A Review of the Clarity Generalisation Platform and the Customisations Developed at Ordnance Survey Research

Patrick Revell
patrick.revell@ordnancesurvey.co.uk
Ordnance Survey® Research, Romsey Road, Southampton, UK

Abstract

Clarity is a generic software platform dedicated to automatic generalisation, produced by 1Spatial. This paper begins by outlining the origins, software architecture and features of Clarity. The database format used by Clarity is presented along with a discussion of the methods for importing and exporting data. There is a section describing the spatial structures used by Ordnance Survey Research for developing generalisation tools, which include topology, Delaunay triangulation, proximity graphs, skeletons and networks. This is followed by a section describing the methods used for partitioning large datasets in preparation for generalisation.

The paper continues with details of all the algorithms developed by Ordnance Survey Research on the Clarity platform. The methods for constructing generalisation flowlines are discussed along with a set of example applications. The paper concludes with an overview of the results achieved at Ordnance Survey using the Clarity system and a outlook on the future direction of this work.

KEYWORDS: Clarity, generalisation, Java, Agents, topology, spatial structures, measures.

1 Introduction

1.1 Origins of Clarity

Clarity is a generic software platform dedicated to automatic generalisation, produced by 1Spatial, formerly Laser-Scan (1Spatial 2008). It consolidates the experience gained from the prototype developed by the European AGENT (Automatic GENeralisation New Technology) research project (Barrault et. al. 2001, Agent 2000). The initial development of Clarity was funded by IGN France, KMS Denmark, Ordnance Survey (OSGB) and IGN Belgium. This group of four NMAs forms the Mapping Agencies Generalisation NETwork (MAGNET). In July 2003, the first version of Clarity software was provided to all the MAGNET partners (Lecordix et. al. 2005). The MAGNET partners coordinate Clarity developments with 1Spatial and now have an agreement for sharing their custom code with each other (Regnauld et. al. 2007).

1.2 Features of Clarity

Clarity makes use of the proprietary Gothic object-oriented database technology. This database can store any number of geographic datasets. The datasets are versioned so that each version just stores the changes from the previous version. It is possible to backtrack to any previous version. Within a dataset a schema can be defined, which includes feature classes, attributes and methods. Multiple inheritance is supported between feature classes. *References* can be defined, which allow objects to be linked much more efficiently than in a relational database.

The object-oriented database provides the foundation for the Agent system originally prototyped during the AGENT project and re-implemented for inclusion in Clarity. At a basic level this allows Gothic objects (*agents*) in a dataset to perceive their surroundings, decide if they conform to particular geographic/cartographic rules (*constraints*), and propose generalisation algorithms (*actions*) to improve their conformance (*satisfaction*). The happiness of an agent is computed from the satisfaction of its constraints. An agent is able to find the best compromise between a set of (possibly conflicting) constraints using an iterative search method to maximise its happiness. All previous states are stored on an agent, so it can backtrack to previous states during its search for the optimum state.

Clarity also has the benefit of the Gothic topology engine. This can structure spatial data ‘on the fly’ as objects are created in a dataset and store topological relationships persistently. 1Spatial have made

this powerful topology engine available for use directly on an Oracle database in the Radius Topology product. Details of the topology model are provided in section 2.1.

Clarity comes with a set of algorithms for generalising buildings and roads, which were developed and tested during the AGENT project and augmented during subsequent projects funded by IGN France and KMS Denmark (Lecordix et. al. 2005). It also has a good geometry-handling API which allows new generalisation algorithms to be developed. There is a facility for creating and navigating through *markups* on objects, which is useful for flagging places where generalisation has failed, for later manual inspection/correction.

1.3 Clarity Architecture and Customisation

The Gothic database technology has been available for many years via the Gothic Developer suite of products. This comprises three main applications: Manage (for Gothic database management), Translate (for transferring spatial data to/from Gothic datasets) and LAMPS2 (for interactive editing and automated processing of data in Gothic datasets). The LAMPS2 Generaliser is part of this package, which was an early attempt at an automated generalisation solution. Gothic Developer allows customisations to be developed using a bespoke procedural language called Lull. This language is time-consuming to use, it is not object-oriented and there is no simple way of calling functions from external software libraries.

During development of Clarity the new Java-In-Gothic API was created. This provides access to the same core functionality, but using the Java language. This offers benefits of object-oriented programming, rapid user-interface design and easy integration with powerful external Java libraries. 1Spatial are yet to make all the Lull functionality available in Java, including most of the generalisation algorithms. However they have fortunately supplied a workaround which allows Lull to be called from Java; a technique referred to as *Java wrapping*. Custom Clarity developments at Ordnance Survey have primarily been written in Java, and Lull is only used when there is no alternative.

Clarity is a Java application which makes use the Java-in-Gothic API to display, query and run automated processes on objects in a Gothic dataset. It is very easy to add new menus to Clarity, which can be used to invoke custom Java processes. Unfortunately the standard Clarity user interfaces are quite limited and do not offer the rich functionality available in Gothic Developer. Therefore Manage, Translate and LAMPS2 are supplied as legacy packages with Clarity. These require emulation software to run in a Windows environment, which adds an additional cost.

1.4 Database Management and Batch Processing

Clarity contains no tools for Gothic database management, so a suite of custom Java tools have been developed at Ordnance Survey which provide an alternative to Manage. There are now Java tools to create/delete a database and list all the current databases. When attached to a particular database, there are tools to create/delete a dataset and list all the datasets.

In addition there are tools to attach to, commit edits to, discard edits from and backtrack a dataset. All these tools are also available for use in batch mode, which is extremely useful for production-oriented tasks where a graphical user interface is not required.

The ability to invoke Clarity in batch mode has enabled Clarity to be set up on a web server at Ordnance Survey. A test has been carried out to prove that Clarity generalisation can be employed remotely from any web client by transferring spatial data and parameters across the internet using the WebGen technology (Neun and Burghardt 2006)(Regnauld 2006).

1.5 Data Import and Export

Before any generalisation can be carried out, it is necessary to import spatial data into a Gothic dataset. Once generalisation is complete, the modified data must be exported back. This dependence on the Gothic database is a major restriction at a time when organisations are increasingly opting for off-the-shelf enterprise database solutions, such as Oracle. 1Spatial do have aspirations of making Clarity work directly on an Oracle database by transferring data 'behind the scenes' to and from Oracle. They developed a prototype for this with Intergraph in 2004 (Watson and Smith 2004). A

current partnership between 1Spatial and Star-APIC has developed a GML3 transfer between Clarity and the Oracle-enabled MercatorDB cartographic styling software (1Spatial 2007).

A Clarity import and export routine has been developed at Ordnance Survey for the current in-house tile-based format used to hold Ordnance Survey large scale data. There is currently a project at Ordnance Survey which is moving this data into a seamless Oracle database. Therefore if Clarity is to be viable as a generalisation solution, it will be essential for it to interoperate with Oracle. For research purposes this is not required, and Shape files are the most common method for transferring data.

Custom Clarity shape file import and export interfaces have also been developed at Ordnance Survey, which allow these operations to be easily carried out without using the Translate application. This includes a new interface where a mapping can be defined between Shape file attributes and Gothic attributes. These interfaces are shown in Figure 1. Sometimes shape files have "coded values", where attribute strings have been replaced by an enumeration. A function has been developed which allows the enumeration metadata to be imported, and the coded values translated back to string attributes.

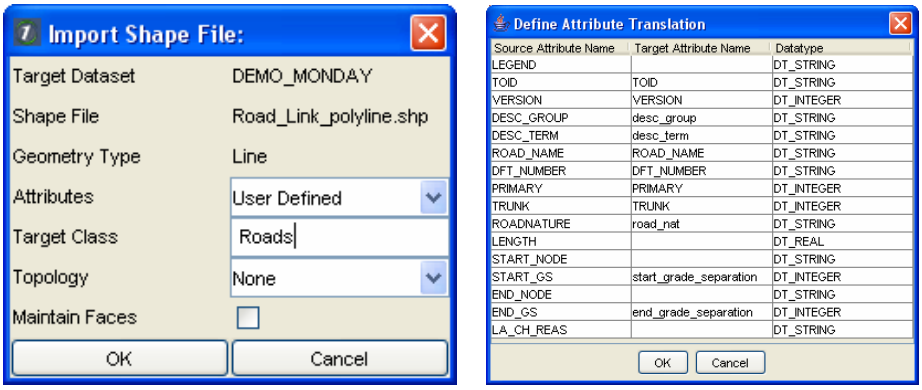

Figure 1. Shape file import and attribute translation interfaces.

