An Ontological approach to On-demand Mapping

Nick Gould[1], Omair Chaudhry[2]

[1]Manchester Metropolitan University, Manchester, UK
Email: nicholas.m.gould@stu.mmu.ac.uk

[2] Manchester Metropolitan University, Manchester, UK
Email: o.chaudhry@mmu.ac.uk

1 Introduction

Automatic generalisation for map production has been in use for decades (Li, 2007a). The process is still, however, only semi-automatic in that the expert selects and sequences the required generalisation operators and the algorithms that implement them and provides parameter values. Different techniques can be applied to rural and urban areas at the discretion of the expert, working to a fixed target scale and with familiar feature types (Regnauld and Revell, 2007). But in the case of on-demand mapping the expert will be replaced by a system that will be able to automatically select, sequence and execute map generalisation operations according to user requirements.

The aim of this project is to develop a workflow generation engine that is at the core of an on-demand mapping system (Balley and Regnauld, 2011). The concept of *abstract* tasks as represented by generalisation operators (*Simplification*, *Amalgamation* etc.) and *concrete* tasks as represented by algorithms, that implement operators, will be employed. The separation of abstract and concrete tasks allows for a separation of the definition of the requirements and its implementation (by web services). The process to generate a workflow for on-demand mapping can be broken down as follows[1]:

1. Define abstract tasks - operators
2. Define concrete tasks - algorithms
3. Generate workflow
4. Execute workflow

Before we can automate any task it is necessary to understand it (Georgakopoulos, et al., 1995). We need to formalise the *why*, *when* and *how* of generalisation (McMaster and Shea, 1992). This is particularly important if we want an open, interoperable system. Ontologies allow us to semantically enrich the descriptions of both data and services such that the data and services can become machine-interpretable (Lutz, 2007). This paper focuses on the semantic description of generalisation operations and algorithms using ontologies.

To work effectively, an ontology has to be designed for a specific task (Noy and McGuinness, 2001). Section 3 describes the development of an ontology for automatically selecting generalisation operators. Section 4 describes an ontology for the automatic selection of algorithms to implement the chosen operators. Possible options for deploying the ontologies are discussed in section 5 along with some conclusions. The next section discusses previous work in geospatial ontologies and in on-demand mapping.

[1] The collection and interpretation of user requirements is beyond the scope of this project.

2 Related work

One possible solution to on-demand mapping is to avoid the dynamic generalisation of data and use a Multi Resolution DataBase (MRDB) (Dunkars, 2004). However, if we define on-demand mapping as generalisation according to user requirements, and potentially integrating user-supplied data, then the MRDB approach is not applicable. Bernier and Bédard (2007) describe a hybrid approach – if the data can be generalised quickly and without human intervention then it should be – otherwise the data should be extracted from the MRDB.

If an on-demand system is to integrate user-supplied data in an ad-hoc fashion then automatic, on-demand, generalisation is required. However, if the process is to be completely automated then we first need to formalise the knowledge required to produce a generalised map (Touya et al, 2010). Generalisation is achieved by applying one or more transformations or *operators* (Sarjakoski, 2007). However, following a series of interviews with cartographers, Rieger and Coulson (1992) concluded that there was no consensus over the description of these operators; cartographers had different definitions of the same term and different terms for the same definition. Rieger and Coulson were attempting to elicit *declarative* knowledge about the procedures as opposed to *procedural* knowledge, which describes how the task is carried out. Declarative knowledge, that knowledge contained in declarations about the world, can be extended by reasoning processes that derive additional knowledge (Genesereth and Nilsson, 1998). Can such a method be applied to generalisation?

There have been a number of attempts to classify and describe generalisation operators (Foerster, et al., 2007a; McMaster and Shea, 1992; Roth, et al., 2011) but the problems highlighted by Rieger and Coulson (1992) remain. As well as differences between the proposed categories of operators there are also differences in naming (*Aggregation* or *Combine?*) and in granularity; McMaster and Shea (1992) define *Smoothing*, *Enhancement* and *Exaggeration* where Foerster et al. (2007a) simply define *Enhancement*. There is also disagreement as to what functions can be regarded as generalisation operators. For example, is *Symbolisation* a generalisation operator (McMaster and Shea, 1992; Roth et al., 2011) or a pre-processing step (Foerster et al., 2007a)?

The use of different operator taxonomies in closed systems does not matter, but, if we are to develop an interoperable on-demand system, an agreed taxonomy *and* the semantic description of the operators is required. This is because we cannot simply ask for a web service that performs *Smoothing*, say, since that operation can be performed by a number of different algorithms (Gaussian, Cubic Spline, Fourier transform etc.), often with different results. Similarly, some operators apply to different geometry types and will need to be implemented by different algorithms. Likewise some algorithms specialise in different feature types e.g. buildings (Guercke and Sester, 2011). Thus these details need to be formally defined so that automatic selection and execution is possible by the on-demand system.

