Improving Automated Generalisation for On-Demand Web Mapping by Multiscale Databases

Alessandro Cecconi, Robert Weibel and Mathieu Barrault

Department of Geography, University of Zurich,Winterthurerstrasse 190, CH-8057 Zurich, Switzerland, {acecconi, weibel, barrault}@geo.unizh.ch

Abstract

This paper describes the generation of maps on-demand with the use of a multi-scale database. It is based on an analysis of the requirements of on-demand mapping and points out the different requests and limits of on-demand cartography. The central idea is to combine two commonly used approaches in cartography: On the one hand the use of a multi-scale database which includes two or more levels of details, on the other hand the use of cartographic generalisation methods. For selected object classes the paper discusses and evaluates design and implementation options for the multi-scale database and the generalisation of parts of the framework. The importance lies in the optimal combination of these two methods – which tasks must be solved by the MSDB and which through the generalisation process.
Keywords: multi-scale database (MSDB), on-the-fly generalisation, level of detail (LoD), web mapping, on-demand mapping

1 Introduction

In recent years, a new field of application has opened for cartography: web mapping (Peterson 1999). More and more websites offer maps of various kinds (topographic maps, thematic maps) and topics (route planning, city guides, among many others.) over the net. Many of these websites are equipped with zooming capabilities, but as these are not true generalisation capabilities, the quality of most web maps is poor.

1.1 Maps on the Internet

Maps created for the Internet are based on different conditions than paper maps. The map provider cannot control many technical constraints and requirements. With respect to visualisation, several parameters cannot be defined, such as hardware (display resolution), system software (operating system) or application software (browser). Since the bandwidth is typically still narrow, it is also very important to take into account the amount of transmitted data in distributed systems. Despite technical restrictions it's possible to create maps for the web, witness the numerous examples such as route planning, location finders and others (*www.mapquest.com, www.map24.com*). In functional terms most of those services, however, are not flexible as they have been designed for a well-defined purpose, such as providing locator maps for user-specified street addresses. Hence, the user has no opportunity to change anything or to define his/her purposes and requirements for the map graphics. To remedy these drawbacks a new strategy must be found which allows more flexibility for web mapping.

1.2 On-demand Mapping

On-demand mapping is concerned with the generation of maps based on user request and according to user requirements. Users are able to produce their own maps and customise the process of generation. A short explanation of on-demand mapping can be found on *http://www.ngdc.noaa.gov/seg/tools/gis/ondemand.shtml*. It deals with the dynamic creation of digital cartographic products, like topographic or thematic maps. It is only concerned with displaying data and does not create a target data set (van Oosterom and Schenkelaars 1995). In principle, however, the approach could be extended to include on-demand creation of reduced databases too. For the flexible creation of maps on-demand at arbitrary scales and for arbitrary themes cartographic generalisation is a necessity. However, cartographic generalisation is also known to be a time-consuming process, which runs counter to user expectations for Internet services, where the time factor is critical. Based on possible parameters defined by the user, three scenarios can be determined (table 1). These differ in the strength of the generalisation process and hence in the time required for map creation.

Table 1. Description of different scenarios for the on-the-fly and on-demand mapping

	Description	Map generation
Scenario 1	The user wants to have a first map for an overview of the requested information. The compute time should be very short and generalisation is only partially applied.	On-the-fly
Scenario 2	The advantages of the other two scenarios are combined here. The user needs a map with good cartographic quality within a limited time.	On-the-fly and on-demand
Scenario 3	In this scenario, the time component plays a minor role, and the user is willing to wait for the desired map. The generation can thus encompass the entire process of generalisation to obtain a high-quality map.	On-demand

The generalisation process critically determines how long it takes to create a map. Because automated generalisation is only partially solved, other solutions must be found 1) to substitute for missing automated generalisation operations and 2) to speed up cartographically appropriate yet computationally intensive generalisation algorithms. Multi-scale databases offer a possible solution (Buttenfield 1993, Timpf and Devogele 1997). Using a multi-scale database, the automated generalisation process can be divided into two steps. First, time consuming generalisation algorithms can be precomputed in advance (offline) and the results stored as levels of detail (LoDs) in the multi-scale database. Non-existing generalisation algorithms can be substituted (i.e., simulated) by means of interactive operations used to build the multi-scale database. Second, computationally efficient generalisation algorithms can be computed on-the-fly and used to refine the nearest LoD to the requested map scale. By means of this combined process on-demand map creation can be optimised and made more flexible at the same time.

Section 2 explains how such a MSDB can help to speed up the generalisation process and, conversely, how generalisation can help to render a more flexible MSDB. Section 3 briefly presents an example. Section 4 studies the generalisation process for selected object classes and shows the use of the different generalisation operators and algorithms. The paper will finish off with some conclusions and an outlook (section 5) of future work.

