# SPATIAL STRUCTURES AS GENERALISATION SUPPORT SERVICES

Moritz Neun, Dirk Burghardt and Robert Weibel

Department of Geography, University of Zurich – {neun,burg,weibel}@geo.unizh.ch

**Joint ISPRS Workshop on Multiple Representation and Interoperability of Spatial Data**

**KEY WORDS:** map generalisation, web services, generalisation web services, graphs, triangulation

**ABSTRACT:**

While standards for automated access and presentation of cartographic data over the internet are defined, services for automated generalisation and the transfer and storage of the data involved are not yet standardised. Web service technologies can be used to establish an interoperable framework between different generalisation systems. The service architectures can be distinguished into middleware services, which are delivering original and/or pre-generalised data from a database through a WMS or WFS, and research platforms which provide only access to the generalisation operators and can be used with own data. Three categories of generalisation web services are identified. Support services are enriching the raw input data with additional information or expressing structural and spatial relationships. Operator services deliver the functionalities of individual generalisation operators and take enriched data as input. And finally, process services use the two other service types in order to control the generalisation process. The modelling of structural and spatial relationships is critical for the understanding of the role of cartographic features and thus for the automated generalisation. On one hand the relationships can be hierarchical structures between map features or feature classes. On the other hand there can be relations between the features themselves which can be expressed by matrices or graphs. This paper aims to discuss and define which types of support services can be established for enriching the raw input data and how the complex output of such services can be represented and used in a web services environment.

## 1. INTRODUCTION

The map generalisation research community has recently started to develop an interest into Generalisation Services, driven by the desire to develop a common open research platform that would allow testing and sharing of generalisation algorithms (Edwardes et al., 2003). Such an open generalisation system supports co-operation by sharing of techniques within the cartographic research community, for example new algorithms, data structures or measures. It also supports external collaboration through the application of generalisation in other GIS research areas, for example in geo-visualisation and location-based services.

### 1.1 Interoperability through Web Services

In the “Issues of Interoperability in Map Generalisation” (ICA, 2004) a strong interest for developing a standardised service based architecture has been expressed. This special type of service, the “Generalisation Service”, has so far not been standardised by the Open Geospatial Consortium (OGC, 2002). The starting point for such a Generalisation Service should be some suitable small services (ICA, 2004) without dealing with the harmonisation of data types and structures (Lehto and Sarjakoski, 2004; Sester et al., 2004; Illert and Afflerbach, 2004). This has already been demonstrated for interoperable services for the visualisation of spatial information (Fitzke et al., 2004). There are several advantages of using Generalisation Services in a collaborative and distributed research environment as well as for on-demand map production. First of all, the platform independence makes the development independent from the operating system and the hardware used, which also allows the use of this technology in a mobile context. Secondly, the service can be integrated in any software platform, such as web browsers, GIS, or map production software. Furthermore, the possibility exists to write specific algorithms for special computer architectures such as clusters, grids, or other parallel processing systems and to offer such services to the subscribers. Last, services can be accessed over the internet or locally.

### 1.2 Generalisation Service Architectures

From the above, we conclude that generally two generalisation service architectures – middleware generalisation services and generalisation research platforms – seem to have a promising range of application.

A *middleware generalisation service* can deliver original or pre-generalised data from a web server such as a Web Map Server (WMS) or Web Feature Server (WFS), as shown in Figure 1. A similar approach is described by Sarjakoski et al. (2005) using a modified WFS. A possible use is an adaptive zoom for Web Map Services as it is currently the domain of multi-resolution databases (MRDB).

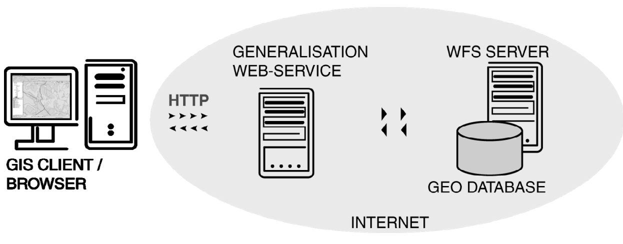



Figure 1: Middleware generalisation service

This service would require a fully automated, real-time execution of the generalisation process. Another possible use are generalisation support services (see section 1.4) which pre-process cartographic data e.g. from a WFS so that they can then be used by more complex generalisation algorithms.

The other service type is the *generalisation research platform*, an interactive Generalisation Service for GIS research and for GIS users in organisations such as national mapping agencies (NMA) and Universities. In this case the Generalisation Service would provide its functionality and its calculation power to the service subscribers. It also fulfils the requirements of a common research platform (Edwardes et al., 2005), where the researchers want to have access to a common generalisation framework. This WebGen platform offers the ability to provide specialised or novel algorithms to the research community without forcing the participating researchers to adapt their systems to the specific needs (Neun and Burghardt, 2005).

In an open research model every researcher can present his/her own generalisation service (Fig. 2). Through the internet and the use of platform independent technologies such services can reside on servers all over the world. For discovering these services some “Yellow Pages” are needed, indicating which services are available, where they are located and what algorithms they offer (Burghardt et al., 2005). This is the “Registry” for generalisation services. The registry offers a single access-point where all further information is to be found. Whilst services can change or move they are always to be found again through the registry. This model for sharing and discovering generalisation services can be summarised by the “publish-find-bind” paradigm (UDDI, 2005) where the service is published, for instance, by the author in the registry and can then be easily found and used by others.

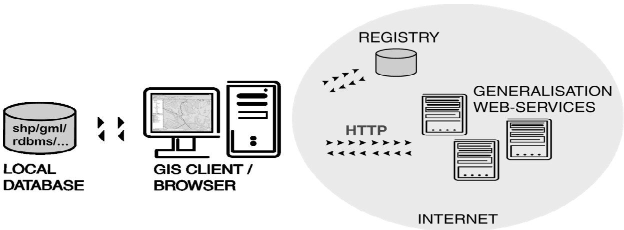



Figure 2: Generalisation research platform

First Generalisation Services have already been implemented (Neun and Burghardt, 2005). These are rather simple services (e.g. Douglas-Peucker line simplification, building simplification) where spatial context is not considered. The objective was to show the feasibility of the service-based approach and to describe the minimum set of components needed to run the Generalisation Services over the internet. The service algorithms are written in Java for the WebGen framework (Neun and Burghardt, 2005). This framework uses a web-server with Java servlets, SOAP (2000) for the communication and JTS Topology Suite (2006) for the geometry representation. JTS conforms to the Simple Features Specification for SQL, published by the Open GIS Consortium (OGC, 1999).

