Automatic Resolution of Road Network Conflicts using Displacement Algorithms Orchestrated by Geographical Agents

Stuart Thom

Ordnance Survey, Research and Innovation
Romsey Road, Southampton, UK
stuart.thom@ordnancesurvey.co.uk

Abstract

This paper details the design of a system which automatically applies road section displacements to solve spatial conflicts. The dataset used in the study is the OS MasterMap Integrated Transport Network™ Layer (ITN). The dataset can be partitioned using the major roads as boundaries to give a set of non-overlapping polygons which are known as partition areas. The method to detect and measure conflicts between roads within partition areas (Thom 2006) has been extended to work on groups of road sections. We describe the resolution of conflicts using two commercially available displacement algorithms, Beams (Bader and Barrault 2001) and Least Square Adjustment (Push) (Sester 2000). Also automatic trimming of culs-de-sac is utilized where appropriate. The rationale and method of applying these algorithms using the Agent system, and the use of the improved conflict detection method to assay their success is described. This paper will explain the necessary components of the system, and then the order of applying the processes will be laid out.

Keywords: automated generalization**,** multi-agent system, conflict detection, Delaunay triangulation, skeleton, elastic beams, least square adjustment

1. System Components

The software platform chosen for developing this generalisation prototype is Clarity (Neuffer, Hopewell, and Woodsford 2004) (1Spatial 2007) developed by 1Spatial.

Road sections

ITN is a fully topologically structured road centreline network. The spatial conflicts occur when the symbolization of the roads at the target scale overlap with neighbouring roads. This study does not approach the problem of self overlapping roads; i.e. the ones which zigzag up the side of mountains (since this problem has already been addressed and solutions are available in Clarity). A process has previously been developed to produce the collapsed centre lines of the dual carriageways and traffic islands of an ITN dataset (Thom 2005). The road sections used in this study had been collapsed in this fashion.

Multi-agent system

Clarity was built on the results of a European project called AGENT (Lamy, Ruas, Demazeau, Jackson, Mackaness, and Weibel 1999) which was investigating the use of multi-agent technology to build a generic generalisation systems. Clarity agents provide mechanisms to try different algorithmic solutions and validate the best one. Clarity also includes the Beams algorithm.

Fixed Outer Partition

This is an area, created from Motorways and possibly A-roads, whose boundary can be considered fixed in space. Its constituent Motorways (and A-roads) cannot be displaced when resolving conflicts. All junction nodes on this boundary are also fixed in space, e.g. a road section which joins, at a T-junction, the fixed partition edge cannot be slid along the boundary to resolve its conflicts.

Movable Inner Partitions

These are areas, created from Motorways, A-roads and B-roads, which divide up the fixed outer. They are needed to ease any processing problems when creating the triangulation of the road network. Both the Push algorithm and the conflict detection algorithm use such structures, and splitting the fixed partitions up can speed up these procedures. In addition any triangulation failures have diminished impact affecting solely the area defined by the smaller inner partitions.

Conflict region features

The conflict detection software aims to place a *conflict region* feature on top of any part of the road network where it spots a conflict. Figure 1 shows some examples of conflict regions. The road sections are shown as thick green lines. The conflict information is created from a skeleton (thin red lines) derived from a 'constrained' Delaunay triangulation (thin green lines). One important property of the skeleton is that by definition it's a line that represents the equidistance to the sides of a polygon. Here the two polygon sides are two separate roads.

A *conflict region* describes the extent and the intensity of a conflict. It stores the type of conflict, the closest separation, the threshold distance, and its associated *adjacency lists*; lists of the road sections which are involved. Depending on type there can be up to four lists, to the right, left, start and to the end.