2 Combination of Map Generalisation and a MSDB for On-demand Map Creation

This paper explores an approach that combines components of a MSDB (Devogele et al. 1997) with components of on-the-fly generalisation (van Oosterom and Schenkelaars 1995). On the basis of user specifications (map scale, map purpose, symbology, information density) the appropriate object classes are selected from the database LoD for the desired map scale and further refined at run-time by on-the-fly generalisation algorithms. The overall design of this combined approach was described in Cecconi and Weibel (2001). To implement this architecture the main issues to be resolved are:

- to study the generalization process in conjunction with multi-scale databases;
- to develop a schema and structure for the MSDB which includes hierarchical linking of corresponding objects between different levels.

This paper focuses on the discussion of the former issue, trying to develop cartographically sound generalisation processes that exploit the combined strength of existing generalisation algorithms and MSDB. For brevity, we refrain from the discussion of the second issue. A first schema based on a relational approach has been designed for the road network (Cecconi 2001). More work on schemas for other object classes will follow after a careful study of the generalisation process (parts of which are presented in the paper), which allows us to better define the structure of the MSDB. Jones et al. (2000), as a further example of a combined approach, devote more attention to MSDB design, spending less time on building generalisation processes. Glover and Mackaness (1999) experimented with on-the-fly generalisation from a single scale database.

2.1 Use of a Multi-Scale Database for Cartographic Generalisation

As explained in the previous section the MSDB is used as a base element for the generation of maps. It is defined as a composition of different data sets, in which corresponding object elements are linked (Kilpeläinen 1997).

The MSDB includes a minimum of two data sets of geographic objects represented at different levels of detail (LoDs), or scales. In our case study, the first level is equivalent to a 1:25’000 topographic database. The second level is equivalent to 1:200’000. Fig. 1 (left) shows this set-up. The corresponding objects of the two levels are linked together (shown by dashed lines). The definitions of the links (Fig. 1, right) are very important and needed for on-the-fly generalisation. Every object in one scale has to know its counterpart in the larger or smaller scale. This information can be used to simplify the generalisation process. Given the links between the different levels of scale, generalisation can be understood as an ‘interpolation’ (or morphing) process between two different geometries.

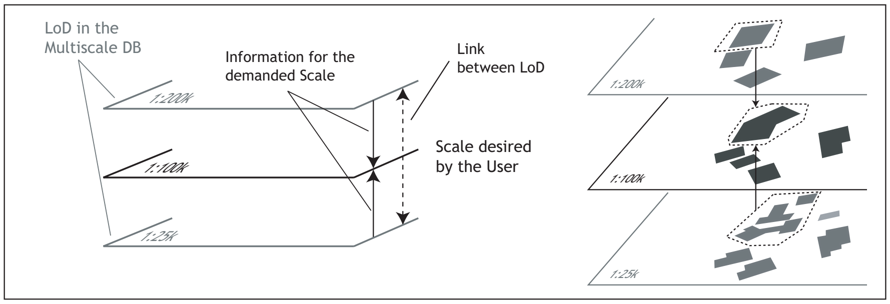

Fig. 1. Which data set should be selected for the desired scale (left)? How are the objects linked between the different data sets (right)?

For creating a map at the scale desired by the user (1:100’000 in our case) the appropriate LoD has to be determined. To this end, so-called limits of applicability are defined. These pre-defined limits can vary from object class to object class and depend on the given levels of detail. Fig. 2 shows the selection of the corresponding data set for the desired map scale based on these limits.

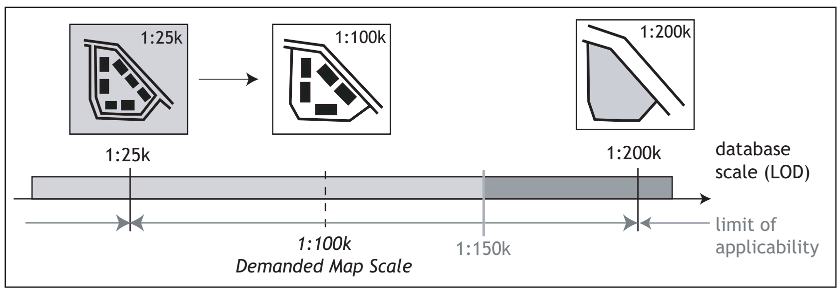

Fig. 2. Definition of the limits of applicability for the desired scale

Based on the assumption that the depiction of individual buildings is only applicable at scales of 1:150,000 or larger, the data set at1:25’000 was set to 150’000. Hence, in our example, the appropriate LoD is 1:25’000. For scales smaller than 150’000 the 1:200'000 LoD will be taken and built-up areas rather than individual buildings are shown.