It has also been necessary to read data in the formats used by Ordnance Survey external products. An OS MasterMap® reader has been developed which can read GML for the Integrated Transport Network™ (ITN) and the Topography Layer products. A simple GML exporter has also been developed, which can export any class and a selection of its attributes to GML. When conducting generalisation research it is useful to overlay generalisation results over manually created maps, for comparison purposes. A function has been developed which allows a raster in ECW format (ER Mapper 2008) to be viewed within Clarity.

1.6 Symbolising Spatial Data

There are no user interfaces for symbolising spatial data in Clarity. To achieve this it is necessary to write *display methods* in Java or Lull. A display method can be used to customise the display of objects in a feature class based on their geometric, semantic and topological properties. This technology is extremely flexible, although quite time-consuming to set up.

For creating finished maps a function has been created at Ordnance Survey which saves results to geo-referenced raster tiles in any Java-supported image format, such as JPEG, TIF, BMP and GIF. A function has recently been developed which allows maps to be exported directly to Postscript from Clarity, which can be subsequently converted to high-quality pdfs.

2 Spatial Structures

It has long been acknowledged that automating the process of generalisation requires a phase of analysis to detect the geographic structures or phenomena which are relevant to the target map (Brassel and Weibel 1988). Spatial relationships are often not modelled explicitly in spatial data. They usually express some aspect of proximity between features and can be quite computationally expensive to derive. Given that this information is heavily used during the generalisation process, it is clear that these relationships should be stored, at least temporarily, into *spatial structures*. Such data structures allow spatial relationships to be expressed between geographic features (Regnauld 2005).

2.1 Topology

Topology is a mathematical concept sometimes defined as “characteristics of geometry that do not change when the coordinate space is deformed”. It explicitly expresses geometric relationships, such as “connects to”, “touches”, “adjacent to” or “within”. The Gothic topology engine offers much better topology support than other GIS. The topological relationships are stored persistently in the dataset using node (point), link (line) and optionally face (area) primitives. References express the spatial relationships between the primitives and also specify how real-world objects are constructed from these primitives. This model covers a substantial subset of the ISO 19107 Spatial Schema standard, and includes support for dynamic update of the primitives in response to change to real-world features (Hardy et. al. 2003).

Although data can be topologically structured ‘on the fly’ at the time of import, it is sometimes preferable to import the data unstructured, either because it makes the import quicker or because there are topology errors which are better to resolve later. A custom function has been developed at Ordnance Survey for topological structuring, which improves upon the LAMPS2 function by attempting to resolve topology errors. If an object fails to structure, the structuring is automatically reattempted several times using different tolerances. Often this approach will result in successful structuring and if not the error is marked up for later correction. Area primitives (faces) are optional at the time of structuring, but a function has been developed which allows faces to be constructed retrospectively.

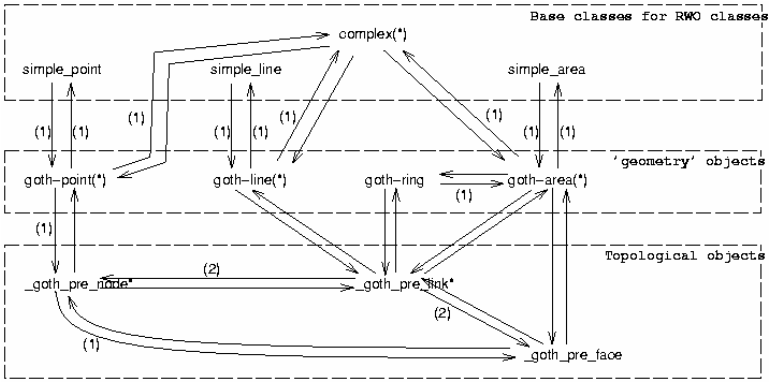

Figure 2. Gothic topology model (1Spatial 2006).

The Gothic topology model is illustrated in Figure 2. The real world objects (RWO) must inherit from the base-classes shown at the top of the diagram. The ‘geometry’ objects are hidden from the user and serve to decouple the geometry from the RWO. A given set of node, link and face classes is termed a manifold, and they must inherit from the base-classes shown at the bottom of the diagram. By default there is only one manifold, but it is possible to have any number of different manifolds which separate out the topological relationships between the different real world classes. For example topographic landcover polygon data could make use of one manifold and a road network could use another. Each class can have a set of relationships between other classes, expressed in terms of a snapping tolerance and a topology rule. These rules are illustrated in the three columns of Figure 3.

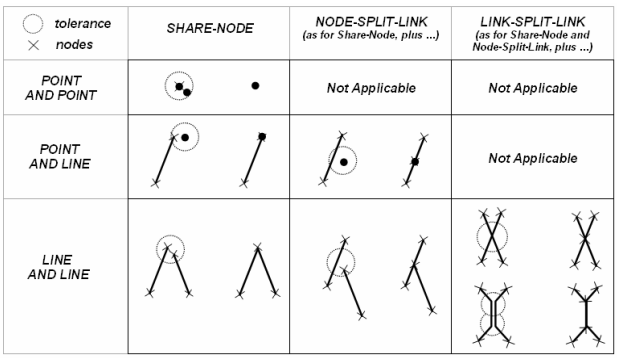

Figure 3. Gothic topology rules (Laser-Scan 2002).

The Java interface shown in Figure 4 has been developed at Ordnance Survey for defining topological rules in Clarity. This improves upon on the LAMPS2 interface by clearer visualisation of manifolds, topological classes and their rules. Manifolds are listed on the left and under each manifold the real world classes are grouped into points, lines and areas. Selecting a real world class displays the topological rules which that class has with other classes, including the tolerance and the rule type.

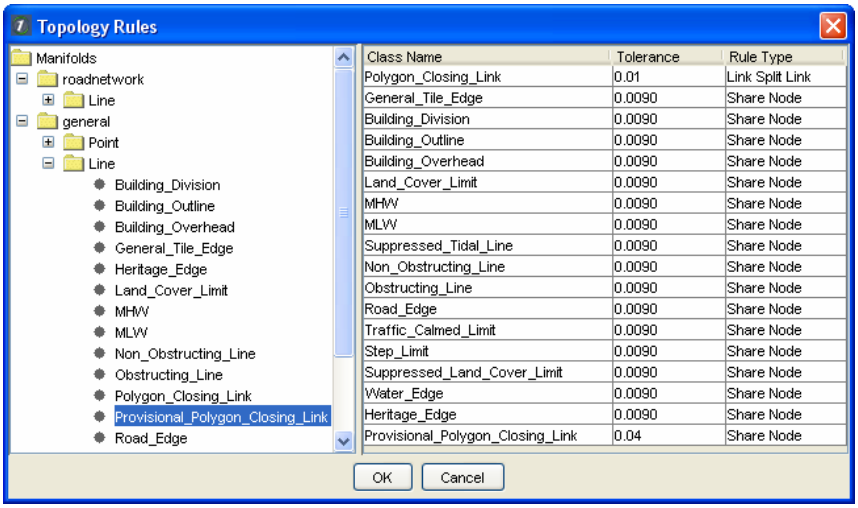

Figure 4. New interface for viewing and editing topology rules.

A large API has been developed at Ordnance Survey for querying real world objects and their topological primitives for connectivity information. This is much faster and more efficient than spatial querying.