The client for accessing the services is currently available for the JUMP Unified Mapping Platform (2006). The client plug-in offers the configuration and selection of the desired service. The data to be processed by the service together with the parameters for the algorithm are encoded and sent to the service. The result, sent back from the service, is then decoded and presented in the client. A client with similar functionalities could also be developed for many of the commercial desktop GIS having an API for adding functionalities.

### 1.3 Categories of Generalisation Services

The overall generalisation process involves both, rather simple generalisation operations which are applied only to individual map objects (so-called independent generalisation), as well as highly context-dependent operations which require control over the generalisation workflow (so-called contextual generalisation). Hence, Burghardt et al. (2005) argue that three categories of generalisation services must be offered in order to enable comprehensive web-based generalisation. These three categories are shown in context in Figure 3 and briefly discussed below.

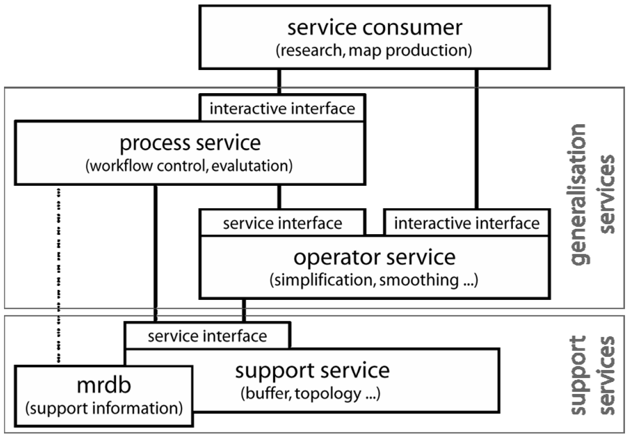



Figure 3: Categories of generalisation services

**1.3.1 Generalisation Support Services** are, for example, services for buffering or for creating a topological data structure, a skeleton or a constrained Delaunay triangulation. Results of such a service can be seen as additional (enriching) cartographic information in support of the automated generalisation process. Support services take raw data with a simple structure as input and deliver either a simple structure but with additional enriching information or a more complex data structure with object relations as output. Thus a main goal of such services is to make structural information explicit, representing common structural properties such as alignments, neighbourhood or proximity relations, which can be usefully exploited by generalisation operations (Neun et al., 2004). Sometimes the establishment of such supporting information is very expensive in terms of time and memory so that optionally the persistent storage in a special multi-resolution database (MRDB) can be useful e.g. for real-time generalisation.

**1.3.2 Generalisation Operator Services** deliver the functionality of standalone generalisation operators such as the ones defined by McMaster and Shea (1992). Examples are services for simplification, smoothing, aggregation, amalgamation, merging, collapse, refinement, exaggeration, enhancement and displacement. These generalisation operator services can be further subdivided for point, line and area objects and specialised depending on object classes. It is obvious that rivers, political boundaries, or railway tracks have to be generalised in a different way, despite the fact that all of them are represented as line objects. The generalisation operators of this second service category are offered in an interactive mode, with the user selecting appropriate generalisation operators and algorithms as well as setting the control parameters of the algorithms.

**1.3.3 Generalisation Process Services** use services from the lower categories for the control and orchestration of generalisation operators. Examples are services for automated orchestration, services for the evaluation of generalisation results, as well as meso agents (Ruas 2000) as described for the advanced Generalisation Service category. Automated control of the generalisation process presently receives ample attention as a research topic. Besides agent-based modelling, combinatorial and continuous optimisation approaches are also proposed in the literature (Harrie and Weibel, 2006). Simulated annealing (Ware et al., 2003), as a combinatorial optimisation approach allows the selection of generalisation operations controlled by assigning costs to each operation. Continuous optimisation approaches include the finite element method (Højholt, 2000), snakes or elastic beams (Burghardt and Meier, 1997; Bader, 2001; Galanda and Weibel, 2003) and least-squares adjustment (Harrie, 1999; Sester, 2000).

## 2. TAXONOMY OF GENERALISATION SUPPORT SERVICES

The execution of generalisation operators or algorithms depends on the input they receive. Of importance are elements such as algorithm parameters, the character of the map features to generalise, and also mutual influences between map features, such as roads exerting a push on nearby houses in the map feature displacement.

In generalisation, implicit knowledge about the spatial context in the algorithms is a very important factor (Mustière and Moulin, 2002). So, relations between the map features are explicitly established or basic geometrical operators are executed. Examples range from the creation of a simple buffer to a topological data structure, a skeleton or a constrained Delaunay triangulation.

Often, when a researcher develops a new generalisation algorithm he/she will have to create all these supporting functionalities and structures from scratch or develop converters to use existing ones. In contrast, support services try to offer a framework of common supporting functionalities and data structures which can then be (re-)used by more advanced generalisation operators. Offering these support services as web services (Neun et al., 2005) makes the use of the supporting framework easier and platform independent. Therefore the output of the support services must be storable and transferable across networks in a standardised way. Support services can be used for both, data pre-processing and real-time generalisation. The limiting factor is the calculation time of the specific support service not the interface.

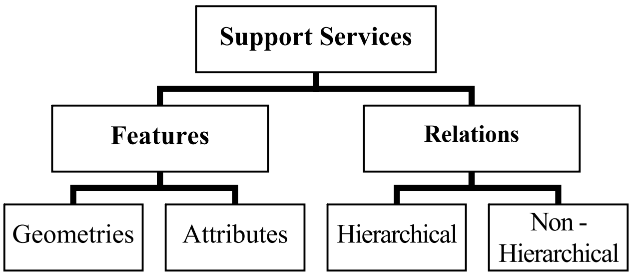



Figure 4: Support service types

Support services can be distinguished by the type of the supporting data they offer, and thus by the output they deliver to the service consumer (Fig. 4). There are two main types. First, there are the support services which deliver features with new or modified geometries or attributes which help the generalisation operator (see section 2.1 for details and examples). Second, there is the large family of graphs (section 2.2). They range from (directed or undirected) networks, such as transport graphs and triangulations, to hierarchies which can be expressed by trees. All graphs and trees can also be represented as a data structure by matrices or arrays.