2.2 Application of Generalisation Operators in different Scale Bands

The manual generalisation process is of an intrinsically holistic nature. For automated generalisation it is – at least for the time being – almost impossible to develop such a holistic solution. Hence, the generalisation process must be subdivided into a set of generalisation operators. McMaster and Shea (1992) have defined several generalisation operators, which – in an extended version – are building the basis of this research. It should be noted that other authors use different operators and may define them differently (see Weibel and Dutton 1999 for a review). The task of a generalisation operator is to solve a specific generalisation problem. Using such operators, it is possible to break down the generalisation process into smaller sub-processes; the combination of several operators can then be used to build an entire generalisation process or workflow.

The use of generalisation operators for the creation of maps is fundamentally dependent on the scale(s) and object classes involved (e.g., road network, hydrography). Depending on scale different operators (e.g. selection, simplification, typification and displacement) will be applied in different sequences. This knowledge of map generalisation can be used to design the contents of a multi-scale database. In particular aspects like 1) the levels of detail, 2) the limits of applicability for the LoDs of the MSDB, and 3) the generalisation operators used to transform between the LoDs can be addressed. Obviously, these decisions must be made separately for each object class and depend largely on the symbol specifications used (e.g. whether built-up areas are shown by individual buildings or simply as a tinted area). So, how can the above elements be defined in a meaningful way? Surely, we will seek to minimise the number of LoDs, and extend the limits of applicability as far as possible. Each additional LoD implies a significant additional cost, not only during database creation but even more importantly during database updates (which need to be consistently propagated across LoDs). In addition, it is rarely possible that data for the different LoDs can be gathered from existing sources, such as a topographic map series. We propose to analyse the ‘generalisation complexity’ as well as the application scope of the generalisation operators over the desired range of scales (Fig. 3). By ‘generalisation complexity’ we mean the combined ‘cost’ involved in generalisation; low values indicate simple operators can be used (e.g., selection and simplification) while high values hint at major modifications requiring complex contextual operators such as typification or displacement. The complexity values and generalisation operators can be established by analysis of maps and literature. While this is merely a qualitative method, it is nevertheless systematic and allows us to define the limits of applicability (i.e. the points where the ‘regime’ of generalisation operators changes) and hence the appropriate minimal number of LoDs. Note that this procedure bears some resemblance with the *points de généralisation* observed by Ratajski (1967).

Fig. 3 shows the behaviour of generalisation complexity and the relevant generalisation operators for a range of scales and for several selected object classes. For lack of space, we will only point out some essential features. The complexity curve for the “road network” class suggests three scale bands. In the first scale band (large scales down to 1:50’000) the selection and simplification operators are sufficient because there is usually enough room for displaying all road objects without much modification. In the intermediate scale band (1:50’00 to 1:300’000) important shape changes become noticeable. Because there is still enough space most road objects can be displayed, but must be transformed due to increased symbol width. This will be achieved by the operators typification (transformation of an initial set of objects into a new set, maintaining the typical arrangement) and displacement. Hence, the generalisation complexity increases.

In the third scale band the available space is so small that only the most important road objects appear; as the number of objects is low, typification and displacement are becoming less important and generalisation complexity decreases again.