Various API functions have been developed at Ordnance Survey for performing topological modifications, such as creating new lines from links (allowing polygons to be converted to lines, or lines to be broken down into their component parts at intersections) and creating new polygons from sets of links (allowing existing polygons to be merged together or reclassified). It is also possible to modify primitive geometries directly, provided that the modified links do not intersect. Simplifying a link will automatically update the area and line features which rely on that link. Moving a node will automatically update the point, line and area features that share the node.

2.2 Delaunay Triangulation and Proximity Graphs

Whilst topology is useful for expressing relationships between features that touch, it is often necessary to compute proximity relationships between non-touching features. For this purpose a new Delaunay triangulation Java API has been developed at Ordnance Survey. Being object-oriented this is significantly more powerful than the Lull-only delaunaylib supplied with Clarity. A Delaunay triangulation is a graph (i.e. a set of nodes connected by edges). Edges of the triangulation do not cross and the triangles formed by the edges have the property that their circum circle does not contain any other node of the graph.

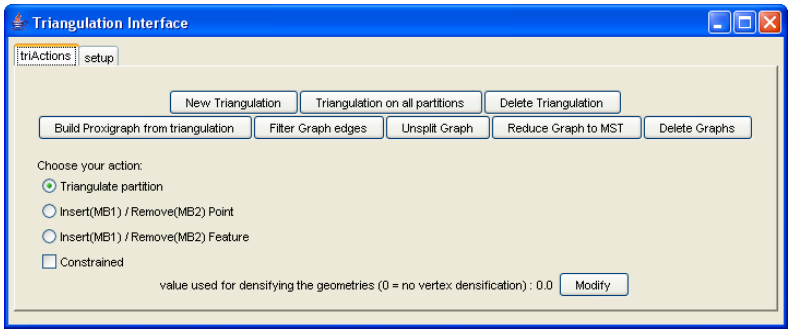

Figure 5. Triangulation interface.

When computing proximity between geographic features, there is an additional requirement that the edges of the triangulation should not cross the outline of these features. The API supports constrained triangulation which meets this requirement by relaxing the circum circle rule. An alternative is to add vertices to the geometries (densifying) until no feature outlines are crossed by a triangulation edge (conforming triangulation). Once a triangulation has been built, it is straightforward to construct a proximity graph, which can be used to group features according to distance. There is also an option to reduce the proximity graph to a minimum spanning tree (Regnauld 2005). The triangulation interface is shown in Figure 5.

2.3 Skeletons and Networks

Delaunay triangulation can be used for constructing a centreline skeleton. At Ordnance Survey this has been used in two instances. The first application is for collapsing road network dual carriageways to single centrelines. The process starts by using the topology structure to track dual carriageways, producing a set of strokes. The strokes are broken up at intersections, and the resulting edges and nodes are used to populate a graph. The graph is used to pair up the ends of the strokes and hence pair up the dual carriageways. The dual carriageways are finally collapsed down to single centrelines using a skeleton computed from a triangulation (Thom 2005). In places this can result in the network to becoming disconnected, so short line 'tags' are added to ensure that the topological connectivity is retained. An example result is shown on the right of Figure 6.

Ordnance Survey large scale data contains both a road network and road polygons. An algorithm has been developed at Ordnance Survey for attributing road centrelines with widths computed from their underlying polygons. This allows the road classification to be enhanced.

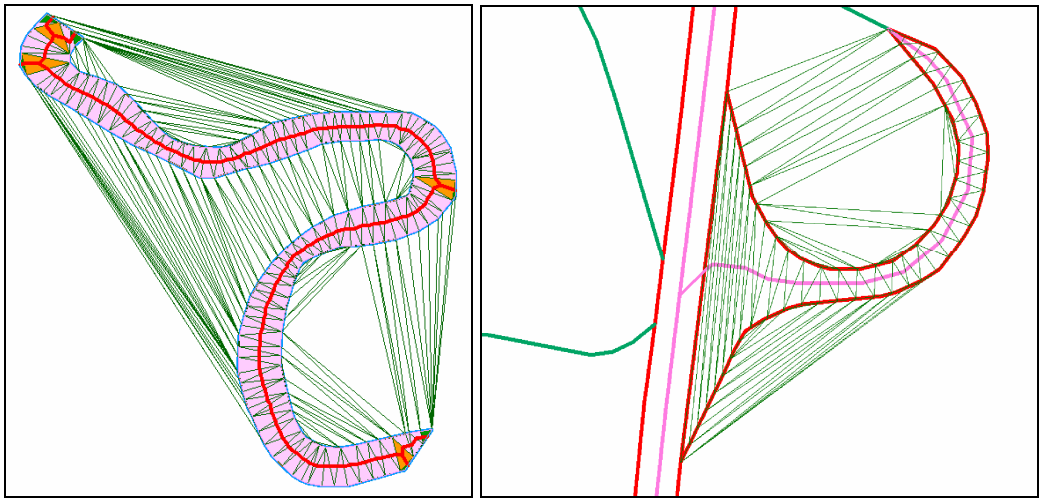

Figure 6. Examples of a skeleton built from a triangulation: water polygon centreline (left) and dual carriageway centreline (right). Ordnance Survey © Crown Copyright. All rights reserved.

The second example application of a skeleton is for computing a path, track or water centreline inside a polygon. The derived skeleton comprises skeleton edges and nodes, which hold connectivity information. Individual skeletons can be joined together for polygons which are adjacent or in close proximity to form a skeleton network. The centreline for a river polygon is shown of the left of Figure 6.

In Ordnance Survey large scale data, water is broken when it passes under other topographic features such as roads. In order to generalise this data in a consistent manner, it is necessary to construct hydrology networks from the disjoint water. For this purpose an algorithm has been developed at Ordnance Survey which starts by collapsing all water polygons to centrelines. Gaps in the network are deduced and reconnected, with a confidence factor being attached to each deduced link to facilitate manual checking. The network is finally classified into categories according to the widths of the source polygons (Regnauld & Mackaness 2006). The results are shown in Figure 7.

Similar network deduction techniques have been applied to create a path and track network from the disjoint path and track polygons in Ordnance Survey large scale data. The algorithm adds additional

links to connect the path and track network to the road network. Eventually these techniques will be adapted to construct a railway network from the disjoint rail lines in Ordnance Survey large scale data.

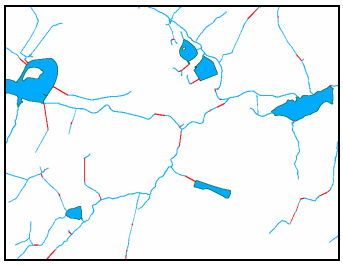

Figure 7. Automatically deducing connections (red) in a hydrology network (blue). Ordnance Survey © Crown Copyright. All rights reserved.

2.4 Clustering

Proximity graphs are useful when handling a small number of features where all the calculations can be performed in memory. For a large dataset it is often necessary to group together objects into proximity clusters. It is not currently possible to create an in-memory triangulation for a large number of features, so clustering is the preferred option. A Java clustering algorithm has been developed at Ordnance Survey which uses the Clarity buffering function to compute proximity relationships. This algorithm is described in (Revell et. al. 2005).

3 Partitioning

Partitioning is another aspect of structure recognition. It is essential for automatic generalisation, both as part of a "divide and conquer" strategy and for applying different types of generalisation to different regions, depending upon their characteristics.

3.1 Urban/Rural Identification

A technique for automatically identifying urban areas has been developed at Ordnance Survey that uses the clustering functionality to group together buildings. The resulting clusters are simplified using a utility function that removes holes below a threshold size and simplifies the geometries using combination of Douglas Peucker (1973) simplification and morphological erosion/dilation. The final geometries are classified into a number of parameterised buckets, which in the case of buildings represent urban areas an rural settlements (Revell et. al 2005). A sample result is shown in Figure 8.