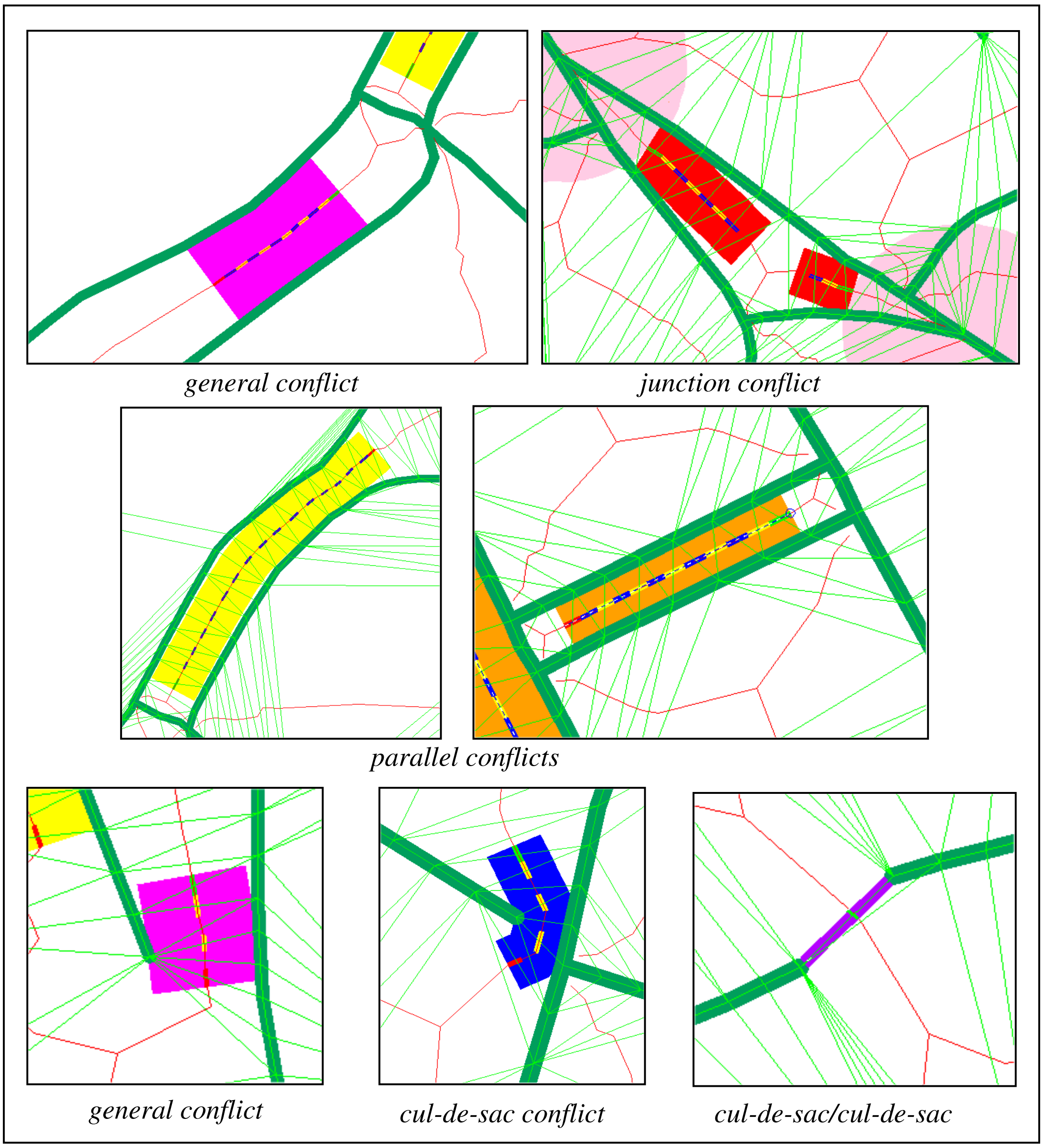

Figure 1 showing types of conflict region

Conflict cluster agent

Using the adjacency lists it is easy to trigger a process where a conflict region's adjacent road sections are searched for their involvement in other conflict regions etc. etc. In this way we can aggregate conflicts producing a list of conflicting road sections which might be best displaced as a group. We will call such a group a *conflict cluster agent*. It may contain sections from more than one inner partition area, allowing displacement algorithms to act to resolve the local conflict scenario as a whole.

Displacement software

Work at the French National Mapping Agency (Lemarie 2003) used the Beams displacement algorithm to resolve conflicts in their network when depicted at 1:100 000 scale, illustrating how Beams can be used for the displacement of line networks. In our study the algorithm will be applied to a group of road sections managed by a conflict cluster agent.