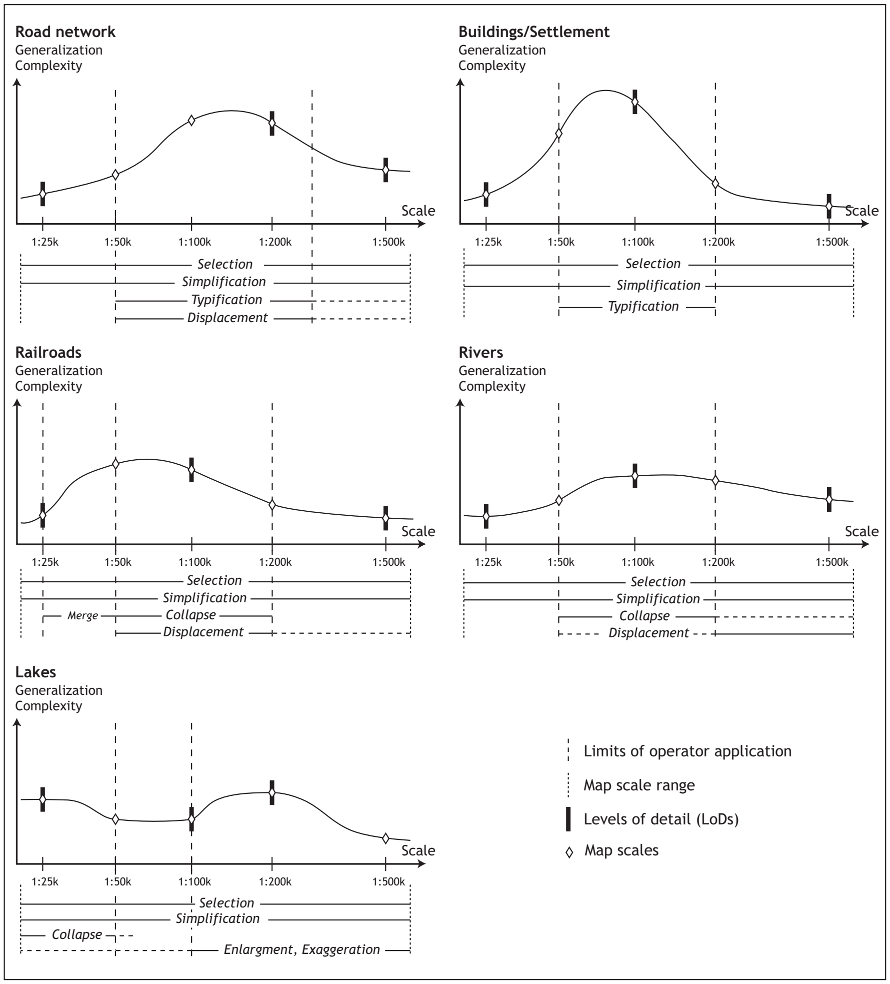

Fig. **3(a-e)** The 'generalisation complexity' and the application scope of generalisation operators for different object classes

The three scale bands can be represented by three LoDs, 1:25'000, 1:200'000 and 1:500'000. The object class "buildings/settlement" again shows three scale bands. The main break point (between scale bands 2 and 3) is the transition from individual buildings to tinted polygons for the display of built-up areas. On Swiss topographic maps (which are used in our research) this occurs around 1:200'000. A further break point can be found around 1:50'000. Up to 1:50'000 , the main operators include elimination of small buildings (i.e. selection of large buildings) and simplification of building outlines. Typification of groups of buildings and object displacements only occurs later, and significantly increases the generalisation complexity.

The object class “railroads” shows a different behaviour with a narrow band at large scales. The large complexity of in the first scale band is due to merge and collapse operations in station zones and switchyards, while overland tracks do not pose a problem due to large curve radii. It is only later, in the intermediate scale band, that overland tracks need to be generalised by various operators. Finally, at small scales beyond 1:200’000 only a small number of objects remain, decreasing the generalisation complexity.

For the class “rivers” most objects can be drawn as polygons up to 1:50’000. At medium scales objects need to be collapsed to lines and must eventually be displaced if overlaps are detected. As the collapse operation actually creates space the complexity in the intermediate scale band is lower than for other object classes.

Finally, the object class “lakes” exhibits a special behaviour with high complexity at large scales, low complexity at medium scales and large complexity at small scales. The large complexity in the first scale band is due to collapse operations of small inlets of rivers leading into the lakes.

Clearly, the advantage of using MSDB in conjunction with generalisation is that the automated generalisation process must not be carried out from a single detailed data set. Rather, it should start from a data set that is closer to the target scale, thereby simplifying the generalisation process enormously.

3 Example – Description of a More Detailed Scenario

We are presently implementing an experimental platform that uses the following scenario, taking place in an Internet (Client/Server) environment.

Technical specifications: The information will be transferred via network from a server to a local computer (client). Bandwidth is presently not of concern, as it is very hard to predict. The following technical restrictions must be taken into account (Kraak and Brown 2000): The display resolution of 72 dpi and the colour depth of 16 bits influences the symbolisation process; the display area of 1024x768 limits the real estate available for map display.

Application scenario: The main aim of the work is to generate a user-defined map from a multi-scale database online with the help of different generalisation techniques. The focus is on the generalisation components of the map creation process, meaning that the quality of the maps must be adapted to the user request. The user is assumed to request a topographic map, where the main focus will be on the road network, while the other topographic elements are used for orientation purposes. The target scale is assumed at 1:100’000, the MSDB that is being used consists of two data sets at 1:25’000 and 1:200’000, respectively.

Test area: The data set represents an area of 210km^2 with different topographic feature elements. The elements, which will be used for map display, are road network, buildings and hydrography.

4 Generalisation Processes for Selected Object Classes

This section intends to define possible sequences of the generalisation process for the selected object classes and determine which generalisation operators and algorithms will be used. A generalisation operator conceptually defines the type of (generalisation) transformation that is to be achieved; a generalisation algorithm is then used to implement the particular transformation. Note, however, that we are currently building a MSDB that relies on data from Swiss topographic maps. Hence, the characteristics of the Swiss national map series apply.

4.1 Highways

At both scales of our MSDB, 1:25'000 and 1:200'000, all highways are drawn. Modifications in the transition from the large to the small scale are relatively small, mainly affecting interchanges and ramps (entries, exits). Table 2 shows which operators are used for the generalisation process and which algorithms will therefore be implemented.