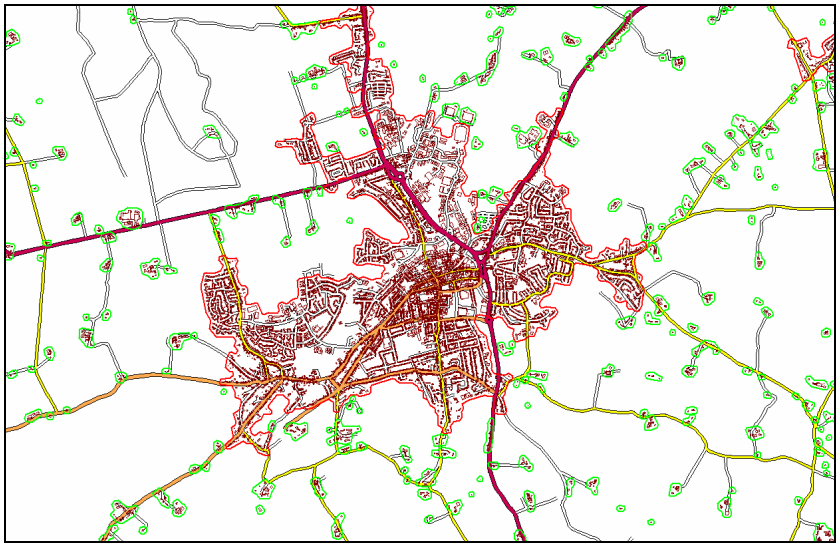

Figure 8. Automatically detected urban areas (red) and rural building clusters (green). Ordnance Survey © Crown Copyright. All rights reserved.

3.2 Hydro-Transport Network Partitioning

Clarity comes with a partitioning process which creates polygons from closed loops in topologically structured networks. This is hard to set up, so a utility function has been added at Ordnance Survey, that allows any number of line classes to be selected. If the data are not already structured, topological structuring is performed automatically. There is an option to create a convex hull around the outside of the data, which prevents “open” partitions around the edge of the dataset.

The Clarity partitioning can run out of memory when processing a large dataset, so a workaround has been developed that processes the urban areas and rural settlements one by one, and only creates partitions inside these regions. Partitioning is key to automated generalisation, so it is hoped that 1Spatial will make this functionality more accessible and scalable.

Once partitions have been created it is possible to generalise the features inside each partition autonomously from the other partitions. Thus a large dataset can be processed in manageable units.

3.3 Road Network Conflict Clustering

Hydro-transport network partitioning is not a total partitioning solution, since it overlooks how the networks themselves will be generalised. For example algorithms for displacing roads are typically unable to handle large networks. This problem was addressed and solved by IGN France during the Carto2001 project using *flexibility graphs* (Lemarie 2003). This solution was not easily adaptable to other data models, so an alternative approach was developed at Ordnance Survey. The new process first identifies conflicts between roads requiring displacement. These conflicts are stored as objects in the dataset and are grouped together by proximity and connectivity to form *conflict clusters* (Thom 2006). Each road only occurs in one conflict cluster, which effectively breaks the road network down into sub-networks.

4 Measures

Measures are a key part of any generalisation system. During an Agent process they can be used by a constraint to determine if it is satisfied, and if not, it can use them to decide which actions to invoke. They can also be used internally by algorithms to guide generalisation decisions. Clarity comes with area, concavity, shortest edge, squareness and minimum bounding rectangle measures in Lull. There also some Lull measures supplied with Clarity which were originally developed by IGN France for generalising sinuous mountain roads.

At Ordnance Survey a set of measures has been developed in Java. There are some general line and polygon measures and some specific measures applicable to buildings and roads. There is not space in this paper to detail these measures. They can all be invoked from menus on the user interface, which permits individual objects to be measured or an entire feature class to be processed, storing the results of the measure as an attribute on the objects.

5 Generalisation Algorithms/Actions

An algorithm that has an interface to the Agent system is termed an *action*. All the generalisation algorithms described below are available in Java, but not all of them have been enabled as actions yet. They can all be invoked as standalone processes, such as from Clarity menus, which permits individual objects to be generalised or an entire feature class to be generalised. Having all the algorithms available as menu items is useful when experimenting with parameters and sequences of algorithms. Note that Clarity has no functionality for generalising railways and Ordnance Survey have not yet addressed this research issue.

5.1 Line simplification, filtering, smoothing and interpolation

A basic Douglas Peucker (1973) line filtering algorithm is available in Clarity. A more sophisticated Java implementation of this algorithm has been added by Ordnance Survey which uses recursive convex-hull based linear feature partitioning (Zhou and Jones 2001). A weighted effective area (WEA) algorithm has also been developed at Ordnance Survey (Zhou and Jones 2004), which provides a generic implementation of line generalisation, based on weighted effective area. This is an extension

of the algorithm described by (Visvalingam and Whyatt 1993) and can be parameterised to give the same result as their algorithm. Sample results are shown in Figure 9.

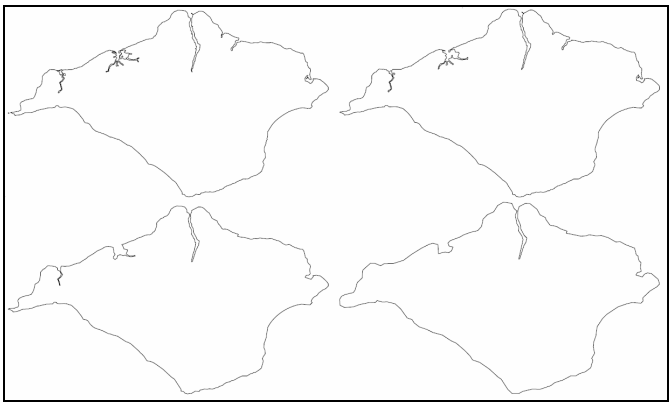

Figure 9. The coastline of the Isle of Wight generalised using various WEA parameter values.

There are some line simplification algorithms in Clarity which were developed at IGN France for generalising sinuous mountain roads. These include 'min break', 'max break', 'plaster', 'accordion' and 'bend remove'. An application has not yet been found for these algorithms on Ordnance Survey data.

Clarity comes with a good set of algorithms for line smoothing. These are not easy to access, so a single Java class has been developed at Ordnance Survey that provides interfaces to all of them. The available algorithms include Gaussian smoothing, Akima interpolation, three types of Cubic Spline interpolation, McConalogue interpolation and Linear interpolation.

5.2 Polygon generalisation

A very basic set of polygon generalisation algorithms were initially developed at Ordnance Survey that delete small areas, remove small holes from polygons, simplify polygons (by applying Douglas Peucker to the inner and outer rings) and enlarge polygons by a scale factor. A simple dissolve algorithm was developed to merge together touching polygons. A shrink-wrap aggregation algorithm from the LAMPS2 Generaliser was made available in Java. The problem with these algorithms is that they ignore the generalisation context, causing the resulting polygons to overlap with each other or intersect with network features. A more sophisticated suite of polygon generalisation algorithms have now been implemented by Ordnance Survey, which make use of topological primitives.

The *dissolve by attribute* algorithm merges together adjacent polygons with zero, one or more matching attributes. It is possible to specify line classes blocking the dissolve, such as streams. There is also an option to delete line classes which have been dissolved across. The *dissolve small holes* algorithm removes holes below a threshold size, along with any polygon and (optionally) line features which they contain. The *dissolve small areas algorithm* merges polygons below threshold size either with the largest neighbour or the neighbour with the longest shared boundary. It is possible to specify a list of area classes not to merge into, such as buildings.

The *topological simplification* algorithm works by obtaining the set of links for each polygon and applying Douglas Peucker to each link. Updating the links modifies all connected polygon and line features, and it is possible to specify a list of classes not to simplify (eg buildings). These topological polygon generalisation tools have been successfully used for landcover generalisation (Revell 2007a, 2007b).

5.3 Building generalisation

There are three building simplification algorithms available. Two of them are part of Clarity and have now been Java wrapped by Ordnance Survey; one is from the LAMPS2 Generaliser and the other is from the AGENT project. The third algorithm is a Java implementation of the algorithm currently in use for Ordnance Survey 1:10 000 scale production. In addition there is a Java wrapping of the Clarity

local building enlargement algorithm. This has been adapted by Ordnance Survey to incrementally enlarge the narrowest part of a building, which is more likely to produce a result than performing the enlargement in one step.

