Ontological modelling of cartographic generalisation

Nicholas Gould[1], Nicolas Regnauld[2], Jianquan Cheng[1]

[1]Manchester Metropolitan University, Chester Street, Manchester
Tel. +4416124747945
nicholas.m.gould@stu.mmu.ac.uk
[2]Ordnance Survey, Adanac Park, Southampton

KEYWORDS: on-demand mapping, generalisation, ontologies, OWL

1. Introduction

Government, both national and local, is making increasing amounts of spatial data freely available (Tinati et al. 2012). For example, the DataGM website provides access to georeferenced data for road traffic accidents, fire and rescue incidents, bus stops, bus routes and traffic signals in Greater Manchester (Trafford Council 2012). However, how can thousands of road accidents be mapped without obscuring the underlying road network (Figure 1) and how can this data be mapped by the non-expert? Tools such as the Google maps API (Batty et al. 2010) provide only a partial solution in that they merely overlay data on base maps. There is no integration of user-supplied data. What is required is cartographic generalisation on-demand. But to automate the map creation process it is necessary to formalise the knowledge required for generalisation – the selection and sequencing of the generalisation operations.

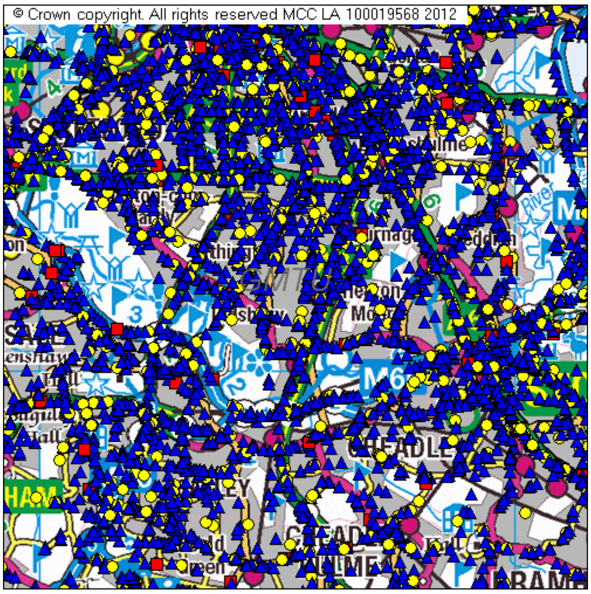

© Crown copyright. All rights reserved MCC LA 100019568 2012

Figure **1**. Road accidents mapped in Greater Manchester

The aim of this project is to determine the *why*, *when* and *how* of generalisation (McMaster and Shea 1992). The need to produce a legible map is the reason *why* we need to generalise the data. The existence of geometric conditions (congestion, imperceptibility) in the mapped data determines the *when*. The existence of these conditions can be determined by using measures such as the density and distribution of features (Stigmar and Harrie 2011). *How* is answered by the use of generalisation operations such as amalgamation and collapse.

Currently cartographic knowledge is embedded in the configuration of sophisticated software applications or in the expertise of their users (Revell et al. 2011). One possible option for representing and sharing that knowledge is to use ontologies (Gruber 1993). We can also *reason* with ontologies and apply them to decision-making. This paper describes an attempt to model the process of generalisation, using ontologies, in an effort to facilitate the automation of map generation.

2. Background

Most research in geographic information ontologies to date has been focussed on real world objects and less on processes such as generalisation (Couclelis 2010). Kulik et al. (2005) employed an ontology in generalisation but used it specifically to enhance a line simplification algorithm rather than to describe the process of generalisation. Touya et al. (2010) developed a domain ontology for the whole generalisation process that included a taxonomy of generalisation operations. However, elements of the ontology, such as the taxonomy of generalisation processes, are platform specific. In other domains ontologies have been used to match students to courses (Kontopoulos et al. 2008) and applicants to jobs (García-Sánchez et al. 2006).