Table 2. Use of generalisation operators for the object class "highways" and the corresponding algorithms

	Operators	Algorithms
Selection	Because every object has to be selected, the selection operation is trivial to handle.	-
Weeding Simplification	While weeding is necessary, changes should be very small since highways are very important features on topographic maps.	Implementation of the Douglas-Peucker algorithm for weeding. Simplification of slip roads is done by using the information from the 1:200'000 data set.
Smoothing	Weeding is followed by smoothing.	Algorithm for smoothing by Lowe (1988).
Displacement	Due to the overriding importance and small number of highways, displacement will only occur in few places.	Displacement algorithm using elastic beams (Bader 2001).

The proposed generalisation process using a MSDB for the object class „highways" is shown in Fig. 4. Shaded boxes denote use of a data set, while white boxes indicate procedures. Starting with the data set of 1:25 '000, the data set 1:100'000 for the requested map will be created. Because all highway elements will be selected, the first step (selection) is trivial. After that the symbolisation (i.e. line width, colour type) must be defined either by the user (as part of his/her map request at the beginning) or by default values. Next, the weeding operator will reduce the number of points of the line, followed by the smoothing operator. The entries and exits will be simplified by using the knowledge of the data set at 1:200’000. This information can help to decide which segment must be shown. The rendering step is responsible for the representation of the elements on the screen. If unsolved problems remain after rendering and subsequent displacement, the process may return to symbolisation and restart with smaller symbol sizes.

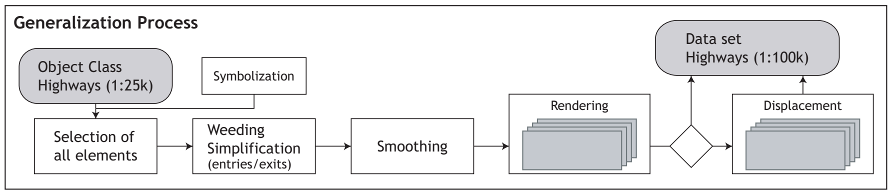

Fig. 4. The different steps of the generalisation process for the object class “highways”

The displacement step is optional. In the case of highways, it will only be needed in special situations, as highways are of key importance and hence will resist displacement.

4.2 Main and Minor Roads

The object class “major and minor roads” is more interesting because the selection operator plays an important role. At a scale of 1:25’000 usually all main and minor roads are represented, because sufficient space is available. At smaller scales, only main roads and important minor roads are displayed. Special road elements, such as “roundabouts” or squares will be collapsed into road sections.

Table 3. The use of generalisation operators for the object class “main and minor roads” and their corresponding algorithms

	Operators	Algorithms
Selection	Road selection is important. The roads are selected by importance and road class (comparison with LoD 1:200k).	Selection by using information of 1:200'000.
Morphing	The data sets 1:25'000 and 1:200'000 form the ‘key frames’. The map scale 1:100'000 are generated from these data sets by a morphing algorithm.	Morphing algorithm by Sederberg and Greenwood 1992 (shape blending).
Weeding	If there is no possibility for morphing a road (the road segment is not conclude in both data sets) the weeding process must be done.	Douglas-Peucker algorithm.
Smoothing	Weeding is followed by smoothing.	Lowe (1988).

Displacement	Displacement plays an important part in this case. After the symbolisation process it could happen that road objects are located too close to each other or to other objects.	Displacement algorithm using elastic beams (Bader 2001).

Very complex elements such as hairpin bends must be simplified or replaced by information from the data set 1:200’000. Table 3 shows the selected generalisation operators for this object class and the corresponding algorithms. The step ”morphing” does not belong to the generalisation operators in the closer sense (McMaster and Shea 1992), because it needs two data sets to transform the selected features. Nevertheless, in our case – a MSDB is used – it can be utilised for the generalisation process.

Fig. 5 shows the proposed generalisation process for the object class ”major and minor roads. Road selection is carried out in a two-step sequence. The first selection step is based on the comparison with the smaller data set. Every road section that appears at both scales must appear. The second step decides on the basis of the road attribute data and which road section is to be shown (assuming that the data in the database has been semantically enriched by importance scores in advance).

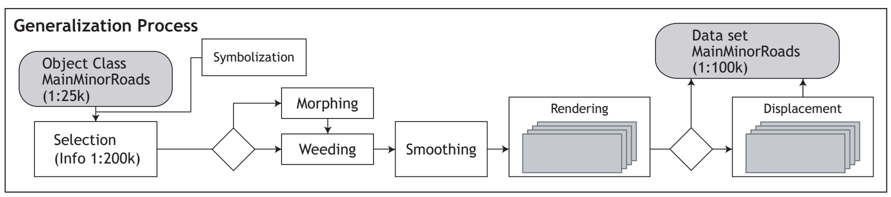

Fig. 5. Generalisation process for the object class “main and minor roads”

The symbolisation step is the same as for highways, though more classes need to be distinguished (major and minor roads are displayed differently). Next, it has to be decided whether a morphing transformation will take place or a weeding operator is directly applied. The difference between the two options depends on the data in the database. If the corresponding road section is contained in both data sets, a morphing process can be started; otherwise the weeding operator is used. The remainder of the process (rendering and displacement, as well as potential reiteration) is equivalent to the one explained for highways.