On the contextual side, Ordnance Survey have Java wrapped the Clarity building amalgamation algorithm. The Ruas building displacement algorithm (1998) implemented during the Agent project has not been included in Clarity, but IGN France have provided an improved version in Lull.

In 2004 Ordnance Survey Research investigated generalising large scale buildings to 1:50 000 scale. It was found that the Clarity algorithms did not produce results conforming to the amalgamation style used for Ordnance Survey maps. Therefore bespoke tools were developed at Ordnance Survey in Java. There is a significant difference in the generalisation style between urban and rural areas, so the urban/rural clusters were used for applying different algorithms to these regions. Results of the amalgamation are shown in Figure 10. Partitions were used inside the urban/rural clusters to break down the generalisation process into autonomous units.

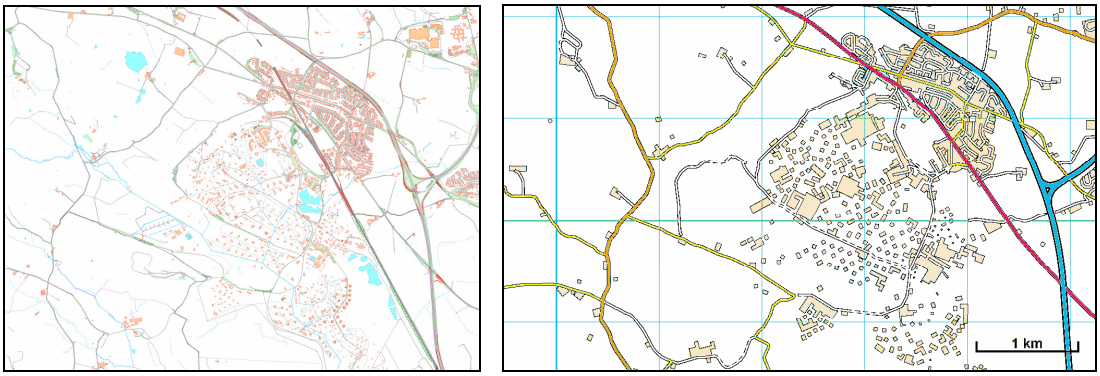

Figure 10. Buildings in OS MasterMap (left) generalised to 1:50 000 scale (right).

In rural areas buildings are amalgamated into shapes comprised of straight segments and right angles. For small groups of buildings a simple algorithm was developed which creates oriented squares and rectangles representing the positions and orientations of the original buildings. For larger, more complicated groups, an amalgamation algorithm was developed which initially encloses the buildings with the smallest geometry aligned to the general orientation of the group, comprised of straight segments and right angles. Algorithms were added to simplify this geometry by removing concave corners, enlarging juts and widening narrow parts.

To ensure the building fits well with the road symbolisation, algorithms were developed to adjust the building so that a sufficient portion of the geometry is visible, and there are no slivers of small holes between the building and the road (Revell 2004). All these algorithms have been set up to be invoked by constraints in an Agent process, allowing the result to be optimised.

For the urban building generalisation a growing tide algorithm was implemented. Buildings inside an urban partition are split into groups using a proximity graph computed from a triangulation. This approach allows restrictions to be applied which prevent buildings from being grouped together across railways, hydrology features and dead end roads. The shadow of each building group on the surrounding roads is computed and buffered by the minimum building width. Any buildings outside these initial buffers are included using the rural building amalgamation and simplification algorithms. The last step is to eliminate any small holes. The urban building generalisation is also controlled by an Agent process (Revell et. al. 2005, Regnauld and Revell 2007).

5.4 Road network generalisation

A major advance has been made in road network displacement which makes use of the conflict clusters described in the section 3.3. The Agent system has been configured to allow both the Beams (Bader 2001) and Push (Sester 2005) displacement algorithms to be invoked. The conflict resolution is

automatically evaluated and the best result is selected (Thom 2007). In some cases dead ends are trimmed back to reduce the conflicts. The results are shown on the right of Figure 11.

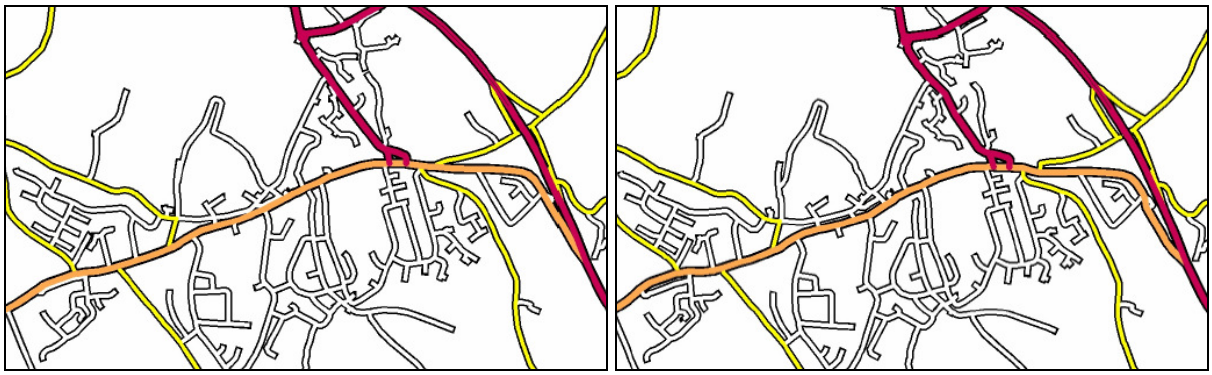

Figure 11. Automatic network displacement: source roads (left), displaced roads (right). Ordnance Survey © Crown Copyright. All rights reserved.

The Beams line displacement algorithm is supplied with Clarity. The implementation uses topology to retain the connectivity of the network during displacement. Beams is complex to set up, so a Java utility function has been developed at Ordnance Survey, which allows beams to be run on a set of line classes, or just on a set of selected objects. In addition Ordnance Survey have licensed the Push displacement algorithm from the University of Hannover. This reads and writes shape files, so to call it from Clarity a function has been created which exports a subset of roads from Clarity, invokes Push, then re-imports the result back to Clarity.

If running displacement on a subset of data, it is possible that topological inconsistencies between features not directly involved in the displacement. Clarity comes with a diffusion function, which allows the displacement to be absorbed automatically into any topologically connected line and point features. This function has been Java wrapped at Ordnance Survey.

5.5 Landcover generalisation

An MSc project evaluated by Ordnance Survey developed an algorithm for generalising large scale tree polygons to 1:50 000 scale. This algorithm has subsequently been integrated into Clarity, parameterised and made available as a web generalisation service (Mackaness et. al. 2007).

A generic method for reclassifying landcover data was developed during a year-long project investigating generalising Ordnance Survey large scale data to 1:10 000 scale. This consists of a user interface for defining the rules to control the landcover reclassification process. The tools can be used on any polygonal landcover dataset, but are most suited to specifications permitting combinations of landcover types. The topology-based polygon generalisation tools were then used to simplify the data after reclassification (Revell 2007).

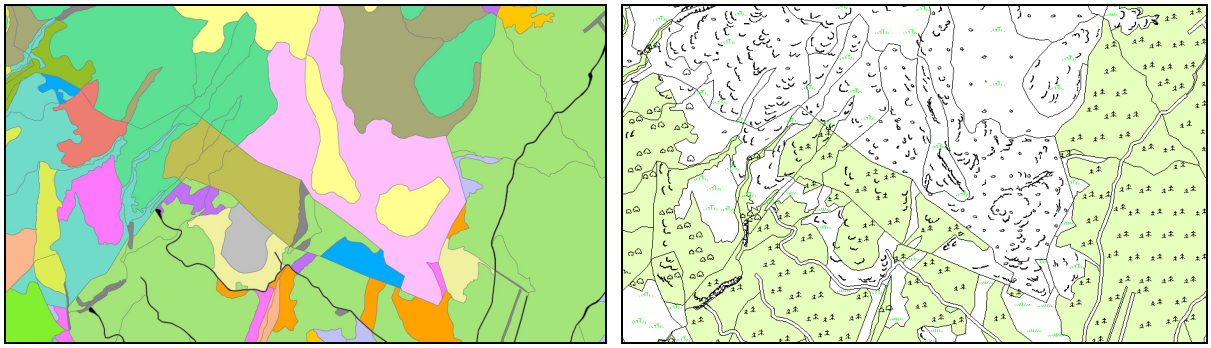

Figure 12. Large scale landcover polygons (left), automatically generalised and symbolised at 1:10 000 scale (right). Ordnance Survey © Crown Copyright. All rights reserved.