### 2.1 Feature Support Services

The most obvious output of a support service are simple features (OGC, 1999) with supporting attributes or geometries. Functionalities to read and write these supporting data structures are included in most GIS software. These services are just enriching features with additional attributes or are creating new support geometries.

**2.1.1 Creating Geometries:** Output of such a support service is just geometries. They can easily be expressed by simple features. Thus, no extra data format is needed to get results.

An example of a simple support service is the creation of the buffer polygon from an input geometry which could also be used for a selection service (Fig. 5). Taking, for instance, a road and a set of houses as input a selection service returns the houses contained in a certain buffer around the road. The output of both services are just simple features.

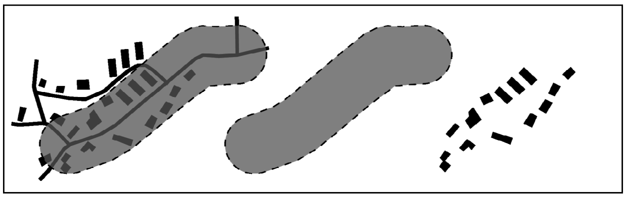


Figure 5: Buffering / feature selection

Similarly the edges of a road network could be used for partitioning and thus for the selection of features. This functionality could be available through a topology support service (see 2.2.2)

A further example of supporting geometries includes the computation of alignment lines, chaining together a group of map features such as buildings (Christophe and Ruas, 2002). The creation of inflection points or localisation local extremes for line generalisation (Plazanet, 1995) serves as a final example of supporting geometries. It delivers a series of critical points which can then be used, among others, for line segmentation (partitioning) or the creation of trend lines, approximating a line (Fig. 6).

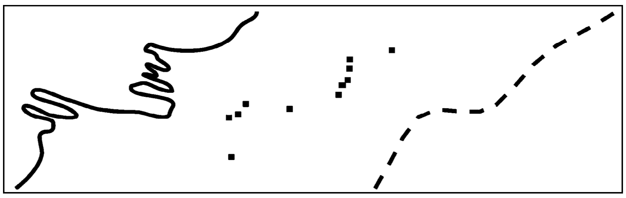


Figure 6: Inflection point / trend-line generation

**2.1.2 Generating Attributes:** These services take map features as input and return the features with changed or new attributes. In essence, most of these services are performing an analysis of the shape and structure of map features, also termed cartometric analysis (McMaster and Shea, 1992). An example of such a function is the calculation of the sinuosity or density of a line (Plazanet, 1995) which is stored as an attribute of the line feature. Plenty of other measures can be calculated, such as area size, shortest distances to next neighbours, and many more. Often, these measures can be used in a comparative way to establish priority orderings among map features (e.g. small polygons may be defined to be insignificant and therefore omitted).

### 2.2 Relation Support Services

The modelling of spatial relationships is critical for the understanding of the role of cartographic features and thus for automated generalisation (Regnauld, 2005). Additionally, structural and semantical relationships are essential ingredients for many complex generalisation operators. Relationships between cartographic features can be modelled as graphs (Fig. 7). Examples are *topological data structures (e.g. polygonal maps), transport* and *neighbourhood graphs, triangulations* or *surface networks*. As a more specific case, hierarchies can be expressed by trees. Hierarchies occur for semantical as well as for spatial relationships.

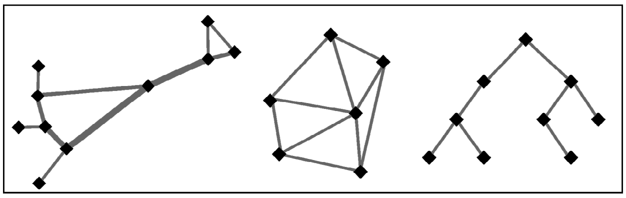


Figure 7: Examples of graphs: Weighted graph, triangulation and tree

Thus, the following support services for expressing relations have as common output a data structure which can be expressed by a graph. Graphs can be further differentiated into (general) networks and trees, with trees being a special form of a graph with no cycles, rooted in a single node.

**2.2.1 Hierarchical Relations** can exist between cartographic features. Then, the hierarchy creating criterion is any property of the feature such as the position or an attribute such as a class. However, hierarchies can also consist only of attribute values or counts, such as feature classes or the frequency of a certain feature type. Hierarchical relations can usually be expressed by trees.

The following hierarchical data structures are supporting generalisation algorithms and are therefore potentially useful as generalisation support services:

- Complex features (feature groups)
- Hierarchical similarity trees or dendrograms
- Links between levels of detail (LoDs) of MRDBs
- Hierarchical network ordering
- Reactive data structures
- Hierarchical surface data structures
- Partitioning and distributed processing

*Complex features:* Map features often form meaningful groups, that is, complex map features consisting of simple features. Examples include a cluster of buildings that form a small settlement (however, not being represented explicitly), a group of buildings and surrounding streets forming a city block, or several fields, ponds, trails, etc. forming a park. Complex features are building groups, thus they are the simplest and also most general case of hierarchical (partonomic) relations. Such complex features can either be user-defined or established automatically by cartographic pattern recognition (e.g., clustering procedures).

*Similarity trees or dendrograms:* For the aggregation of geometries or the translation of features from one feature schema to another, often a reclassification of the feature categories is needed. A reclassification needs some sort of rule base how to assign new categories to the input features. This rule base can be in many cases a strict hierarchy which assigns a new category to every input category (e.g. 'deciduous forest' and 'coniferous forest' are both reclassified to 'forest'). This hierarchy can be defined by the user of the system or generated automatically, for instance by a statistical evaluation of the input categories to establish their relevancy. A reclassification service would request a **similarity tree** from the support service and then classify the map features accordingly. Thus a similarity tree expresses the semantic similarity or adjacency of the feature’s classes which can then be used for reclassifications or aggregations (van Smaalen, 2003). A special case is encountered if multiple output categories are possible for a given input category. However, in such a case these dependencies are no longer hierarchical and a weighted decision graph (e.g. represented as a directed acyclic graph, DAG) can be used instead (see 2.2.2).