4.3 Buildings

Buildings are displayed in different shapes depending upon the scale. At a scale of 1:25’000 most objects are represented as polygon elements, at a scale of 1:200’000 as rectangle items, with the exception of large buildings (e.g. factory buildings). The smaller the map scale, the simpler the outlines of the buildings are. In densely settled areas (e.g. town centres) they may simply be represented as city blocks. At smaller scales the representation changes: the built-up area is now shown as tinted polygons instead of individual buildings. As we are using Swiss data in this work, however, this change of representation will not take place. Both scales use individual buildings.

Thus, the generalisation process is based on the assumptions of table 4. The main task of settlement generalisation for smaller scales using individual buildings is to maintain the overall pattern and arrangement of the large-scale representation. Hence, particularly in dense areas (like town centres), typical arrangements of buildings (clusters, alignments etc.) must be identified and defined in the MSDB. These specific groups of buildings must be linked across scales. This data enrichment process is typically carried out offline and prior to on-demand mapping process.

Table 4. Generalisation operators for the class “buildings” and the corresponding algorithms

	Operators	Algorithms
Selection	With the data set, 1:200'000 it can be decided which (groups of) objects are represented.	The 1:200'000 LoD provides the basis (data enrichment: building blocks and clusters in dense areas).
Typification	A reduced set of buildings that exhibits the typical arrangement is displayed at the small scale.	Interpolation-based typification (Hangouët 1996).
Rectification	Rectifies the geometry of objects, which are expected to have a rectangular shape.	Airault (1996), global algorithm for rectification.
Displacement	On overlap of single objects a displacement can be executed. In the case of groups of objects data from the 1:200’000 LoD can be used.	Displacement algorithm using elastic beams (Bader 2001).

The generalisation process for this object class is summarised in Fig. 6. As explained above using the previously defined cluster elements can alleviate the selection step. Also, the typification process can work with these clusters. The rectification step can be essential for the appearance of the buildings in the final map if the available data have not been previously rectified (e.g. if they were manually digitised). Displacement can be accomplished relatively easily since overlaps are easily detected.

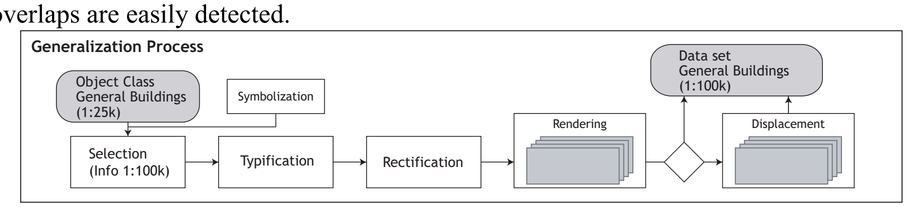

Fig. 6. Generalisation process for the object class “buildings”

Special buildings are buildings that are represented by a special symbol on large and small-scale maps. Examples include churches, castles, power plants etc.

The generalisation process can be executed easily in this case. In many cases all elements can be selected (compare with 1:200’000) and replaced by a previously defined symbol. In case of overlaps with other map objects the object under consideration can be eliminated or, alternatively, displaced if it is a very important building. More important is the definition of the symbol that represents the corresponding object. Typically, this information is stored in the MSDB. Due to the special symbols used, rendering also becomes more demanding.

4.4 Rivers

River items are represented at smaller map scales – in contrast to larger map scales – as linear elements. The change from surface objects to linear objects frees up map space (cf. section 2.2).

The selection can be simplified by the definition of stream orders for river sections (e.g. the Horton order). Table 5 shows the operators and algorithms used. Offline data enrichment can save precious time for on-the-fly generalisation. Two things can be done during the MSDB creation.

First, collapsing polygons to linear objects can be precomputed and added to the data in the MSDB as an alternative representation (i.e. a river object may both have a polygon and line representation). Second, stream ordering (in particular the so-called Horton order; Thomson and Brooks 2000) for the river network can be precomputed and stored as attributes for each river section. The Horton order can directly be used for river selection by network pruning.

Table 5. The use of generalisation operators for the object class “rivers” and the corresponding algorithms

	Operators	Algorithms
Collapse	The collapse process will be carried out offline and stored in the database.	Collapsing using a skeleton algorithm (Bader 1997). Takes place offline.
Selection	The rivers are selected on-the-fly depending upon the defined ordering scheme. LoD 1:200'000 can also be used to decide which elements should be displayed.	Selection by a predefined ordering scheme of river sections (Thomson and Brooks 2000).
Morphing	The LoDs 1:25'000 and 1:200'000 form the basis. Scale 1:100'000 will be generated from these by morphing.	Morphing algorithm by Sederberg and Greenwood 1992 (shape blending).
Weeding/ Smoothing	Weeding is followed by smoothing.	Douglas-Peucker; Lowe (1988).
Displacement	Due to polygon-to-line collapse, additional space is available, thus decreasing the need for displacement. On overlap, the same algorithms can be	Displacement using elastic beams (Bader 2001).

	used as for roads.