3. Designing the ontology

There is no single correct way of modelling a domain and ontological engineering is necessarily an iterative process (Noy and McGuinness 2001). There are a number of methodologies available to guide the process (Sure et al. 2009) but in our case the “simple” method described by Noy and McGuinness (2001) was employed.

The first stage in designing an ontology is to determine its scope by defining a set of competency questions that the ontology is expected to answer (Noy and McGuinness 2001). In the domain of on-demand mapping the competency questions include:

- Which measures should be applied to which mapped feature collections?
- Which generalisation operations should be applied?
- Which generalisation algorithms should be applied?

Once the generalisation algorithms have been selected, the sequencing of their application has a big impact on the final output. However, sequencing is a complex issue. It can be used to satisfy dependencies or used for optimisation. A sequence can also be *static* (to solve a particular problem) or *dynamic* (reacting to the effects of the last operation). But the ontology was not designed to provide any answers to the sequencing problem which will be left to a *workflow engine*.

Once the scope of the ontology had been determined the next step was to enumerate the key terms in the ontology (Noy and McGuinness 2001). Some of the key terms are highlighted in the following paragraph.

For any given *mapped feature collection* it is necessary to identify the *problem features* (Figure 2). This will be dependent on the *target scale* and will be identified by a number of *measures*. Measures are selected based on the *geometry* of the mapped feature collection. For example, *density* will be a relevant measure of *congestion* for features with any geometry; feature *complexity* will be a relevant measure of *imperceptibility* applicable to line and area geometries only. The measure *algorithm* will return a set of feature collections identifying features with a particular *condition*. The next stage is to identify the generalisation *operations* (amalgamation, collapse etc.) that will remedy the condition.

We could make a direct connection between the geometric conditions and the algorithms that resolve them, but by defining the generic operations we can use inference to make the connection. In addition the operation concept is well-understood and is in common use in the domain (Regnauld and McMaster 2007) where most algorithms are described by the generalisation operations they implement (Li 2007). Since each operation can be implemented by a number of algorithms a further stage, matching operations to algorithms is necessary.