*MRDB links:* For the generation of multi-resolution databases (MRDB) out of different datasets the **matching** of the features on the different levels of detail (LoD) is indispensable. The **links** between the features on the higher scale with the matched features on the lower scale are mostly of nature 1:0, 1:1 and 1:n as shown by Timpf (1998). 1:1 and 1:n relations can be expressed by a simple tree. However, n:m relations as they exist, for instance, in typification operations (e.g. 5 buildings are typified by 3 buildings) are more complex to model and require a DAG (directed acyclic graph) as they can’t be modelled by a tree.

*Hierarchical network ordering:* A well known example for a hierarchy is the **Horton-Strahler** ordering of hydrology networks (Horton, 1945; Strahler, 1952). This is a very obvious and intuitive example for a tree as a river from the embouchure to the source links has a natural tree-like structure. In many cases this tree structure must not be explicitly generated again by the support service but the ordering can just be assigned as weights to the branches of the river network. Hierarchical networks of the main hydrological features, stream channels and ridges, can also be modelled as interlocking networks (with channels being the dual of ridges; Werner, 1988). Such structures were also used by Weibel (1992) to represent the 3-D skeleton of a terrain surface and hence generalise digital terrain models.

*Reactive data structures:* For the efficient storage and access of line simplifications or polygon aggregations in MRDBs tree structures provide a useful hierarchy for the contained features. Ballard (1981) introduced the **strip tree** which uses rectangular strips for partitioning a line. On the highest level of the strip tree the whole line is contained within one strip (i.e. co-axial minimum bounding rectangle, MBR). Proceeding down the binary strip-tree the strips get smaller and deliver a better approximation of the line. Another hierarchical structure for line approximation in a multi-resolution environment is the **BLG-tree** (van Oosterom, 1993). This reactive data structure uses the Douglas Peucker line simplification algorithm for splitting up the line.

*Hierarchical surface data structures:* Hierarchical structures for storing surfaces include, for example, the multi-resolution topographic surface database (Ware and Jones, 1992) which uses hierarchical surface triangulation as well as hierarchical TINs (HTINs) which use also a hierarchical triangulation for multi-resolution surface description (De Floriani and Puppo, 1995). Hierarchical surface data sturctures are mainly used in applications that require real-time rendering of surfaces, such as terrain surfaces in flight simulation.

*Partitioning and distributed processing:* Spatial partitioning of map data sets is often helpful to assist the generalisation process. For instance, settlement generalisation is facilitated by partioning the settlement into city blocks formed by the street network (Ruas 2000). Additonally, such partitions might also be usefully exploited to increase the speed of complex generalisation operations. Also, the distributed processing of a partitioned dataset on multiprocessor computer architectures or even on clusters could be possible. Unfortunately, appropriate methods for distributed processing do not exist.

**2.2.2 Non-Hierarchical Relations**: Common representations of non-hierarchical data structures are graph-based network structures. These networks can contain cycles, are possibly weighted, and/or directed. The graphs proposed here as support services describe the relationships between cartographic features in many ways.

The following networks may support generalisation algorithms and are therefore potentially useful as generalisation support services:

- Transport graphs
- Minimum spanning trees (MST)
- Neighbourhood graphs
- Topology graphs
- Triangulations and related structures
- Surface networks
- Decision graphs

*Transport graphs:* Roads or railway networks form in a way a natural graph. This **transport graph** can be used for example to generalise a road network connecting a set of cities by selecting the roads in the minimum spanning tree (Mackaness and Beard, 1993; Thomson and Richardson, 1995). This way the connectivity of the transport network is assured.

*Minimum spanning trees (MST):* A minimum spanning tree (MST) can also be used in automated road matching for multi-resolution databases. The MST is, for instance, used for the selection of candidate roads (Lüscher, 2005). Although being a tree the MST does not express a hierarchical relationship between features. For managing building relations in the building displacement process Bader et al. (2005) use a **ductile truss**. This elastic truss connects the building centroids, adding further edges to a minimum spanning tree, forming cycles until every node is connected to two neighbours.

*Neighbourhood graphs:* A very direct and geometric relationship is the neighbourhood of a feature. Anders (2004) used **neighbourhood graphs** for the interpretation of spatial data, data analysis and data enrichment of disjoint objects (e.g. the buildings of Fig. 8). As support service for generalisation a **nearest neighbourhood** graph or **a relative neighbourhood** graph can be used in many cases where one feature influences surrounding features due to its generalisation.

*Topology graphs:* For expressing adjacencies between map features **topological data structures** are used. A topological data structure is an extended planar graph and uses nodes, edges and faces to represent the topological relations of the features. Thereby the space is divided completely by the nodes and edges of the map features. The use of such a topology graph in an algorithm ensures data integrity, shared boundaries and connected networks. For example, in generalisation topological rules (embedding, planar enforcement, etc.) can be used to prevent holes which are created due to a geometry type change from a polygon to a line or point (Bobzien and Morgenstern, 2002). A data format for storing and exchanging topological data structures is available through GML 3 (OGC, 2004). Thus, support services for adding topology to a dataset are possible and do not need any new data format. However, full topological structuring is quite heavyweight and sometimes, such as for assuring connectivity, a much simpler graph could be used. For instance, Petzold et al. (2005) used a 'polygon connectivity graph' to relate polygonal road segments in order to generate the road skeleton.

*Triangulations and related structures:* The **Delaunay triangulation** of a point (vertex) set is a collection of triangles satisfying the property that no other vertexes are within each triangle's circumcircle (Fig. 8). The constrained and the conforming Delaunay triangulations are adaptations of the Delaunay triangulation which can be used to triangulate over polygonal objects by incorporating the polygons edges as predetermined or constrained triangle edges.