The selection can be simplified by the definition of stream orders for river sections (e.g. the Horton order). Table 6 shows the operators and algorithms used.

After having selected the objects that are retained the morphing or the weeding step will follow (Fig. 7). Morphing will be done if both data sets contain the corresponding object. Otherwise, simple weeding can be applied. Displacement will not be used often since the collapsing process can generate more space for the object.

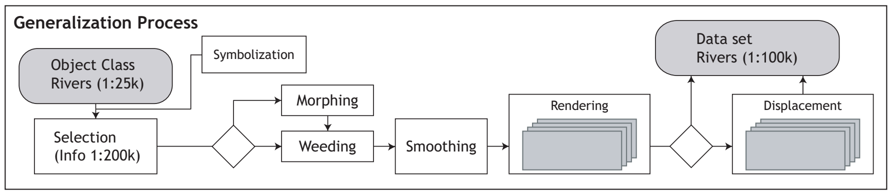

Fig. 7. Generalisation process for the object class "rivers"

4.5 Lakes

Lakes will be retained as long as possible through the scales. In most cases, they are represented as a polygon; if they are long and narrow they are collapsed to linear elements and treated as river object (stored as an alternative representation in the MSDB).

The lake surface generally determines which items will be displayed. As the map scale gets smaller, the outlines of lakes are increasingly simplified in each case. As lakes often exhibit elongated shapes, the minimum width of narrow passages and river inlets must be observed. Table 6 shows the generalisation operators and algorithms used for this object class.

Table 6. The use of generalisation operators for the object class "lakes" and their corresponding algorithms

	Operators	Algorithms
Selection	The lake surface (compared to 1:200'000) determines whether a lake will be retained.	Selection of the lakes in comparison with the data set 1:200'000.
Morphing	The LoDs 1:25'000 and 1:200'000 form the basis. Scale 1:100'000 will be generated from these by morphing.	Morphing algorithm by Sederberg and Greenwood 1992 (shape blending).
Weeding	Data reduction requires co-ordinate weeding.	Douglas-Peucker algorithm.
Smoothing	Weeding is followed by smoothing.	Lowe (1988).

Because lake objects are very important for orientation it makes sense to select as many as possible. If the surface of a lake falls below the minimum size limit, it is eliminated. Objects that appear in both data sets, however, should be shown (including those that are too small, in which case they will be enlarged).

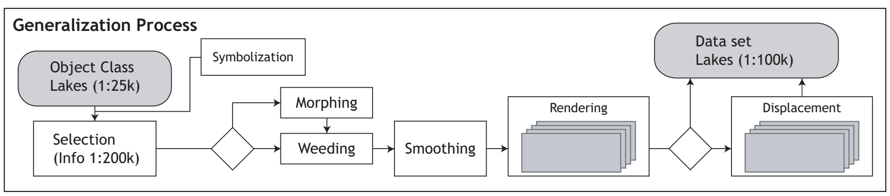

Fig. 8. Generalisation process for the object class "lakes"

Again, it must be decided if morphing or simple weeding will be used (Fig. 8). Since so many lakes will be retained at the small scale, it is important that the outline of the selected lakes is sufficiently simplified.

5 Conclusions and Outlook

This paper has proposed new ways of on-demand mapping in an Internet context. The creation of maps on-demand is a time-critical task. Yet, despite the fact that map generalisation is a computationally intensive task, capabilities for scale changing and generalisation should be added to on-demand mapping services on the web. Existing examples that offer zooming and scaling functions (e.g. www.mapquest.com) are however, usually defined for a single purpose and use multiple (independent) LoDs. As such, they do not allow flexible generation of maps at arbitrary scales and for arbitrary themes. Furthermore, multiple scale levels invariably mean more overhead during database creation and even more so during database updates (as updates need to be propagated across scales to maintain consistency). Fully automated map generalisation is not yet possible – some generalisation operations are still unresolved, and others are too time-consuming. Therefore, we propose to combine both multi-scale databases (MSDB) and map generalisation to exploit the strengths of both approaches in mutual support of each other. On the one hand, MSDB alleviates and speeds up the generalisation task and even substitutes for missing generalisation operators. On the other hand, generalisation renders MSDB more flexible and leads to better cartographic results.