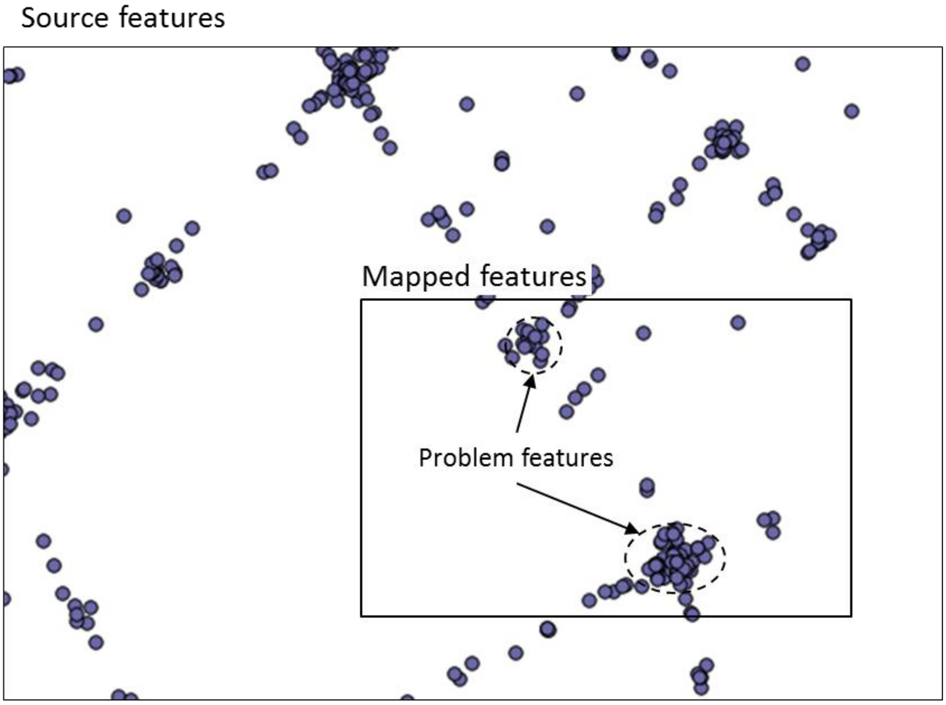
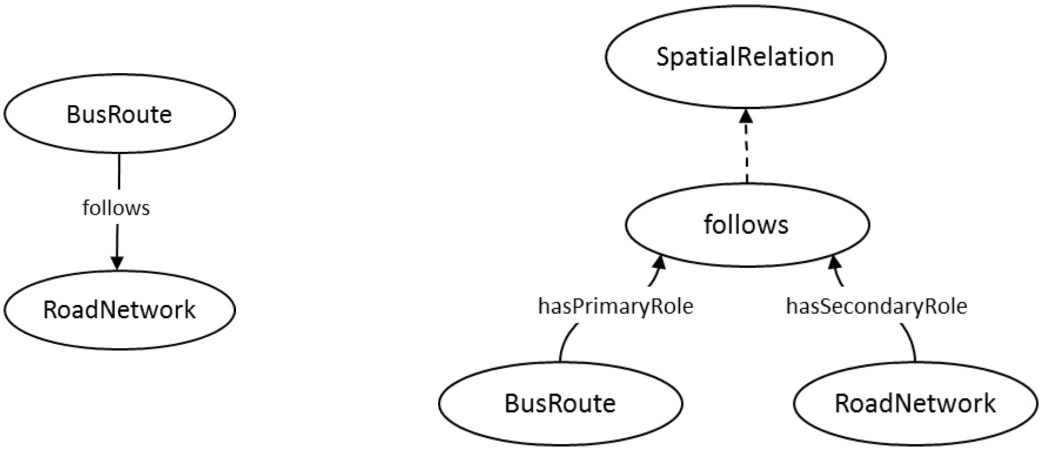


Figure **2**. Identifying problem features within the mapped feature collection

Once the key concepts have been identified then the classes and the class hierarchy can be created. For some ontologies, design seems to stop after the definition of the class hierarchy and are strictly speaking taxonomies rather than ontologies (Gruninger et al. 2007). However, ontologies allow us to define “properties” (relations) between the individuals in classes. As with the definition of classes, the definition of relations is not always an easy task. This can be highlighted if we consider the concept of *spatial relations*. If an on-demand mapping system is to integrate thematic (user) data with base topographic data then it is necessary for the ontology to understand the concept of spatial relations; for example, the relationship between a bus route and the road network. At first it might seem obvious that spatial relations should be modelled as properties of classes (Figure 3a) i.e. we are modelling a relation using a relation. However, in OWL (the Web Ontology Language), properties have limited characteristics. We can define an inverse property (the inverse of *follows* is *isFollowedBy*) or define a property as symmetric (*nextTo* would be a symmetric property) but there is no way of adding attributes to properties. Modelling spatial relations as properties also limits us to modelling binary relations so we cannot model higher order spatial relations, such as the intersection of three different features. One solution is to model spatial relations as classes (Figure 3b) but this makes the reasoning process more complex.



(a) (b)

Figure 3. A spatial relation modelled as a property (a) and a class (b). Classes are represented as ovals and class/sub-class relationships as dotted lines. Properties are represented as solid lines.

The ontology design is a work in progress and will require further refinement as implementation is attempted.

4. Implementing the ontology

The ontology has been developed in the Web Ontology Language (OWL) (Antoniou and van Harmelen 2009) using the Protégé-OWL editor (Horridge 2011). As the ontology was built it could be tested by issuing Description Logic queries from Protégé. For example, to determine the relevant measures for point data the following query (in the Manchester OWL syntax) would be used:

MeasureAlgorithm and *hasGeometry* some **PointGeometry**

The ontology was populated with enough information for a use case that involved mapping road accidents and the underlying road network.