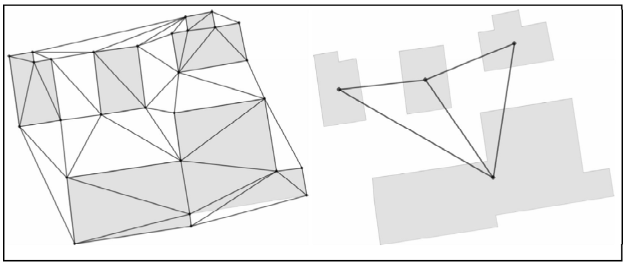


Figure 8: Delaunay triangulation (non-constrained) of polygons and of polygon centroids

Jones et al. (1995) use the constrained Delaunay triangulation in their simplical data structure (SDS) for implementing exaggeration, collapse, amalgamation, reduction and displacement algorithms. Ruas (1998) uses a Delaunay triangulation of building centroids to represent proximity relations for managing the building displacement process.

The **Voronoi diagram** is the geometric dual to the Delaunay triangulation. It is also known as Thiessen polygons and defines the border of the space which is closer to the contained object (e.g. a point) then to any other object. Thus, the result is a complete tessellation of the space between the objects. Chitahambaram and Beard (1991) use these properties for creating the skeleton of a polygon.

*Surface networks:* **Surface networks** (Woods and Rana, 2000) use a graph-theoretic approach originally suggested by Pfaltz

(1976). Surface topology is stored as a weighted graph consisting of vertices representing the surface-specific points (peaks, passes, and pits), and edges representing connecting ridges and channels. Both point features (peaks, pits, and passes) and line features (ridges and channels) are assigned a weight according to their degree of importance. Thus, relatively unimportant parts of the network can be removed and the remaining network readjusts automatically.

*Decision graphs:* Strictly partonomic relations can be expressed by hierarchical trees (see section 2.2.1). Partonomies with multiple associations, depending on an attribute or any other criteria, however, can not be represented in a tree because of the possibility of cycles in the structure. Therefore a **weighted decision graph** (or a weighted adjacency matrix) can express such relationships. Such a graph can also be represented as a directed acyclic graph (DAG) which has, if needed, weighted edges. An example is the reclassification step which precedes the aggregation or amalgamation of cartographic features. As shown in Fig. 9 an alley could, depending on its size, become part of a city block or part of the road network.

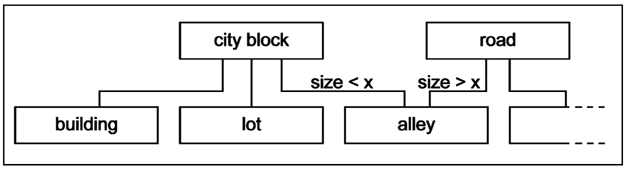



Figure 9: Partonomy with multiple associations

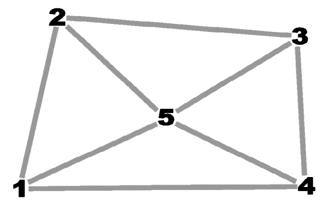
## 3. REPRESENTING GRAPHS

In the context of web services we distinguish the representation of a graph as a data structure in the computer's main memory and the representation as a persistent data format for the exchange of the graphs.

### 3.1 Data Structure

The in-core data structure has to facilitate the generation, modification and especially the querying of a graph in the computer's main memory. This data structure is depending on the platform and the programming environment. In general we can distinguish list or matrix based as well as object oriented representations.

For all types of graphs there are two common node based data structures for graph representation. In the **adjacency list** all nodes are listed in an ordered fashion and every node has a unique identifier (number). Corresponding to every node there is a list or array containing the identifiers of its neighbours, connected by incident edges. The **adjacency matrix** uses a two-dimensional boolean array of size #nodes x #nodes to represent node adjacencies. In many cases the adjacency matrix (see Fig. 10) is easier and faster to use. For graphs which have a high number of edges per node, not much space remains unused in the matrix. However, if there are not many edges per node (i.e. if the node degree is low), such as in triangulations with large numbers of nodes that are on average of degree 5 to 6, a lot of space is wasted in adjacency matrices. Thus usually matrices are not applicable in real application because they require too much data storage already for normal size datasets. In adjacency lists, on the other hand, the size of a list is dynamically dependent on the number of nodes and edges. For triangulations also a **face list** (actually, a triangle list) data structure is possible. In this case a list containing the nodes with unique identifiers and a list containing the faces of the graph (i.e. triangles) each with the three nodes of the triangle are built. This data structure is easier to query, e.g. for containment in a triangle.




| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 1 | 1 |
| 2 | 1 | 0 | 1 | 0 | 1 |
| 3 | 0 | 1 | 0 | 1 | 1 |
| 4 | 1 | 0 | 1 | 0 | 1 |
| 5 | 1 | 1 | 1 | 1 | 0 |

Figure 10: Simple graph (triangulation) and corresponding adjacency matrix

**Object-oriented (OO) concepts** allow a more flexible graph representation. In this case the processing efficiency is more important than the storage. Regnauld (2005) proposes a Java based object-oriented data structure to represent proximity graphs and triangulations. OO inheritance allows extending a basic graph representation with nodes and edges.

In a simple basic OO graph representation (see Fig. 11) a graph consists of a list of nodes and a list of edges. Each node contains only a list of the incident edges. The edges have two variables which point to the two end-nodes of the edge.

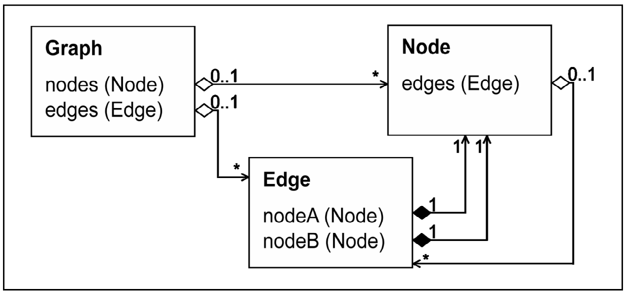



Figure 11: Basic object-oriented graph representation

For the use as generalisation support such a basic graph can be easily extended through class inheritance. In graphs which represent cartographic features, that is, where the nodes correspond to nodes or vertices of cartographic features, the nodes have coordinates and/or contain a link to the corresponding cartographic feature. For planar graphs like topology graphs or triangulations the basic graph model can be extended with faces or triangles. An example of such an extension is shown for Delaunay triangles by Regnauld (2005).