Using our proposed approach, automated generalisation can be separated into two tasks and thus greatly simplified. The first task is carried out offline and consists of MSDB creation and data enrichment. The second task is computed online, adjusting the LoDs pre-specified in the MSDB to the required target scale, using on-the-fly generalisation and morphing. In this paper, we have presented the general approach and then concentrated primarily on the design of generalisation processes related to the second of the above tasks. To this end, we have discussed a method for the selection of appropriate LoDs for relevant object classes based on an analysis of the 'generalisation complexity'. Furthermore, we have presented in detail the structure and contents of the generalisation processes for several object classes. This information is now serving as a basis for the implementation of an experimental MSDB as well as an MSDB-based generalisation prototype. The experimental case study that we are pursuing focuses on on-demand topographic map creation (using Swiss maps as an example). Besides the more cartography-oriented aspects of the work reported here, an initial relational database schema design has already been accomplished (Cecconi 2001) and a rudimentary version of the test database has been built. However, we intend to extend the schema using object-oriented techniques and implement it based on a commercial system (Lamps2). Following that, the database for the test region in the Canton of Fribourg (Switzerland) will need to be populated, in particular by defining the links between corresponding objects of different scales. Next, the algorithms discussed in section 4 need to be adapted for our purposes and integrated into the MSDB context. The ultimate goal is to expand the specifications for the specific test scenario and build a generic module for on-demand mapping.

Acknowledgments

This research is part of the project "GENDEM: Map Generalisation for Thematic and On-demand Mapping in GIS" supported by the Swiss National Science Foundation under contract 20-59249.99.

References

Airault S (1996) De la base données à la carte: une approche globale pour l'équarrissage de bâtiment. Revue internationale de géomatique 6(2-3): 203-217

Bader M (2001) Energy Minimizing Methods for Feature Displacement in Map Generalisation. Ph.D. thesis, Department of Geography, University of Zurich

Buttenfield B (1993) Research Initiative 3: Multiple Representations. Closing Report, National Center for Geographic Information and Analysis, Buffalo

Cecconi A, Weibel R (2001) Map Generalisation for On-demand Mapping. GIM International, Magazine for Geomatics 15(5):12-15

Cecconi A (2001) Schema for a Multiscale Database for Selected Object Classes [online]. Internal Report, University of Zurich, Available from: www.geo.unizh.ch/~acecconi/gendem/msdb

Devogele T, Trevisan J, Raynal L (1997) Building a Multi-Scale Database with Scale-Transition Relationships. In: Kraak MJ, Molenaar M (eds) Advances in GIS Research II. Taylor & Francis, London pp, 559-570

Glover E, Mackaness WA (1999) Dynamic Generalisation from Single Detailed Database to Support Web Based Interaction. In: Proceedings of the 19th International Cartographic Conference. Ottawa, pp 1175-1184

Hangouët, JF (1996) Automated Generalisation Fed on latent Geographical Phenomena. In: Proceedings of InterCarto 2. Irkutsk, pp 125-128

Jones C, Abdelmoty AI, Lonergan ME, van der Poorten P, Zhou S (2000) Multi-Scale Spatial Database Design for Online Generalisation. In: Proceedings of the 9th Spatial Data Handling Conference. Beijing, pp 7b.34-7b.44

Kilpeläinen, T (1997) Multiple Representation and Generalisation of Geo-Databases for Topographic Maps. Ph.D. thesis, Publications of the Finnish Geodetic Institute, Helsinki

Kraak MJ, Brown A (2000) Web Cartography: Developments and Prospects. Taylor & Francis, New York

Lowe DG (1988) Organization of Smooth Image Curves at Multiple Scales. In: Proceedings of the 2nd International Conference on Computer Vision. Tampa pp 558-567

McMaster RB, Shea KS (1992) Generalisation in Digital Cartography. Association of American Geographers, Washington

Peterson MP (1999) Trends in Internet Map Use: A Second Look. In: Proceedings of the 19th International Cartographic Conference. Ottawa, pp 571-580

Ratajski L (1967) Phénomènes des points de generalisation. In: International Yearbook of Cartography, Vol. VII. Gütersloh Bertelsmann, Bonn-Bad Godesberg pp, 143-152

Sederberg TW, Greenwood E (1992) A Physically Based Approached to 2-D Shape Blending. Computer Graphics 26(2): 25-34

Timpf S, Devogele T (1997) New Tools for Multiple Representations. In: Proceedings of the 18th International Cartographic Conference. Stockholm, pp 1381-1386

Thomson RC, Brooks R (2000) Efficient Generalisation and Abstraction of Network Data Using Perceptual Grouping. In: Proceedings of the 5th International Conference on GeoComputation. Manchester (CD-Rom)

van Oosterom P, Schenkelaars V (1995) The Development of an Interactive Multi-Scale GIS. International Journal of Geographic Information Systems 9(5): 489-507

Weibel R, Dutton G (1999) Generalizing Spatial Data and Dealing with Multiple Representations. In: Longley P, Goodchild MF, Maguire DJ, Rhind DW (eds) Geographical Information Systems: Principles, Techniques, Management and Applications. John Wiley, Chichester, pp 125-155