Push is based on an optimization process (Harrie 1999; Højholt 2000; Sester 2000) and operates on ESRI shape files. It has been tailored to be callable from the Clarity system. We write the cluster of sections to a shape file, perform the conflict resolution, read the displaced geometries back in, and apply each to their appropriate road section.

Both Push and Beams have provision for keeping specific nodes fixed. This is important since the end nodes where the cluster is attached to the surrounding network must not move. Both algorithms also allow complete road sections to be fixed, a requirement when resolving conflicts in clusters that include sections from the *fixed outer partition*. In some circumstances either algorithm can return invalid geometries, or geometries which self-intersect or intersect other geometry in the cluster. The returned geometry may overlap sections of the network external to the cluster. Strategies to deal with these occurrences are mentioned in the Section 2.

2. Process methodology

We initially create outer and inner partition areas for the whole dataset. For each and every **outer partition**:

- Create conflict region features for **all its inner partitions**. This involves making a constrained Delaunay triangulation for all the road sections of and within each boundary. From the triangulation we construct a partial skeleton, and from this we create the conflict region features.
- Select all the culs-de-sac conflicts and the culs-de-sac/culs-de-sac conflicts (see Figure 1). The conflict region features store both the closest separation and the threshold distance, so the amounts to prune to achieve the required separation can be calculated. If the necessary prune is greater than or close to the cul-de-sac length we delete the section.
- Our experience with Beams when dealing with conflicting culs-de-sac has proved unsatifactory. Therefore our pruned culs-de-sac will be attributed as non-deformable/rigid. In cases where Beams encounters them later in the process they will not be extended but will retain their new lengths.
- Create the conflict cluster agents. The main restriction, when aggregating the road sections under conflict, is we don't cross the perimeter of the outer partition.
- For each conflict cluster agent we use the Clarity multi-agent system to invoke the displacement algorithms in various combinations until the conflict is best resolved.

    - In order to measure how much conflict resolution has been achieved, we have extended the method of creating conflict region features (Thom 2006) to work on fragments of the network. Figure 2 shows the results for a conflict cluster agent consisting of two road sections. A measure of the clusters conflict is worked out on its single purple *conflict region*. The calculation takes the difference between its threshold and its closest separation and multiplies this by its length. If more *conflict regions* were detected values are added together to get the total measurement for a cluster.

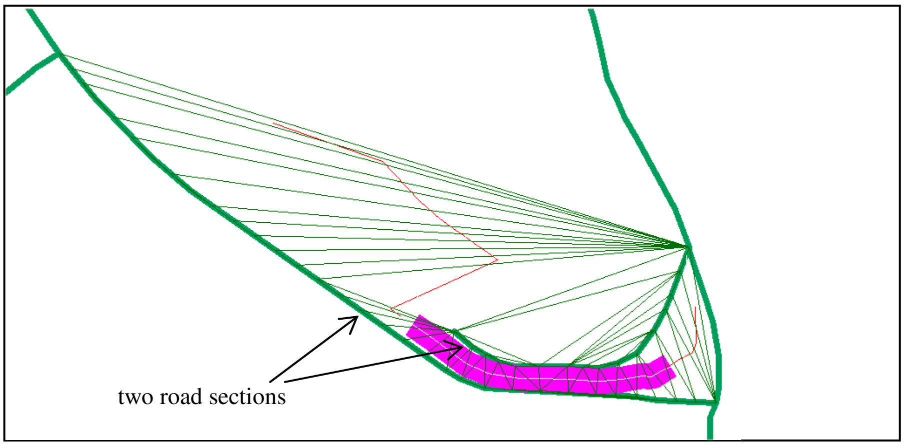

Figure 2 Triangulation on two road sections and resulting conflict region

  - After each invocation of a displacement algorithm, and provided the suggested change produces a significant decrease in the conflict measurement, the system checks the resulting geometries are not self-intersecting and that they don't intersect any of the other sections in this cluster. Spikes and other anomalies are cleaned out.
  - The Clarity system provides methods for interrogating the underlying topology of networks. Using this we check whether the displacements suggested will intersect the network outside the cluster. Providing this is not the case the road sections in this cluster are updated.
  - If any of these validation procedures fail clearly the suggested displacements are abandoned, and the previous best result is returned to.