For creating such an OO Delaunay triangulation usually the algorithm to generate the triangulation receives a set of points as input. The algorithm builds the edges and triangles and stores the nodes, edges and triangles in Java objects. In the same environment the triangulation can now be queried and modified by other algorithms. After the completion of the program the triangulation will be lost. As an additional problem, such objects are not easily exchangeable with other programming environments or across networks. Therefore in the context of support services, where the different services may reside on distributed computers, a data format to persistently store and exchange the graph structures is needed.

The use of the same data format for storing the graph and representing it in the main memory is possible, but not necessary and often not even appropriate. It can be appropriate if both the in-core data structure and the transfer format are based on adjacency lists or adjacency matrices (represented by lists or arrays). For object-oriented data formats, however, it is more suitable to have two different data formats. That is, the in-core representation should be optimised for the querying and modification of the graph while the transfer data format is optimised for being efficiently saved, transferred over a network and parsed at the receiver.

### 3.2 Storing or Exchanging Trees and Graphs

Trees are directly implementable in the hierarchical XML format. Elements of an XML data structure can enclose other elements. Thus they are creating a nested structure which represents a strict hierarchy. The following example shows a building being part of a lot which is again part of a cityblock:

```
<cityblock>
   <lot>
      <building/>
   </lot>
</cityblock>
```

Such an XML representation can also be used in an object-oriented manner in the main memory of a computer. For doing so almost all XML parsers for object-oriented programming languages are offering a tree-like data structure for querying the content of an XML document.

Data formats or languages for storing and exchanging graphs do exist. There are very simple text files containing node and face (i.e. triangle) lists (Shewchuk, 1996) or the XML based *Graph eXchange Language* GXL (Holt et al., 2000). GXL is very powerful and tries to represent a maximum of different graph types. As XML is very flexible and has a hierarchical structure also own XML formats for representing graphs can easily be created.

All these data formats are based on a list representation such as an adjacency list for simple graphs or a triangle list with associated nodes and possibly their adjacent triangles. The advantage of these data formats is that they are compact, they don't contain much redundancy and they can easily be saved in a file or be sent over a network using standard protocols. However, using these data formats as in-core data structure for graph representation is not very practical and efficient. An object-oriented data structure (see above) is much easier to query, modify and extend.

The WebGen research platform described by Neun and Burghardt (2005) uses the XML based SOAP protocol (2000) for the data exchange between the different services and the client, making the use of an XML based graph data format most straightforward. As there exists yet no standard for representing any graph in GIS a data format is desirable which can easily be created and read in different programming environments or GIS platforms. Thus, parsers for the different generalisation support data structures are needed. In the case of an OO representation of the graph on the computer the data structure is converted to an XML structure sent over a network and parsed on the receiving system into its own, internal data structure which must not be the same as the transfer format.

During first tests with graphs in the WebGen framework (Neun and Burghardt 2005) we used a very simple XML representation which is in fact just using a redundancy free adjacency list. The following example shows the sample XML code for a simple basic graph as shown in Figure 10:

```
<ixg:graph>
   <ixg:node ixg:idx="1">120.0,109.0</ixg:node>
   <ixg:node ixg:idx="2">208.0,567.0</ixg:node>
      <ixg:edge ixg:idx="1" />
   <ixg:node ixg:idx="5">458.0,297.0</ixg:node>
      <ixg:edge ixg:idx="2" />
      <ixg:edge ixg:idx="1" />
   <ixg:node ixg:idx="3">765.0,512.0</ixg:node>
      <ixg:edge ixg:idx="5" />
      <ixg:edge ixg:idx="2" />
   <ixg:node ixg:idx="4">782.0,115.0</ixg:node>
      <ixg:edge ixg:idx="5" />
      <ixg:edge ixg:idx="1" />
      <ixg:edge ixg:idx="3" />
</ixg:graph>
```

This data format must be read in the correct order like a list as the "edge" elements do always link the preceding node with the node whose ID is contained in the "idx" attribute of the edge. Thus, this format does not contain any redundancy. This data format can be parsed very fast and converted to an OO graph representation. For direct querying, however, this transfer format is rather tedious.

## 4. DISCUSSION

### 4.1 Standardisation of Spatial Structures and Relations

Standardised functionalities to compute and use spatial and structural relationships are sparse in GIS. Furthermore, geographic databases or data transfer formats do commonly not include the modelling and storage of the advanced relationships that are discussed above.

Many algorithms for enriching data or for creating spatial relationships are available for different platforms and data formats. Interoperability, however, is not ensured and the different implementations of algorithms are not comparable. The development process of advanced generalisation operators takes more time because of the fact that algorithms or converters for the support data structures have to be generated first.

Languages exist (e.g. GXL) for expressing graph-like and matrix-like structures and they can be used in many ways. What is still missing today, however, is an agreement in the generalisation community on how to exploit XML-based languages to achieve a standardised data format for expressing the generalisation support structures. Such a format could then be used in the development of new generalisation support services and possibly also converters for already existing support structures would be developed. Following the categories of generalisation services (see section 1.3) the generalisation support services can then be the foundation of the more advanced generalisation operator and generalisation process services.

### 4.2 Persistency vs. Recalculation

The aim of this paper was to illustrate the possible use of generalisation support services as web services. This prompted the discussion about data models and formats for storing the support structures. Another point is the question why today most support structures are only generated at runtime and are not saved persistently. Some generalisation support structures are very expensive to calculate but could easily be saved persistently for multiple use, for instance, by other generalisation operators.

### 4.3 Usage of Generalisation Support Services

Generalisation support services on the one hand should deliver data structures which are expensive to calculate. On the other hand, and probably most importantly, a generalisation service should deliver the result of complex supporting algorithms, such as complex measures, support geometries or map feature relations, with an easy interface and a preferably simple output. The aim is to make support services available to developers of service-based generalisation architectures in such a way that they can use these in conjunction with generalisation operator services without having to know in detail how the support data is generated. One example is a Delaunay support service that can generate a Delaunay triangulation from a set of points which can later be queried, for instance, for the shortest edge or updated by removing the shortest edge.