The generalisation results were symbolised using a group of flexible tools developed for landcover symbol placement. These extend the algorithms developed for 1:50 000 scale woodland symbol placement (Harrie and Revell 2007). The results are shown on the right of Figure 12.

5.6 Hydrology generalisation

Algorithms have been developed at Ordnance Survey for generalising hydrology, based on the results described in section 2.3. The network is first analysed to construct a hierarchy of rivers, using an technique based on the Horton Ordering (Horton 1945). This permits the main streams to be distinguished from the tributaries. The ordering allows the network to be pruned, following the principles described in (Thomson and Brooks 2000). It is possible to control which rivers are represented by polygons and which are represented by centrelines in the result. Where short sections of polygonal rivers fall below the minimum width, they are widened by buffering the centreline and merging it with the polygon. The rivers represented by centrelines required a light smoothing to remove artefacts left over from the skeleton construction (Revell et. al. 2007).

6 The Generalisation Flowline

6.1 The Agent System

Several examples have been presented when multiple generalisation algorithms have been sequenced together using the Agent system. One was for building generalisation and the other was for road displacement. The Agent system has proved useful for these cartographic generalisation problems, where the exact sequence of actions and choices of parameters are not known in advance.

The Agent system allows a hierarchy of agents to be deployed to control generalisation. Meso agents in the upper levels are used to control groups of features which have a geographic or cartographic meaning. Meso agents are able to create and control other meso agents, while at the bottom of the hierarchy they control micro agents. Micro agents are responsible for generalising individual geographic or cartographic features.

Clarity contains no interfaces for creating meso agents, so a set of utility functions have been developed at Ordnance Survey for this purpose. There is a function to create a single meso agent to control the generalisation of an entire feature class. There is also a function to take a given polygon class (such as partitions) and for each polygon initialise a meso agent to control the generalisation of the features inside that polygon (such as buildings).

A map specification is used to attach constraints to each type of agent. A constraint comprises a specific goal, a method for determining how well that goal has been achieved (*satisfaction*) and a plan of *actions* to achieve that goal. Constraints can be defined in Java or Lull. Ordnance Survey have developed of way of making constraints easier to define in Java. Clarity has a user interface for setting up the map specification, which can also be set up programmatically in Java or defined in XML.

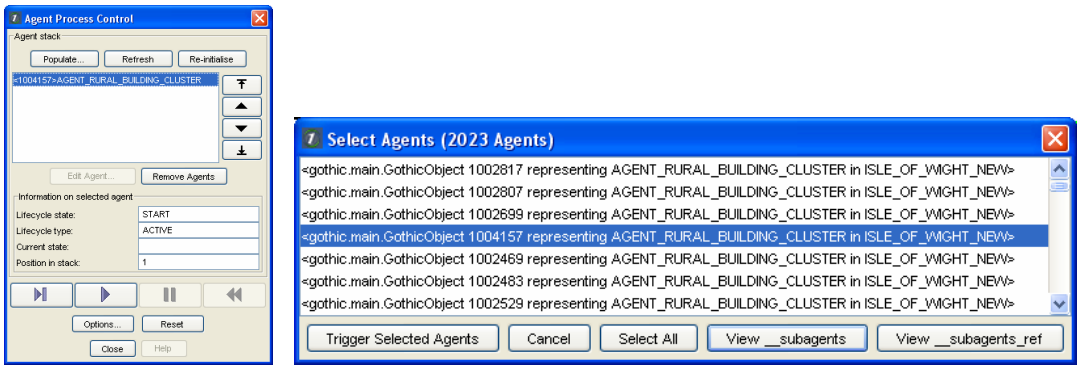

Figure 13. Agent Process Control (left) and a new interface for triggering agents (right).

The Agent system has a scheduler which maintains the stack of agents being processed. A meso agent can add its sub-agents to the stack. When an agent has finished processing it leaves the stack.

Control returns to a parent agent when all of its sub-agents have left the stack. The scheduler will run until all agents have left the stack. Clarity has a user interface for interactively populating the scheduler stack and for triggering the scheduler, shown on the left of Figure 13. It allows the processing to be paused and the state of the agents to be inspected, for debugging purposes.

For triggering agents outside this interface, several options have been developed at Ordnance Survey. A single selected agent in the Clarity display window can be triggered. An entire class of agents can also be processed, pushing them onto the scheduler stack one at a time. A user interface is available which lists all the agents in a given class and allows one or more of them to be triggered. The same interface can also be used for viewing/selecting the sub-agents of these agents. This is shown on the right of Figure 13.

6.2 Process Sequences

The 1Spatial implementation of the Agent system has a significant performance overhead and is quite time-consuming to configure and debug. After many tests Ordnance Survey Research have now decided to only use it when it provides obvious benefits. For purely sequential model generalisation and data enhancement processes, it is better to conduct the processing outside of the Agent system.

Clarity does come with a mechanism for running process sequences defined in XML. This has been evaluated at Ordnance Survey, but it has some major flaws. The main problem is that the entire sequence is run in a single edit session, so if something goes wrong all the changes are lost. For long sequences on large datasets, another issue is that Clarity can run out of memory. It would be better to run each process in a separate editing session using a separate instance of Clarity. Sometimes Clarity can even run out of memory when running a single process.

So far generalisation results on large datasets have been achieved by manual intervention during the processing. For example: letting the processing run for a few hours, stopping edit, exiting Clarity to free up memory, restarting Clarity and continuing the processing for a few more hours. It is clear that such an approach would not be not desirable in a production environment.

A recent project investigating an automated process for production was forced to overcome these problems. The solution was to have a master Clarity instance which invokes slave Clarity instances to carry out each process (but not using the Agent system). The master determines if a slave process has completed successfully, then triggers the next process. If something goes wrong, processing is terminated, but it can be restarted from where it failed. Some of the processes were decomposed to handle subsets of objects in each editing session, storing intermediate process information in the dataset. In one example the sequence ran for 30 hours continuously without manual intervention.

Ordnance Survey are now working on adapting this architecture to work in a generic way. The flowline is subdivided into import, data enhancement, generalisation and export processes. Within each of these categories any number of processes can be added. Generalisation processes can be single algorithms, or Agent processes comprising many actions. This system could record the processing history on objects to facilitate propagation of updates to derived databases (Zhou et. al. 2008).

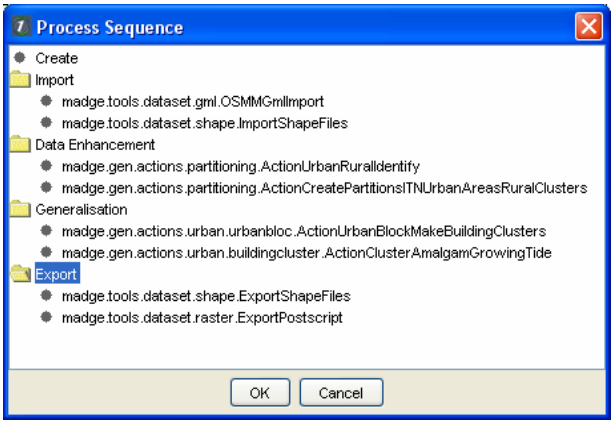

Figure 14. Prototype interface for defining a generalisation flowline.

