sourceforge.net
22 www.gtk.org/

The test proves the fundamental feasibility of on-the-fly-generalisation as well as the MRDB-architecture as aiding tool. It also revealed some basic problems in the applicability of the WebGen-WPS, which have to be modified for a practical use. These are mainly problems while processing large data sets and a missing option to transfer attributes. Additionally it has been shown that the processing via WebGen-WPS is relative slow, what causes huge restrictions for it's applicability of on-the-fly-generalisation. A first analysis has shown that only a small part of processing time depends on the number of processed features (see Table 1). This fact has to be investigated in further work.

Table 1. Analysis of processing time for WebGen-WPS

Tile coordinates	Zoom	Number of geometries	Geometry type	Processing time (in seconds)
X: 546 Y: 344	10	6	LINESTRING	69.292
X: 547 Y: 344	10	8	LINESTRING	68.261
X: 546 Y: 345	10	126	LINESTRING	70.218
X: 547 Y: 345	10	851	LINESTRING	74.145
X: 546 Y: 346	10	7	LINESTRING	67.262
X: 547 Y: 346	10	19	LINESTRING	68.266

6. Discussion and further work

This paper provides basic insights to the applicability of automatic cartographic generalisation in the context of tile-based mapping. It has been described, that tiling of web maps has direct impacts on the applicability of generalisation operations, which are primarily related to the consideration of spatial context. An appropriate architecture for the implementation of on-the-fly-generalisation has been proposed wherefore the tile rendering has to be modified. Additionally the implementation of the proposed architectures and it's applicability has been described.

There is still a huge number of future research questions. For example, the special requirements of label placement have to be analysed. Furthermore it is necessary to investigate the properties of user-generated geoinformation and their impact on the applicability of generalisation. The influence of the whole process flow of tile-based mapping systems, such as computational capacity, server technology, bandwidth or storage capacity (data and tiles), has to be considered in further work. Possibilities to generalise data in relation to their symbolisation (style information) have to be investigated. The manual configuration of style information implies much cognitive performance and happens rule-based. Therefore it is necessary to establish concepts for improving the ability to apply manual cartographic generalisation (e.g.: semantic selection) while preparing the cartographic parameters. Finally, it has to be analysed in how far parallel processing and local implementations can contribute to accelerate the feature generalisation via WPS.

References

Bernier, E.; Bédard, Y.; Hubert, F. (2005). UMapIT: An On-Demand Web Mapping Tool Based On A Multiple Representation Database. In: *8th ICA Workshop on Generalisation and Multiple Representation.*

Burghardt, D.; Purves, R.; Edwardes, A. (2004). Techniques for on-the-fly generalisation of thematic point data using hierarchical data structures. In: *Proceedings of the GIS Research UK 12th Annual Conference*, UK, p.60-63.

Cheng, C.; Lu, F.; Cai, J. (2009). A quantitative scale-setting approach for building multi-scale spatial databases. Computers & Geosciences, 35:2204–2209.

Crampton, J. W. (2009). Cartography: maps 2.0. In: *Progress in Human Geography 33*, p.91-100. Available at: http://phg.sagepub.com/cgi/doi/10.1177/0309132508094074, last call: 30.04.2012.

Douglas, D. H.; Peucker, T. K. (1973). Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. In: *The Canadian Cartographer*, 10(2):112-122.

Edwards, A.; Burghardt, D.; Bobzien, M.; Harrie, L.; Lehto, L.; Reichenbacher, T.; Sester, M.; Weibel, R. (2003). Map Generalisation Technology: Addressing the need for a common research platform. In: *Proceedings of the 21st International Cartographic Conference (ICC)*, Durban, South Africa, 10-16 August.

Foerster, T.; Burghardt, D.; Neun, M.; Regnauld, N.; Swan, J.; Weibel, R. (2008). Towards an Interoperable Web Generalisation Services Framework – Current Work in Progress. In: *Proceedings 11th ICA Workshop on Generalisation and Multiple Representation*, Montpellier.

Foerster, T. (2010). Web-based architecture for on-demand maps – integrating meaningful generalization processing. Enschede: International Institute for Geo-Information Science and Earth observation, Enschede, Dissertation.

Gartner, G.; Schmidt, M. (2010). Moderne Kartographie-Technologische Entwicklungen und Implikationen. Kartographische Nachrichten, Kirschbaum, Bonn, 60(6):299-305.

Glover, E.; Mackaness, W. A. (1999). Dynamic generalisation from single detailed database to support web based interaction. In: *Proceedings 19th Int Cartographic Conference*, Ottawa, p.1-8.

Goodchild, M. F. (2007). Citizens as sensors. the world of volunteered geography. In: *GeoJournal*, 69(4):211-221.

Grünreich, D. (1985). Untersuchungen zu den Datenquellen und zur rechnergestützten Herstellung des Grundrisses großmaßstäbiger topographischer Karten. Hannover: Universität Hannover, WAVH Nr.132, Dissertation.

Samet, H. (1990). The Design and Analysis of Spatial Data Structures. Addison-Wesley, Reading, MA.

Hearn, B. (2011). Tile-based Mapping - Client or Server?. blog entry at MapDotNet (November 15, 2011), http://www.mapdotnet.com/index.php?view=entry&year=2011&month=11&day=14&id=55%3Atile-based-mapping-client-or-server&option=com_lyftenbloggie, last call: 26.04.2012.

Jurk, F. (2010). Integration von Tiled Map Services in Geodateninfrastrukturen. Diplomarbeit, Hochschule für Technik und Wirtschaft, Dresden.