- After the various resolution strategies have been carried out for all the conflict cluster agents in our **fixed outer partition**, a second set of features called *conflict remnants* is created for **all its inner partitions**. They are matched with the original set of *conflict regions*. The adjacency lists are excellent for match making, since displacement solutions may move road sections considerable distances. The *conflict remnants* allow us to compare individual conflicts before and after the resolution process, and to isolate new secondary conflicts caused by our resolution process.

3. Results

Our dataset contained some 100,000 road links. We processed 65 fixed areas producing 3000 conflict cluster agents. The multi-agent system applied our two algorithms repeatedly until no further significant improvement was produced. The total percentage resolution of conflicts was 88%. The system can be interrogated as to the way in which a cluster's conflict was resolved. Depending on the cluster, it could show that a single invocation of Push was sufficient, or it may require Push followed by Beams followed by two invocations of Push to reach the best result. Table 1 presents an overview of the % contribution of Beams and Push to the 88% resolution obtained using our current agent configuration on this dataset.

	single invocation	1st in sequence	2nd in sequence	3rd in sequence	4th in sequence	5th in sequence
Push	13.4	73.43	5.14	0.88	0.14	0.06
Beams	1.09	3.29	1.65	0.61	0.22	0.06

Table 1 showing the contribution of the two displacement algorithms

It must be stressed that the results are not a comparison of Push and Beams. Push was always applied first to the conflicts and so is bound to produce much higher yields. The results show that to get good results it is necessary to apply the algorithms several times. A single Push only worked 13.4% of the time, using our measurements!

The success of these algorithms depends on parameters such as the stiffness, stretchability/compressibility, and pushability of the road sections we are attempting to displace. Beams additionally has several other internal parameters which it uses in the computation process. No attempt has yet been made to improve the performance of the algorithms by adjusting these values.

Of the 12% of conflict that remained after processing, a third of it was untouched by both Beams and Push. These clusters never got to the 1st step! Perhaps Push produced self intersections which were not successfully repaired **and** Beams failed to produce a significant improvement against our measure. In this case maybe progress could come from the actions suggested in Section 4.

Some new secondary conflicts were created equal to 1.67% of the initial conflict measure.