## 5. CONCLUSION

In an earlier paper Burghardt et al. (2005) made a proposal for generalisation web services. In that paper three types of generalisation services were distinguished, including generalisation support services, generalisation operator services, and generalisation process services. In this paper, we dwell primarily on the generalisation support services, which are intended to enrich the raw input data with additional information such as shape or importance attributes and new geometries as well as spatial and structural relationships, hence providing direct support to the other two service categories. As a first contribution, this paper delivers a comprehensive taxonomy of generalisation support services in relation to different generalisation operators. Second, methods are proposed to represent, store and especially exchange the spatial relations generated by support services. Many relations can be expressed in a graph-like form. Thus, the proposed data structures and formats are mainly graph based. Finally, a number of important, yet still open issues are discussed, including standardisation of graph transfer formats, persistency vs. recalculation, and different paths for generalisation service exploitation. The latter issue – service usage – will be of particular importance to the further development of generalisation services. Apart from developing different 'business models' of how generalisation services might best be used, open questions include the right granularity of generalisation operators and support services (i.e. what are useful functional building blocks that may serve to develop service-based generalisation systems), as well as problems of partitioning generalisation problems so that they may be amenable to distributed processing. Obviously, there are still plenty of open problems remaining, as generalisation is revisited given a different architecture (service-based rather than standalone) and at least partially given more stringent constraints w.r.t. processing efficiency.

We plan to address the above issues step by step in the future by further extending the WebGen platform (Neun and Burghardt, 2005; Burghardt et al., 2005) which is intended to serve both as a demonstrator and a proof-of-concept. The generalisation support services discussed in this paper will be integrated into the WebGen platform. Currently first attempts with support services providing a graph as output are evaluated. These support structures can then be used by other generalisation operators over the web services interface. More specifically, we are experimenting with the use of triangulations (as a support service) in conjunction with building typification operator services.

## ACKNOWLEDGEMENTS

The research reported in this paper was partially funded by the Swiss NSF through grant no. 20-101798, project DEGEN. Thanks go to Alistair Edwardes and Matthias Bobzien for helpful comments.

## REFERENCES

Anders, K.-H.. 2004. *Parameterfreies hierarchisches Graph-Clustering-Verfahren zur Interpretation raumbezogener Daten*. Doctoral Thesis, Institut of Photogrammetry, University of Stuttgart, Germany.

Bader, M. 2001. *Energy Minimization Methods for Feature Displacement in Map Generalization*. Doctoral Thesis, Department of Geography, University of Zurich, Switzerland.

Bader M., M. Barrault and R. Weibel. 2005. Building displacement over a ductile truss. International Journal of Geographical Information Science. Vol. 19, No. 8–9, 915–936.

Bobzien M. and D. Morgenstern. 2002. Geometry-Type Chance in Model Generalization - a Geometrical or a Topological Problem? Proceedings of the Joint ISPRS/ICA Workshop on Multi-Scale Representations of Spatial Data, 2002, Ottawa.

Burghardt, D. and S. Meier. 1997. Cartographic Displacement Using the Snakes Concept. In: Förstner, W., and L. Plümer (eds), *Semantic Modeling for the Acquisition of Topografic Information from Images and Maps*, Birkhäuser Verlag, 59-71.

Burghardt, D., M. Neun and R. Weibel. 2005. Generalization Services on the Web – A Classification and an Initial Prototype Implementation. *Cartography and Geographic Information Science*, 32(4): 257-268.

Chithambaram, R., M.K. Beard and R. Barrera. 1991. Skeletonizing Polygons for Map Generalization. *Technical Papers ACSM*. Baltimore. Vol. 2, pp. 44 - 55.

Christophe S., Ruas A., 2002, "Detecting Building structures for generalisation purposes". In: D. Richardson & P. van Oosterom (ed.) *Advances in Spatial Data Handling*, Berlin , Springer, 10th international Symposium on Spatial Data Handling, (SDH'02, Canada/Ottawa), p419-432.

De Floriani L. and E. Puppo. 1995. Hierarchical triangulation for multiresolution surface description. *ACM Transactions on Graphics*, 14(4):363--411.

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

Edwardes, A., D. Burghardt, M. Neun. 2005. Experiments to build an open generalisation system. In: W. Mackaness, A. Ruas and T. Sarjakoski (eds), *Challenges in the Portrayal of Geographic Information: Issues of Generalisation and Multi Scale Representation*.

Fitzke, J., K. Greve, M. Müller, A. Poth. 2004. Building SDIs with Free Software – the deegree project. In: *Proc. 7th Conf. Global Spatial Data Infrastructure* (Bangalore, India).

Galanda, M. and R. Weibel. 2003. Using an Energy Minimization Technique for Polygon Generalization. *Cartography and Geographic Information Science*, 30(3): 259-275.

Harrie, L. 1999. The Constraint Method for Solving Spatial Conflicts in Cartographic Generalization. *Cartography and Geographic Information Science*, 26(1): 55-69.

Harrie, L. and R. Weibel. 2006. Modelling the Overall Process of Generalisation. In: Mackaness, W.A., Ruas, A. & Sarjakoski, T. (eds.). *Generalisation of Geographic Informations: Methods and Applications*. Elsevier Science. Forthcoming.

Højholt, P. 2000. Solving Space Conflicts in Map Generalization: Using a Finite Element Method. *Cartography and Geographic Information Science*, 27(1): 65-73.

Holt R.C., A. Winter and A. Schürr. 2000. GXL: Toward a Standard Exchange Format. http://www.gupro.de/GXL/ (accessed 11/2005)

Horton, R.E. 1945. Erosional development of streams and their drainage basins: Hydrophysical approach to quantitative morphology. *Geological Society of America Bulletin*, 56, 275-370.

ICA. 2004. Brain storming Sessions. ICA Workshop on Generalisation and Multiple Representation (Leicester), available from http://ica.ign.fr/

Illert, A. and S. Afflerbach. 2004. Global schema specification. GiMoDig-project, IST-2000-30090, Deliverable D5.3.1, Public EC report, 35 pgs., http://gimodig.fgi.fi/deliverables.php (accessed 01/2005)

Jones C.B., G.L. Bundy and J.M. Ware. 1995. Map Generalisation with a Triangulated Data Structure. *Cartography and Geographic Information Systems* 22(4), pp 317-331.