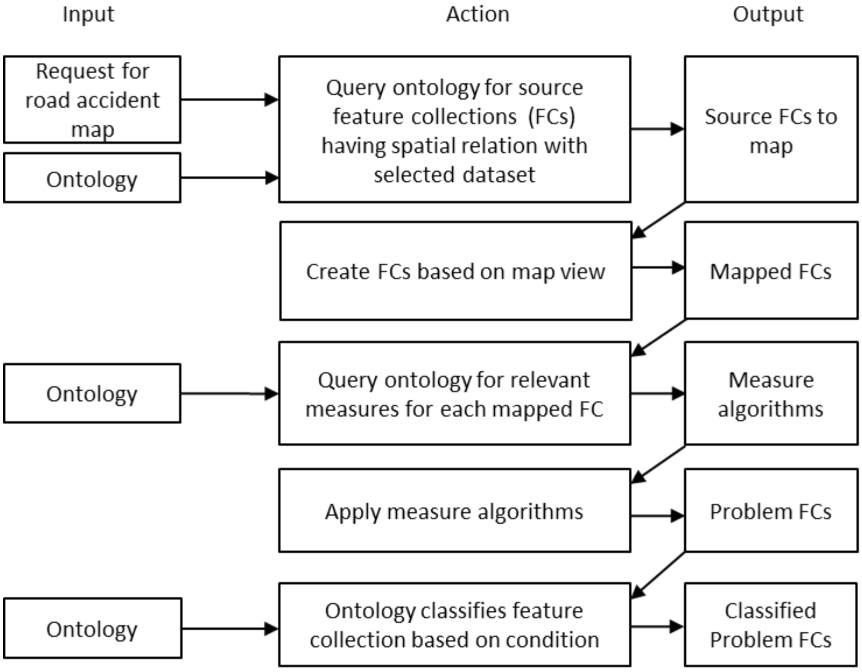


Figure 4. The workflow for identifying which data needs to be generalised.

A Java application is being built using the Geotools library to visualise and manipulate the spatial data, with the OWL API (Horridge and Bechhofer 2011) being used in conjunction with the HermiT reasoner (Shearer et al. 2008) to interact with the ontology. The first stage of the process is summarised in figure 4.

Once the feature collections that require some form of generalisation have been identified, the next stage in the development will be to identify the generalisation operators that will be required to correct the condition. For example, congestion of accident features can be resolved by amalgamating the accidents into clusters. Once that has been achieved the algorithms that will implement the operators will need to be identified.

5. Conclusions

The project has highlighted the difficulties of designing ontologies for decision making based on abstract concepts. The design of the ontology is not yet settled and it was found that it could not be developed in isolation of its implementation. This could be a consequence of how Protégé and the OWL API view an ontology. The *class* is the primary object in the Protégé editor whereas the OWL API views an ontology as a collection of *axioms* (Horridge and Bechhofer 2011). Further work is required to determine the exact extent to which ontologies can be useful in on-demand mapping.

6. Acknowledgements

This project is funded by the Dalton Research Institute at Manchester Metropolitan University and the Ordnance Survey of Great Britain. Thanks to William Mackaness of the University of Edinburgh for his suggestions.

References

Antoniou G and van Harmelen F (2009) Web Ontology Language: OWL. In Staab S and Studer R (eds.) *Handbook on Ontologies*. Berlin/Heidelburg: Springer-Verlag

Batty M, Hudson-Smith A, Milton R and Crooks A (2010) Map mashups, Web 2.0 and the GIS revolution. *Annals of GIS*, 16(1) pp. 1-13.

Couclelis H (2010) Ontologies of geographic information. *International Journal of Geographical Information Science*, 24(12) pp. 1785-1809.

García-Sánchez F, Martínez-Béjar R, Contreras L, Fernández-Breis JT and Castellanos-Nieves D (2006) An ontology-based intelligent system for recruitment. *Expert Systems with Applications*, 31(2) pp. 248-263.