Figure 2 and 3 show a sample area before and after processing

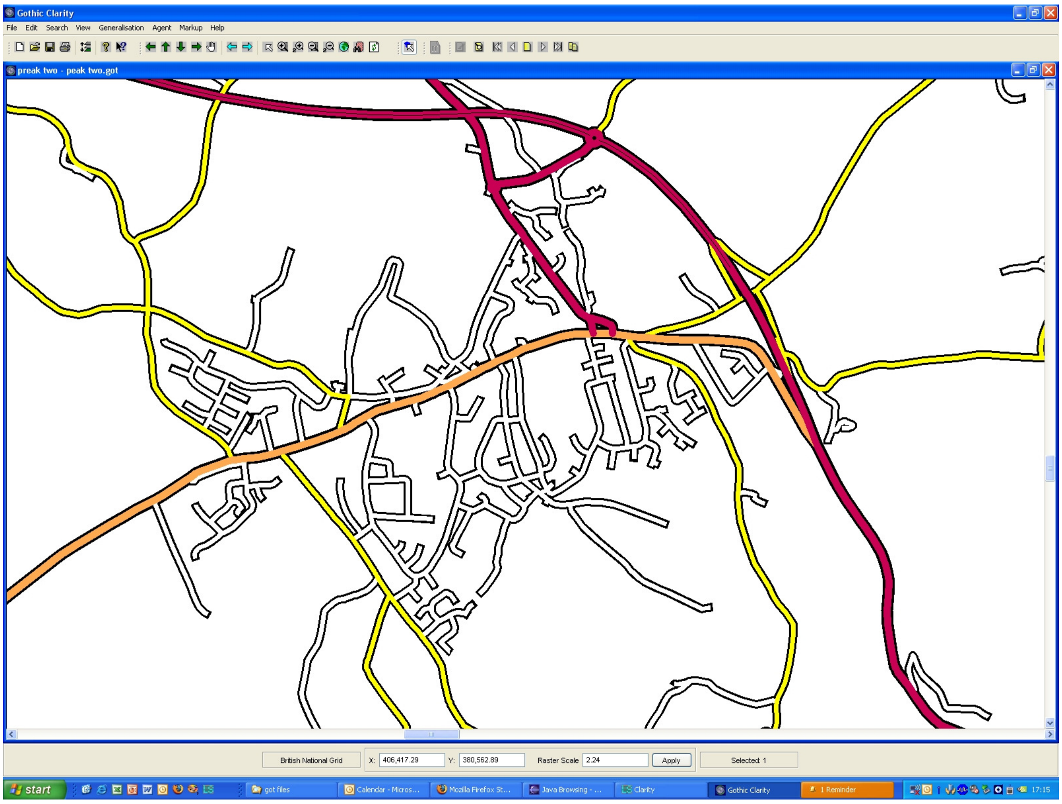

Figure 2 showing an urban area before the conflict resolution process

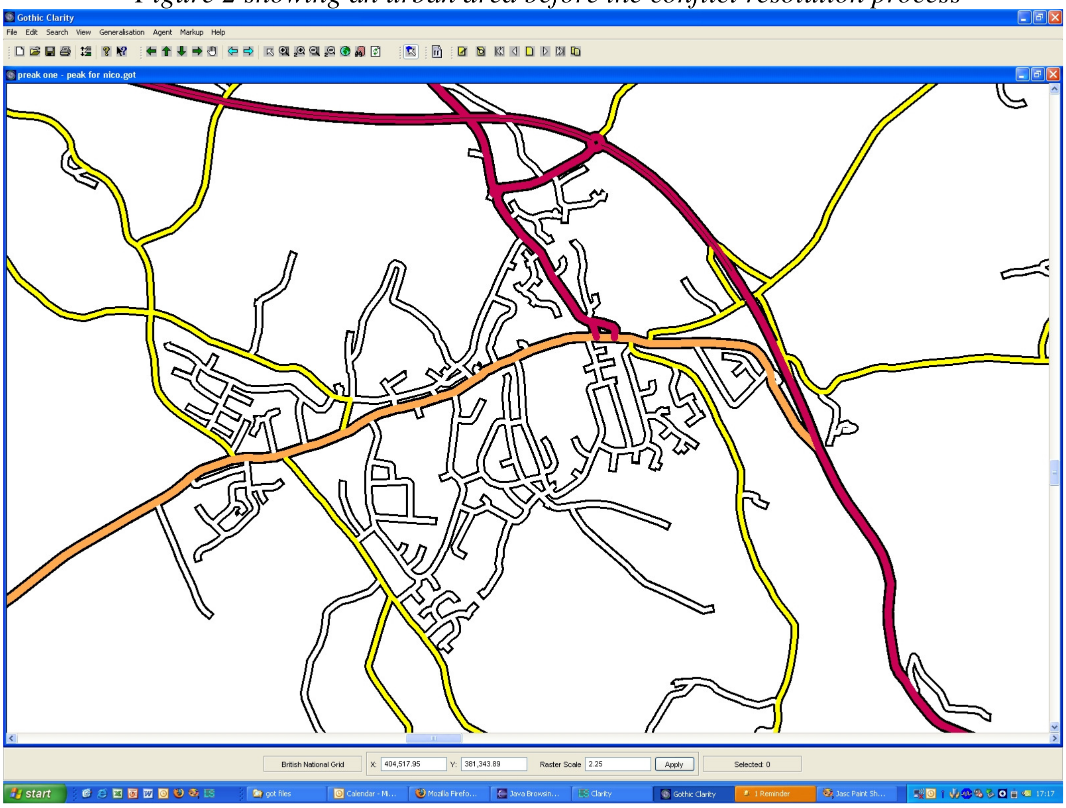

Figure 3 showing the same area after the conflict resolution process

4. Possible strategies

In areas where conflict removal by displacement is unsuccessful, removal of less important road sections will become an essential strategy. Parallel conflicts pose all sort of challenges to the generalization software:

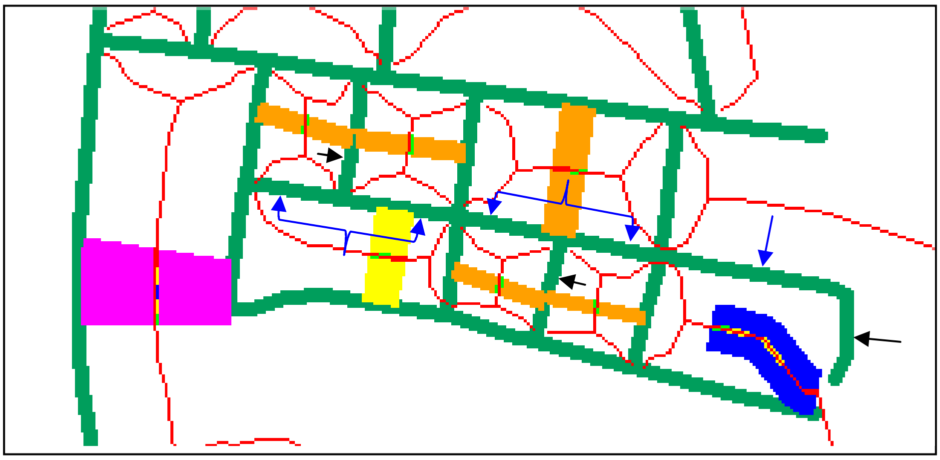

Figure 4 showing various parallel conflict lines

The conflict scenario shown in Figure 4 might be best solved in the following sequence:

- trim cul-de-sac (arrowed)
- remove two (black) arrowed roads – sideways (right/left, not start/end) matches from both sides.
- removal forces regeneration of the triangulation and new conflict regions for this conflict cluster agent.
- remove five blue arrowed roads – side matches on both sides plus alignment for cul-de-sac.
- finally perform displacement on whole cluster.

Conclusion

The first steps towards an automatic method for the application of displacement algorithms to conflicting road section datasets have been put in place. A multi agent system was used to investigate the efficacy of two displacement algorithms, Beams and Push. It is clear that multiple applications are going to be necessary for good resolution of conflicts. Further testing is required before the final components can be decided upon.

Reference List

1. 1Spatial (2007) Radius Clarity - a rule based environment for automated generalisation.

2. Bader, Matthias and Barrault, Mathieu (2001) Cartographic Displacement in Generalization: Introducing Elastic Beams. http://www.geo.unizh.ch/ICA/docs/beijing2001/papers/bader_barraultv1.pdf

3. Harrie, Lars (1999) The constraint method for solving spatial conflicts in cartographic generalization. Cartography and Geographic Information Science. 26: 55-69

4. Højholt, Peter (2000) Solving Space Conflicts in Map Generalization: Using a Finite Element Method. Cartography and GIS. 27: 65-73

5. Lamy, S., Ruas, Anne, Demazeau, Y., Jackson, M., Mackaness, William A., and Weibel, Robert (1999) The Application of Agents in Automated Map Generalisation. Proceedings of the 19th International Cartographic Conference.: 1225-1234

6. Lemarie, C. (2003) Generalisation process for Top100: research in generalisation brought to fruition. http://www.geo.unizh.ch/ICA/docs/paris2003/papers/lemarie_v0.pdf

7. Neuffer, Dieter, Hopewell, Tony, and Woodsford, Peter (2004) Integration of Agent-based Generalisation with Mainstream Technologies and other System Components. http://ica.ign.fr/Leicester/paper/Neuffer-v2-ICAWorkshop.pdf

8. Sester, Monica (2000) Generalisation based on Least Squares Adjustment. International Archives of Photogrammetry and Remote Sensing. 33

9. Thom, Stuart (2005) A Strategy for Collapsing OS Integrated Transport Network™ dual carriageways. http://ica.ign.fr/Acoruna/Papers/Thom.pdf

10. Thom, Stuart (2006) Conflict Identification and Representation for Roads Based on a Skeleton. 12th International Symposium on Spatial Data Handling. 12: 659-680