JTS. 2006. The JTS Topology Suite, Vividsolutions. http://www.jump-project.org/project.php?PID=JTS&SID=OVER (accessed 11/2005)

JUMP. 2006. The JUMP Unified Mapping Platform. http://www.jump-project.org/project.php?PID=JUMP&SID=OVER (accessed 11/2005)

Lehto, L., and T. Sarjakoski. 2004. An Open Service Architecture For Mobile Cartographic Applications. In: G. Gartner (ed.), Location Based Services & TeleCartography, *Proceedings of the Symposium 2004*, Vienna University of Technology, January 28-29, 2004, Vienna, 141-145.

Lüscher P. 2005. Matching von Straßendaten stark unterschiedlicher Maßstäbe und Aufbau einer MRDB. *42. Sitzung der Arbeitsgruppe Automation in Kartographie, Photogrammetrie und GIS*, Wien.

Mackaness W.A. and M.K. Beard. 1993. Use of graph theory to support map generalization. Cartograohy and Geographic Infromation Systems, 20.

McMaster, R., and S. Shea. 1992. *Generalization in Digital Cartography*. Washington D.C: Association of American Geographers.

Mustière S. and B. Moulin. 2002. What is Spatial Context in Cartographic Generalisation? *Conference Geospatial Theory, Processing and Applications*, IAPRS & SIS, vol. 34, No. 4 Ottawa, Canada. Pages 274-278.

Neun, M., R. Weibel and D. Burghardt. 2004. Data Enrichment for Adaptive Generalisation. In: *7th ICA Workshop on Generalisation and Multiple Representation* (Leicester), available from http://ica.ign.fr/

Neun, M. and D. Burghardt. 2005. Web Services for an Open Generalisation Research Platform. In: *8th ICA Workshop on Generalisation and Multiple Representation* (A Coruña), available from http://ica.ign.fr/

OGC. 1999. OpenGIS® Simple Features Implementation Specification for SQL. http://www.opengeospatial.org/specs/ (accessed 11/2005)

OGC. 2002. The OpenGIS® Abstract Specification, Topic 12: OpenGIS Service Architecture, Version 4.3, OGC 02-112. http://www.opengeospatial.org/specs/ (accessed 11/2005)

OGC. 2004. The OpenGIS® Geography Markup Language (GML) Encoding Specification, Version 3.1.1, OGC 03-105r1. http://www.opengeospatial.org/specs/ (accessed 11/2005)

Petzold I., D. Burghardt and M. Bobzien. 2005. Automated Derivation of Town Plans from Large Scale Data in a Cartographic Production System. In: *Proc. XXII International Cartographic Conference (ICC2005),* A Coruña, Spain.

Plazanet C., J.-G. Affholder and E. Fritsch. 1995. The importance of geometric modeling in linear feature generalization. *Cartography and Geographic Information Systems*, 22(4):291–305.

Regnauld, N. 2005. Spatial Structures to Support Automatic Generalisation. In: *Proc. XXII International Cartographic Conference (ICC2005),* A Coruña, Spain.

Ruas A. 1998. A Method for buiding displacement in automated map generalisation. *International Journal on GIS*, vol12 no8 p789-803.

Ruas, A. 2000. The Roles of Meso Objects for Generalisation. In: Proceedings *9th Symposium on Spatial Data Handling* (Beijing, China): 3b50-3b63

Sarjakoski, T., Sester, M., Sarjakoski, L.T., Harrie, L., Hampe, M., Lehto, L. and T. Koivula. 2005. Web generalisation services in GiMoDig - towards a standardised service for real-time generalisation. In: Toppen, F., and M. Painho, (eds.), Conference proceedings, AGILE 2005, 8th Conference on Geographic Information Science, Estoril, Portugal, May 26-28, 2005, pp. 509-518.

Sester, M. 2000. Generalization Based on Least-squares Adjustment. In: International *Archives of Photogrammetry and Remote Sensing*, Vol. XXXIII, Part B4, Amsterdam, 931-938.

Sester, M., L. T. Sarjakoski, L. Harrie, M. Hampe, T. Koivula, T. Sarjakoski, L. Lehto, B. Elias, A.-M. Nivala, and H. Stigmar. 2004. Real-time Generalisation and Multiple Representation in the GiMoDig Mobile Service. GiMoDig-project, IST-2000-30090, Deliverables D7.1.1*, D7.2.1* and D7.3.1, Public EC report, 151 pgs. http://gimodig.fgi.fi/deliverables.php (accessed 01/2005)

Smaalen, J.W.N. van, 2003. Automated Aggregation of Geographic Objects: A New Approach to the Conceptual Generalisation of Geographic Databases. Doctoral dissertation, Wageningen University and Research Centre.

SOAP. 2000. Simple Object Access Protocol (SOAP) 1.1. http://www.w3.org/TR/soap/ (accessed 11/2005)

Strahler, A.N. 1952. Dynamic basis of geomorphology. *Geological Society of America Bulletin*, 63, 923-938.

Timpf, S., 1998, Hierarchical structures in map series. Phd Thesis, Technical University of Vienna.

Thomson R.C. and D.E. Richardson. 1995. A graph theory approach to road network generalization. In: *Proceedings of the 17th ICA Meeting*, Barcelona, Spain.

UDDI. 2005. Universal Description, Discovery, and Integration. http://www.uddi.org/ (accessed 11/2005)

Ware, J.M., C.B. Jones and N. Thomas. 2003. Automated Map Generalization with Multiple Operators: A Simulated Annealing Approach. *International Journal of Geographical Information Science*, 17(8): 743-769.

Werner, C., 1988. Formal Analysis of Ridge and Channel Patterns in Maturely Eroded Terrain. *Annals of the Association of American Geographers*, 78 (2): 253- 270.

Weibel, R., 1992. Models and Experiments for Adaptive Computer- Assisted Terrain Generalization. *Cartographic and Geographic Information Systems*, 19 (3): 133-153.

Woods J.D. and S. Rana. 2000. Constructing Weighted Surface Networks for the Representation and Analysis. In: *5th International Conference on GeoComputation*, UK.