Li's study (2007b) of generalisation *algorithms* (rather than operators) provides a possible framework for the semantic description of the generalisation process. He focuses on algorithms and groups them by geometry and by what function they perform; point reduction of areas, for example.

At some stage in the development of an on-demand mapping system there will be a need for Knowledge Acquisition (Kilpeläinen, 2000; Mustiere, 2005; Rieger and Coulson, 1993) but first it is necessary to define the type of knowledge that needs to be acquired and how it is to be encapsulated. The dominant methods for encapsulating cartographic knowledge are *rules* and *constraints*. The rule-based approach involves defining a set of condition-action pairs that will solve particular problems (Sarjakoski, 2007). Rules encapsulate procedural knowledge. The disadvantage of this approach is that a rule has to exist for every eventuality which means a large number of rules need to be defined for a viable system (Harrie and Weibel, 2007). Unlike rules, constraints do not prescribe how a problem should be solved only the condition that should be maintained (Neun et al., 2009).

Formalisation of knowledge can lead to the discovery of new knowledge as long as appropriate formalisation tools are available (Kilpeläinen, 2000). One such tool is the ontology - the explicit specification of the objects, concepts and the relationships in a body of knowledge concerning a particular subject or domain (Gruber, 1993). Ontologies have the advantage of allowing the sharing and reuse of formalised knowledge (Gruber, 1993). The semantic description of geospatial operations, and the web services that implement them, using ontologies to allow for automatic selection is not new (Klusch et al., 2005; Lutz, 2007; Lemmens, et al., 2007) but there has been little focus on the particular problems of generalisation. Touya et al. (2011) have made progress on a generalisation ontology but not specifically for on-demand mapping.

The next section describes the process for developing a generalisation operator ontology.

3 Developing the operator ontology

There is no single, ideal, methodology for designing an ontology (Noy and McGuinness, 2001). The authors' first attempt to develop a generalisation ontology for on-demand mapping involved attempting to capture, in one-step, all the knowledge that could be used to describe the generalisation process. This led to a large, cumbersome, and ultimately unusable ontology. An alternative approach was taken, that involved defining an ontology for a specific purpose.

The purpose of the *operator* ontology is to describe the properties, behaviours and relations of generalisation operators in such a way that they can be selected automatically. The ontology will be designed by reference to a road accident use case (Figure 1). The model will then be tested against further use cases such as the cycle route planner described by Balley and Regnauld (2011). The requirement of the user is to view the road accidents at a detailed level, where no generalisation is required – showing the road network as polygons and individual accidents (Figure 1a) – and at a small, city-wide scale.

a b c

Figure 1 Road accident use case

The aim of the system is to produce a map that *maintains legibility* as the scale is reduced (*Why* generalise). We can decide when to generalise by describing a number of geometric conditions (McMaster and Shea, 1992). For example, the road network which is described using an area geometry (Figure 1a) becomes *congested* at a smaller scale (Figure 1b) and also suffers from *imperceptibility* as the lines that define the road boundaries become too close. The accident dataset at a smaller scale (Figure 1c, shown separately) suffers from *congestion*, *coalescence* and *overlap*. We also have to define a number of measures, such as feature density, to evaluate when a condition has been reached (Stigmar and Harrie, 2011). We can then say that generalisation is required *when* a particular geometric condition occurs. The condition is resolved by one or more operators (*How* to generalise).

Rather than simply present the completed ontology we have described below the decisions and steps taken to build the ontology. This will facilitate criticism of the resultant ontology and help inform further development. This was thought to be particularly important for ontologies that describe a process rather than a set of tangible objects.

The first version of the ontology can be seen in Figure 2. The labelled solid lines represent object properties and the unlabelled dotted lines represent "is-a" sub-class relationships.