Gruber T (1993) *A translation approach to portable ontology specifications*. Knowledge Acquisition, 5(2) pp. 199-220.

Gruninger M, Bodenreider O, Olken F, Obrst L and Yim P (2008) *Ontology Summit 2007 – Ontology, taxonomy, folksonomy: Understanding the distinctions*. Applied Ontology, 3(3) pp. 191-200.

Horridge M. (2011) A Practical Guide To Building OWL Ontologies Using Protege 4 and CO-ODE Tools Edition 1.3. [Online] [Accessed on 12th November 2012] http://owl.cs.manchester.ac.uk/tutorials/protegeowltutorial/resources/ProtegeOWLTutorialP4_v1_3.pdf

Horridge M and Bechhofer S (2011) *The OWL API: A Java API for OWL ontologies*. Semantic Web, 2(1) pp. 11-21.

Kontopoulos E, Vrakas D, Kokkoras F, Bassiliades N and Vlahavas I (2008) An ontology-based planning system for e-course generation. *Expert Systems with Applications*, 35(1–2) pp. 398-406.

Kulik L, Duckham M and Egenhofer M (2005) Ontology-driven map generalization. *Journal of Visual Languages & Computing*, 16(3) pp. 245-267.

Li Z (2007) *Algorithmic foundation of multi-scale spatial representation*. CRC Press.

McMaster RB and Shea KS (1992) *Generalization in digital cartography*. Washington, D.C.: Association of American Geographers.

Noy NF and McGuinness D. (2001) Ontology Development 101: A Guide to Creating Your First Ontology. Stanford University. [Online] [Accessed on 12th November 2012] http://protege.stanford.edu/publications/ontology_development/ontology101.html

Regnauld N and McMaster RB (2007) A synoptic view of generalisation operators. In Mackaness, WA, Ruas A and Sarjakoski LT (eds.) *Generalisation of Geographic Information*. Amsterdam: Elsevier Science B.V. pp. 37-66.

Revell P, Regnauld N and Bulbrooke B. (2011) OS VectorMap District: automated generalisation, text placement and conflation in support of making public data public. *25th International Cartographic Conference*. Paris, 3-8 July 2011.

Shearer R, Motik B and Horrocks I. (2008) HermiT: A Highly-Efficent OWL Reasoner. In *5th International Workshop on OWL*:

Stigmar H and Harrie L (2011) Evaluation of Analytical Measures of Map Legibility. *The Cartographic Journal*, 48(1) pp. 41-53.

Sure Y, Staab S and Studer R (2009) Ontology Engineering Methodology. In Staab S and Studer R (eds.) *Handbook on Ontologies*. Berlin/Heidelburg: Springer-Verlag

Tinati R, Carr L, Halford S and Pope C. (2012) Exploring the Impact of Adopting Open Data in the UK Government. *Digital Futures 2012*. Aberdeen, 23rd-25th October 2012.

Touya G, Duchêne C and Ruas A. (2010) Collaborative generalisation: formalisation of generalisation knowledge to orchestrate different cartographic generalisation processes. *Proceedings of the 6th International Conference on Geographic Information Science*. Zurich, Switzerland: Springer-Verlag

Trafford Council. (2012) DataGM. [Online] [Accessed on 10th November 2012] http://datagm.org.uk/

Biography

Nicholas Gould is a second year PhD student. Previously he worked in local government developing web applications for transport data.

Nicolas Regnauld has been leading the research team in automated generalisation at Ordnance Survey GB for the last ten years. Before that he had obtained his PhD at IGN France, followed by a four years research fellow position at the University of Edinburgh, working on the European Project (ESPRIT) AGENT.

Jianquan Cheng is a senior lecturer in GIS at the school of science and the environment, MMU. He obtained his PhD in urban growth modelling at University of Utrecht. His research interests are focused on GIS for urban applications using geo-simulation, spatial statistics and spatial decision-making support methods.