7 Applications

The generalisation tools were successfully used for creating 1:50 000 scale maps of Glasgow and Birmingham during 2005 and 2006. An sample of the results from Glasgow can be seen on the bottom left of Figure 15 (Revell et. al. 2007).

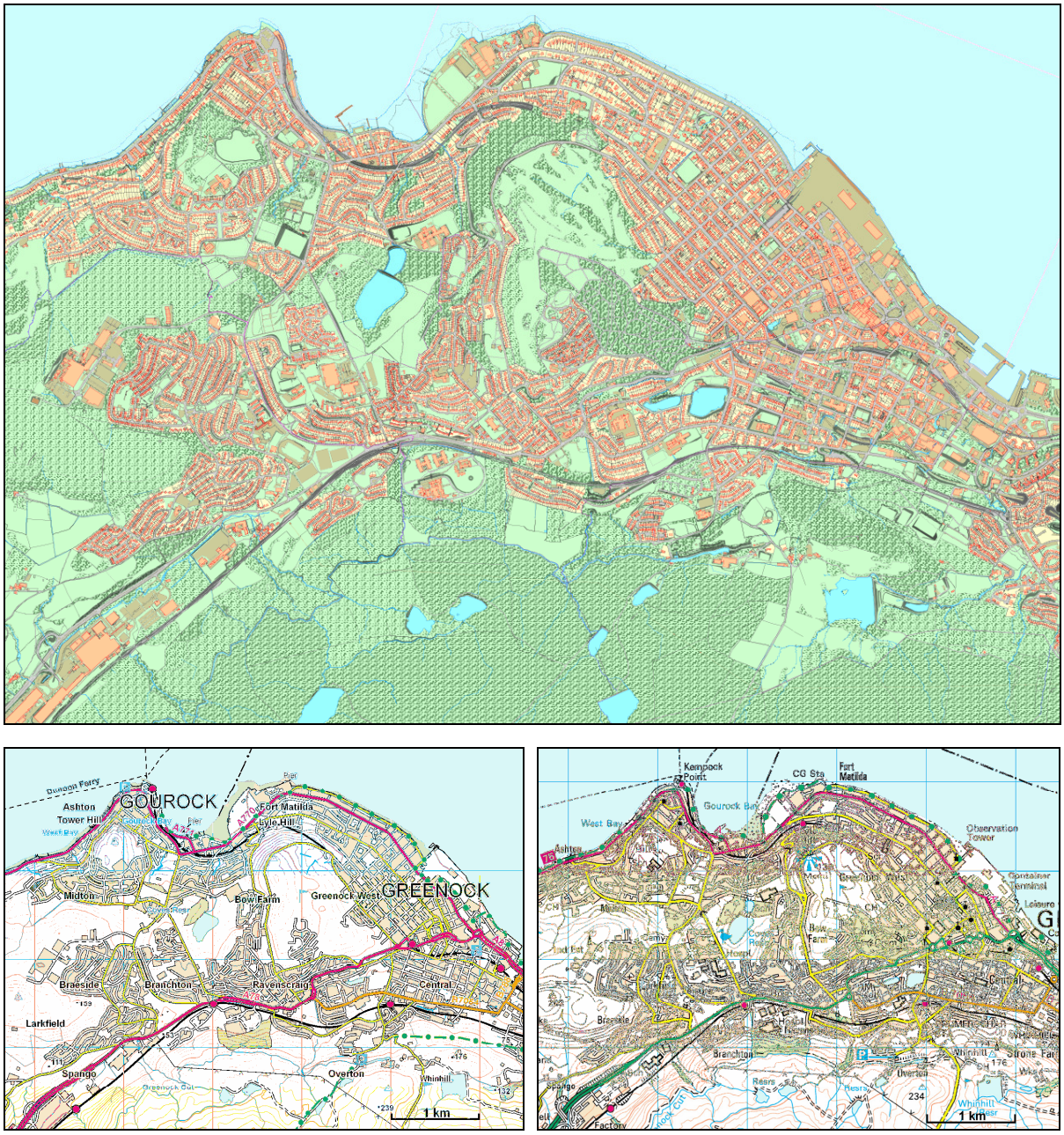

Figure 15. OS MasterMap data (top), compared with 1:50 000 scale mapping, automatically created (left) and manually created (right).

More recently the tools have been used to generate a 1:10 000 scale landcover map from large scale data and displace roads to 1:50 000 scale. The Generalisation team have recently developed production-oriented applications for automatic enhancement of Ordnance Survey large scale data, including complex polygon splitting and creation of path/track networks. This illustrates that Clarity not only suitable for generalisation, but for a wider range of automated spatial data enhancement tasks.

8 Conclusions

A good set of automated generalisation and data enhancement tools has been developed on the Clarity platform at Ordnance Survey. A structure recognition phase was necessary to prepare the data for generalisation, and many of the required tools were found to be absent from Clarity. The generalisation algorithms which are supplied with Clarity have generally not been found to be applicable to Ordnance Survey data. However the platform has proved to be a good base for developing new algorithms, making use of the Agent system, the Gothic topology engine and the geometry-handling tools.

The Generalisation team has the long-term goal of developing a generic generalisation system, suitable for creating any cartographic product from Ordnance Survey large scale data. Therefore any customisations have been made as flexible as possible, so as to be re-usable for other products and by other agencies. The intention is now to attempt to create a new specification product totally automatically, which will get away from mimicking current Ordnance Survey products.

Collaborating with other NMAs through the MAGNET consortium has proved a fruitful means of pursuing Ordnance Survey long-term research goals. It is hoped than more mapping agencies and research institutions will join MAGNET to build upon the strong platform which has already been developed. At the same time it is acknowledged that not everyone has access to Clarity, so the Generalisation Team are investigating making their Clarity algorithms available through the WebGen interface.

Clarity is not easy to use off the shelf, and much of the strong functionality is quite inaccessible. Tools have been developed at Ordnance Survey to make Clarity easier to use, but these really need to be incorporated into the standard Clarity product. It is hoped that 1Spatial will continue to build upon the strengths of Clarity.

References

1SPATIAL (2006), *Gothic reference manual*, part of Clarity user documentation.

1SPATIAL (2007), *1Spatial demonstrates first results from AdV ATKIS Generalisation Project at INTERGEO*. http://www.1spatial.com/news_events/news/news.php?news=226

1SPATIAL (2008), *Radius Clarity – a rules based environment for automated generalisation*, 1Spatial website. http://www.1spatial.com/products/radius_clarity/

AGENT (2000). *Project AGENT, ESPRIT/LTR/24939 documentation*. http://agent.ign.fr/summary.html

BADER, M. (2001), *Energy minimisation methods for feature displacement in map generalisation*, PhD thesis, University of Zurich.

BARRAULT, M., REGNAULD, N., DUCHENE, C., HAIRE, K., BAEIJS, C., DEMAZEAU, Y., HARDY, P., MACKANESS, W., RUAS, A. and WEIBEL, R. (2001), *Integrating multi-agent, object-oriented, and algorithmic techniques for improved automated map generalisation*, proceedings 20th International Cartographic Conference, Beijing, China, Vol. 3, pp 2110-2116.

BRASSEL, K. E. and WEIBEL, R. (1988), *A review and conceptual framework of automated map generalisation*. International Journal of Geographical Information Systems, 2, 229-244.

DOUGLAS, D. and PEUCKER, T. (1973), *Algorithms for the reduction of the number of points required to represent a digitised line or its caricature*, The Canadian Cartographer, Vol. 10, No. 2, pp. 112-122.

ER MAPPER (2008), *ER Mapper – geospatial imagery solutions*, http://www.ermapper.com/

HARDY, P., HAYLES M. and REVELL, P. (2003*), Clarity – a new environment for generalisation using agents, Java, XML and topology*, presented at the 6th ICA Workshop on Generalisation and Multiple Representation, Paris. http://ica.ign.fr/BDpubli/paris2003/papers/hardy_et_al_v1.pdf

HARRIE, L. and REVELL, P. (2007), *Automation of vegetation symbol placement on Ordnance Survey 1:50 000 scale maps*, The Cartographic Journal Vol. 44 No. 3., pp258-267.