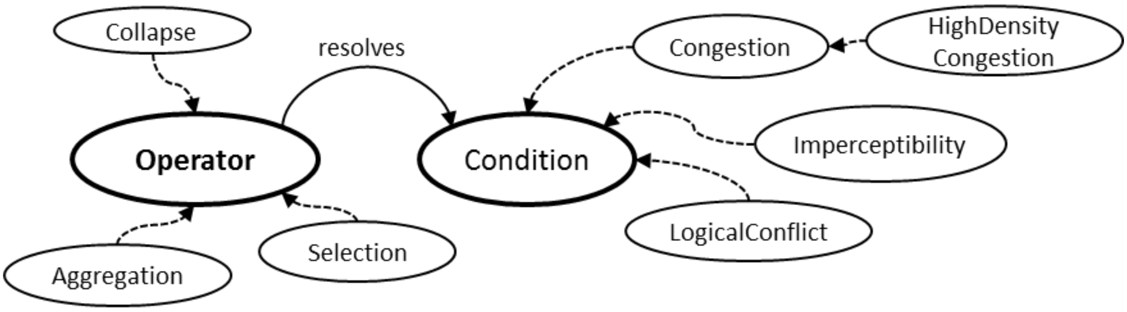

Figure 2 Operator ontology - version 1

The *LogicalConflict* condition is a renaming of the *Conflict* condition defined by McMaster and Shea (1992). Such a condition may occur, for example, when a number of accidents are displayed but the road they lie on has been eliminated for some reason.

Operators can be added to the ontology and linked to one or more conditions (e.g. *Collapse* resolves *HighDensityCongestion*). The measure for *HighDensityCongestion* can be modelled by creating a data property *hasDensity* and adding it to the *HighDensityCongestion* condition with a threshold value. This will need refinement since we will likely have different measures for the congestion of different geometries. The ontology was implemented in Protégé (Horridge, 2011) which allows for the querying of an ontology. So the query[2]:

Operator and *resolves* some **HighDensityCongestion**

might return a number of operators. However, not all operators work on the same geometry types. For example, *Amalgamation* applies to area features and not point features; *Collapse* can apply to areas and lines but not points. By introducing a geometry class and linking specific operators to specific geometry classes, we can reduce the number of operators applicable to a given situation, thus facilitating the automatic selection of an operator. The refined version of the ontology can be seen in Figure 3.

[2] Using the Manchester OWL syntax employed by Protégé. The query can be seen as the consequences of user requirements.

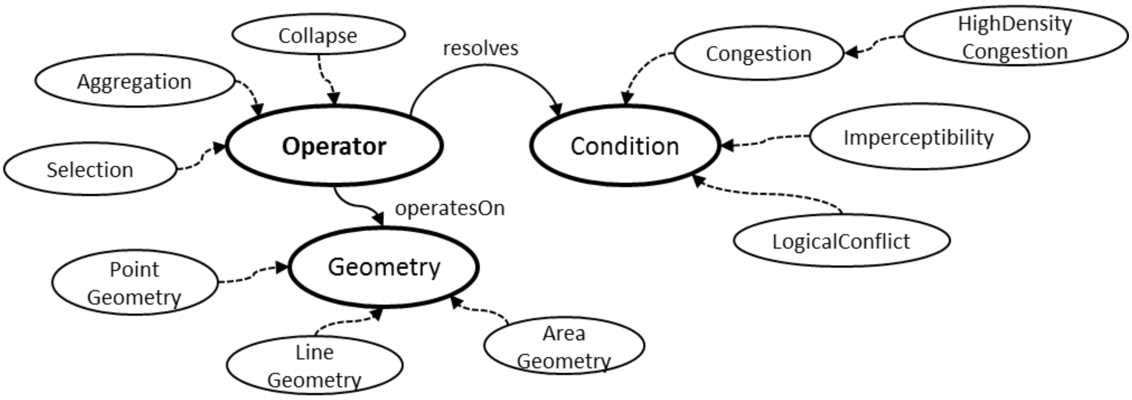

Figure 3 Operator ontology - version 2 (some classes and relations omitted for clarity)

The query can be refined:

Operator and *resolves* some **HighDensityCongestion** and *operatesOn* some **PointGeometry**

Only the operators *Aggregation* and *Selection* would be returned since they are the only two operators that were defined as resolving congestion specifically in point features. The ontology can be further refined when we consider the *Selection* operator in more detail. *Selection* can be used in our use case to reduce congestion by only selecting the most serious road accidents or the most important roads. However, for *Selection* to work the dataset needs an attribute that can be used to rank its features. The ontology therefore needs a concept of a dataset, in particular a ranked dataset, and the concept of an operator transforming a dataset (Figure 4).