Kipeläinen, T. (1992). Multiple Representations and Knowledge-Based Generalization of Topographical Data. In: T.C. Waugh & R.G. Healey: *Advances in GIS Research, Proceedings of the 6th Inernational Symposium on Spatial Data Handling*, IGU Commission on GIS and Association for Geographic Information, Edinburgh, GB, p.882-893.

Kunz, P. (2012). Ableitung von topographischen Webkarten aus amtlichen Geobasisdaten. FOSSGIS'12.

Martin, N.; Lessmann, S.; Voß, S. (2008). Crowdsourcing: Systematisierung praktischer Ausprägungen und verwandter Konzepte. In: *Proceedings of the 2008 Multi-Konferenz Wirtschaftsinformatik (MKWI `08)*, Munich.

Mustafa, N. H.; Krishnan, S.; Varadhan, G.; Venkatasubramanian, S. (2006). Dynamic simplification and visualization of large maps. International Journal of Geographical Information Science, p.273-302.

Mustière, S., Moulin, B. (2002). What is spatial context in cartographic generalisation? In: *Symposium on Geospatial Theory, Processing and Applications*, conference paper, Ottawa.

Naumann, F. (2010). Implementierung Tile Caching fähiger Map Clients. Diplomarbeit, Hochschule für Technik und Wirtschaft, Dresden.

Neun, M.; Weibel, R.; Burghardt, D. (2004). Data Enrichment for Adaptive Generalisation. In: *Proceedings 7th ICA Workshop on Generalisation and Multiple Representation*.

Neun, M.; Burghardt, D. (2005). Web Services for an open generalisation research platform. In: *8th ICA Workshop on Generalisation and Multiple Representation*.
O'Reilly, T. (2005). What Is Web 2.0? - Design Patterns and Business Models for the Next Generation of Software. O'Reilly Media, URL:http://www.oreilly.de/artikel/web20.html.
OGC (2006). Symbology Encoding Implementation Specification. Version: 1.1.0, http://portal.opengeospatial.org/files/?artifact_id=16700, last call: 08.05.2012.
OGC (2007). Styled Layer Descriptor profile of the Web Map Service Implementation Specification. Version: 1.1.0, http://portal.opengeospatial.org/files/?artifact_id=22364, last call: 08.05.2012.
OGC (2010). Web Map Tile Service Implementation Standard. Version: 1.0.0, http://portal.opengeospatial.org/files/?artifact_id=35326, last call: 26.04.2012.
Oosterom van, P. (1989). A reactive data structure for geographic information systems. In: *Proceedings of Auto-Carto 9*, p.665-674.
Oosterom van, P. (1995). The GAP-tree, an apporach to „On-the-fly" Map Generalization of an Area Partitioning. In: J.C. Müller, J. P. Lagrange and R. Weibel (eds.). *GIS ans Generalization: Methodology and Practice*, Taylor and Francis, London, p.120-132.
OSGeo (2006a). Tile Map Service Specification. Version: 2006, http://wiki.osgeo.org/wiki/Tile_Map_Service_Specification, last call: 26.04.2012.
OSGeo (2006b). WMS Tiling Client Recommendation. Version: 2006, http://wiki.osgeo.org/wiki/WMS_Tiling_Client_Recommendation, last call: 26.04.2012.
Peterson, M. P. (2012). The Tile-Based Mapping Transition in Cartography. In: L. Zentai, J. R. Nunez (eds.): *Maps for the Future, Children, Education ans Internet*. Lecture Notes in Geoinformation and Cartography, 5:151-163.
Petzold, I. (2003). Beschriftung von Bildschirmkarten in Echtzeit – Konzept und Struktur. Dissertation, Institut für Kartographie und Geoinformation der Universität Bonn.
Sample, J. T. & Ioup, E. (2010). Tile-Based Geospatial Information Systems. Springer Science+Business Media.
Sarjakoski, L. T.; Sarjakoski, T. (2004). A Use Case Based Mobile GI Service with Embedded Map Generalisation. In: 7*th ICA Workshop on Generalisation and Multiple Representation*..
Sederberg, T. W.; Greendwood, E. (1992). A physically based approach to 2–D shape blending. Computer Graphics, 26:25-34.
Smith, P. (2008). Take Control of Your Maps. A List Apart Magazine and the authors. (No. 256, April 8), http://www.alistapart.com/articles/takecontrolofyourmaps, last call: 22.03.2012
Turner, A. J. (2006). Introduction to Neogeography. Schort Cuts, O'Reilly Media, Inc.
Weibel, R.; Burghardt, D. (2008). On-the-fly Generalization. In: Shekhar, S. & Xiong, H. (eds.). *Encyclopedia of GIS*, Springer-Verlag, p.339-344.
Zipf, A. (2009). Nutzungspotentiale und Herausforderungen von "Volunteered Geography" - Zur Kombination von GDI-Technologie und nutzergenerierten Geomassendaten. In: Kriz, K.; W. Kainz und A. Riedl (Hrsg.) *Geokommunikation im Umfeld der Geographie, Wiener Schriften zur Geographie und Kartographie*, 19:121-128.
Zollinger, S. (2008). Kartenkritik an OpenStreetMap - Eine systematische Beurteilung der durch Mapnik visualisierten freien Weltkarte mit Fokus auf die Schweiz. Zürich: Institut für Kartografie, ETH Zürich, Bericht zur Semesterarbeit in der Lehrveranstaltung Geomatics.