HORTON, H. (1945), *Erosional development of streams and their drainage basins*, Bulletin of the Geological Society of America, 56, pp. 275-370.

LASER-SCAN (2002), *Gothic database concepts*, training course manual.

LECORDIX, F., REGNAULD, N., MEYER, M. and FECHIR, A. (2005), *MAGNET Consortium*, presented at the 8th ICA Workshop on Generalisation and Multiple Representation, La Coruña, Spain.
http://aci.ign.fr/Acoruna/Papers/Lecordix_Regnauld_et_al.pdf

LEMARIE, CECILE (2003), *Generalisation Process for Top100: Research in Generalisation Brought to Fruition* presented at the 6th ICA Workshop on Generalisation and Multiple Representation, Paris.
http://www.geo.unizh.ch/ICA/docs/paris2003/papers/lemarie_v0.pdf

MACKANESS, W., PERIKLEOUS, S. and CHAUDHRY, O. (2007), *Representing forested regions at smaller scales: automatic derivation from the very large scale*, proceedings XXIII International Cartographic Conference, Moscow, Russia, 4th to 10th August 2007.

NEUN, M. and BURGHARDT, D. (2005), Web services for an open generalisation research platform, presented at the 8th ICA Workshop on Generalisation and Multiple Representation, La Coruña, Spain.
http://ica.ign.fr/Acoruna/Papers/Neun_Burghardt.pdf

REGNAULD, N. (2005), Spatial structures to support automatic generalisation, proceedings XXII International Cartographic Conference, A Coruña, Spain, 11-16 July 2005.
http://www.ordnancesurvey.co.uk/oswebsite/partnerships/research/publications/docs/2005/034_NICOLASREGNAULD_gen.pdf

REGNAULD, N. (2006), *Improving efficiency for developing automatic generalisation solutions*, The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. 34, Part XXX, presented at the joint ISPRS Workshop on Multiple Representation and Interoperability of Spatial Data. 22nd to 24th February, 2006, Hannover.
http://www.ordnancesurvey.co.uk/oswebsite/partnerships/research/publications/docs/2006/ISPRS_Regnauld_gen.pdf

REGNAULD, N. and Mackaness, W. (2006), *Creating a hydrographic network from a cartographic representation: a case study using Ordnance Survey MasterMap data*, International Journal of Geographical Information Science, 20(6), 611-631.

REGNAULD, N. and REVELL, P. (2007), *Automatic amalgamation of buildings for producing Ordnance Survey 1:50 000 scale maps*, The Cartographic Journal Vol. 44 No. 3, pp239-250.

REGNAULD, N., FECHIR, A., LECORDIX, F. and REJKJAER, D. (2007), NMA's *collaborations on generalisation*, proceedings XXIII International Cartographic Conference, Moscow, Russia, 4th to 10th August 2007.
http://www.ordnancesurvey.co.uk/oswebsite/partnerships/research/publications/docs/2007/ACI_2007_MAGNET_FULL_PAPER_gen.pdf

REVELL, P. (2004), *Building on past achievements: generalising OS MasterMap rural buildings to 1:50 000 scale*, presented at the 7th ICA Workshop on Generalisation and Multiple Representation, Leicester.
http://aci.ign.fr/Leicester/paper/Revell-v2-ICAWorkshop.pdf

REVELL, P. (2005), *Seeing the wood from the trees: Generalising OS MasterMap tree coverage polygons to woodland at 1:50 000 scale*, presented at the 8th ICA Workshop on Generalisation and Multiple Representation, La Coruña, Spain.
http://ica.ign.fr/Acoruna/Papers/Revell.pdf

REVELL, P., REGNAULD, N. and THOM, S. (2005), *Generalising OS MasterMap topographic buildings and ITN road centrelines to 1:50 000 scale using a spatial hierarchy of Agents, triangulation and topology*, proceedings XXII International Cartographic Conference, A Coruña, Spain, 11-16 July 2005.
http://www.ordnancesurvey.co.uk/oswebsite/partnerships/research/publications/docs/2005/043_PATRICKREVELL_gen.pdf

REVELL, P., REGNAULD, N. and THOM, S. (2006), *Generalising and symbolising Ordnance Survey base scale data to create a prototype 1:50 000 scale vector map*, presented at the 9th ICA Workshop on Generalisation and Multiple Representation, Portland, Oregon. http://ica.ign.fr/Portland/paper/ICA2006-Revell.pdf

REVELL, P. (2007a), *Generic tools for generalising Ordnance Survey base scale landcover data*, presented at the 10th ICA Workshop on Generalisation and Multiple Representation, Moscow, Russia.
http://ica.ign.fr/BDpubli/moscow2007/Revell-ICAWorkshop.pdf

REVELL, P. (2007b), *Automated Generalisation and Representation of Ordnance Survey Polygonal Landcover Data at 1:10 000 Scale*, proceedings XXIII International Cartographic Conference, Moscow, Russia, 4th to 10th August 2007.
http://www.ordnancesurvey.co.uk/oswebsite/partnerships/research/publications/docs/2007/ICC_PRR_OSGB_full_paper_final_gen.pdf

REVELL, P., REGNAULD, N. and THOM, S. (2007), *Generalising and symbolising Ordnance Survey base scale data to create a prototype 1:50 000 scale vector map*, The Cartographic Journal Vol. 44 No. 3, pp 251-257.

RUAS, A. (1998), A Method for building displacement in automated map generalisation, International Journal of Geographical Information Science Vol. 12 No. 8, pp 789-803.

SESTER, M. (2005), Optimising approaches for generalisation and data abstraction, International Journal of Geographic Information Science, Vol .19, No. 8-9, pp 871-897.

THOMSON, R. C. and Brooks, R. (2000), *Efficient generalisation and abstraction of network data using perceptual grouping*, proceedings of the 5th International Conference on GeoComputation. http://www.geocomputation.org/2000/GC029/Gc029.htm

VISVALINGAM, M. and WHYATT, J.D. (1993), *Line generalisation by repeated elimination of points*, Cartographic Journal, 30(1), 46-51.

WATSON, P. and SMITH, V. (2004), *Interoperability of Agent-based generalisation with open, geospatial clients*, presented at the 7th ICA Workshop on Generalisation and Multiple Representation, Leicester.
http://ica.ign.fr/Leicester/paper/Smith-v2-ICAWorkshop.pdf

ZHOU, S. and JONES, C.B. (2001), *Multi-scale spatial database and map generalisation*, presented at the 4th ICA Workshop on Generalisation and Multiple Representation, Beijing.
http://www.geo.unizh.ch/ICA/docs/beijing2001/presentations/n22-Zhou.pdf

ZHOU, S. and JONES, C.B. (2005), *Shape-Aware line generalisation with weighted effective area*, Developments in Spatial Data Handling 11th International Symposium on Spatial Data Handling, Editor Peter Fisher, Springer, pp. 369-380.
http://users.cs.cf.ac.uk/C.B.Jones/Zhou_JonesSDH04.pdf

THOM, S. (2005), *A strategy for collapsing OS Integrated Transport Network dual carriageways*, presented at the 8th ICA Workshop on Generalisation and Multiple Representation, La Coruña, Spain. http://ica.ign.fr/Acoruna/Papers/Thom.pdf

THOM, S. (2006), *Conflict identification and representation for roads based on a skeleton*, 12th International Symposium on Spatial Data Handling in Vienna.

THOM, S. (2007), *Resolution of road network conflicts using displacement algorithms orchestrated by geographical Agents*, presented at the 10th ICA Workshop on Generalisation and Multiple Representation, Moscow, Russia.
http://ica.ign.fr/BDpubli/moscow2007/Thom_ICA_Workshop.pdf

ZHOU, S., REGNAULD, N. and ROENSDORF, C. (2008), *Towards a Data Model for Update Propagation in MR-DLM*, to be presented at the 11th ICA Workshop on Generalisation and Multiple Representation, Montpellier, France.