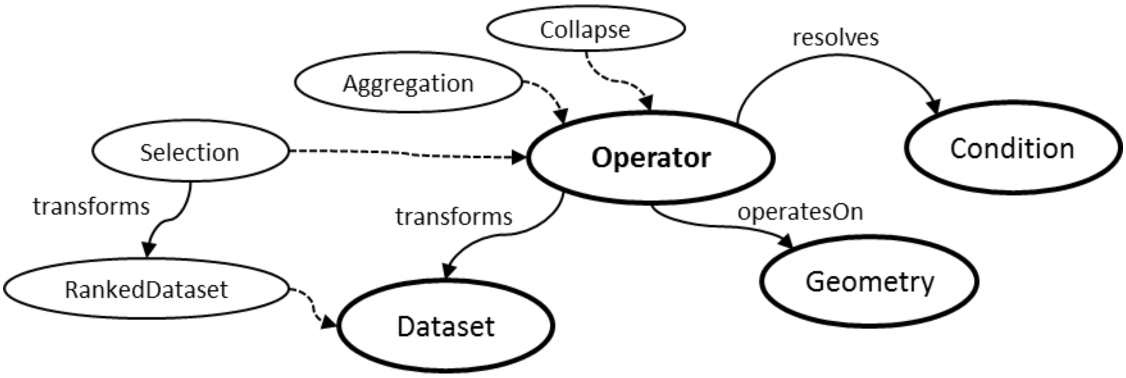

Figure 4 Operator ontology - version 3 (some sub-classes omitted for clarity)

It can be seen that there are a number of relations and classes that could have been defined but were not; for example the relationship between a dataset and its geometry or the possible sub-classes of a dataset (Road dataset, Accident dataset, for example). However, the ontology has been designed on the principle of defining only that which is necessary to fulfil the defined aim (Noy and McGuinness, 2001).

A number of measures were defined for the two datasets in the use case. For point data (the accidents) a density measure of congestion was utilised, based on the number of points per unit map area (pixels). For polygon data (the road network) two measures were defined; an average polygon width (in pixels) as a measure of imperceptibility, on the understanding that if the road section is too narrow then it will be difficult for the viewer to distinguish between opposite sides of the road section. The second measure was for congestion and uses a total feature area per unit map area. The measures were implemented using the Geotools JAVA library and tested on sample data from Greater Manchester. Arbitrary threshold values for the measures were defined and used to indicate whether generalisation was required. The aim is that once the conditions have been identified then the ontology can be queried to determine the appropriate operators to resolve the conditions. This application of measures to trigger generalisation requires further refinement. For the point density measure the effect of symbol size was ignored and no account was made for the spatial distribution of features in either dataset. In addition, each dataset was considered in isolation. It is unlikely that there is a single measure for a condition that is appropriate in all cases and a combination of measures might need to be applied (Mustiere, 2005; Stigmar and Harrie, 2011).

The ontology itself is not complete and there are some unanswered questions. For example, should the operator ontology include the concept of *feature type*? Also, *Amalgamation* may be identified as a suitable operator for congested area features, which may be appropriate for buildings but not for roads or a river network.

The ontology also lacks the concept of precedence. If a query returns two operators that meet the conditions then does this mean that both operators should be applied to the dataset? If so, then in what order? If we apply the first operator and the condition persists do we try the second operator or repeat the first but with a different parameter value? Such a question may lie outside the remit of the ontology and be the responsibility of a Problem Solving Method (PSM) (Gómez Pérez and Benjamins, 1999), which is required to manage the process of constructing and asking the queries and then acting on the results. For example, an agent-based or other optimisation method may be used to define the ideal sequence of proposed algorithms.

Kilpeläinen (2000) refers to the knowledge that is used to select the right generalisation operator for the task as *procedural* knowledge. For the ontological approach to be effective, rather than having to explicitly state the procedural knowledge in the form “operator X resolves condition Y”, the procedural knowledge could be derived from the declarative knowledge by reasoning (Genesereth and Nilsson, 1998). In effect, the “operator resolves condition” relation needs to be made redundant by describing both conditions and operators in such a way that we can derive the relation by query. This requires a more explicit statement of what the operators do and what the conditions are.

The next stage is to develop an ontology that will help select algorithms to implement the selected operators. A separate algorithm is required since algorithms have a different set of properties from operators and an operator can be implemented by a number of algorithms.

4 Developing the algorithm ontology

Before developing the algorithm ontology a survey of generalisation algorithms was done with the intention of informing the design process by highlighting the attributes and behaviours of generalisation algorithms. Algorithms for point aggregation, line *Simplification*, line *Smoothing*, and building *Amalgamation*[3] were examined and common attributes documented (Gould, 2012).

In addition to the operators they implemented and the geometry types they applied to, algorithms varied by *feature type* - some algorithms were specific for roads or

[3] Although building amalgamation was not necessary for the use case, it was included in the survey because of the large number of building amalgamation algorithms.

buildings, for example - and by *terrain* - some algorithms were targeted at mountain roads or rural buildings. Algorithm *parameters* provide further variety - algorithms performing the same task, e.g. line simplification may have different parameters. *Scale* also provides an additional layer of complexity; some algorithms are designed for specific source and target scales. This explains why we need a separate algorithm ontology.

A similar incremental approach to the operator ontology design was used to answer the question: how do we describe an algorithm so that it can be automatically selected? An initial version of the ontology can be seen in Figure 5.

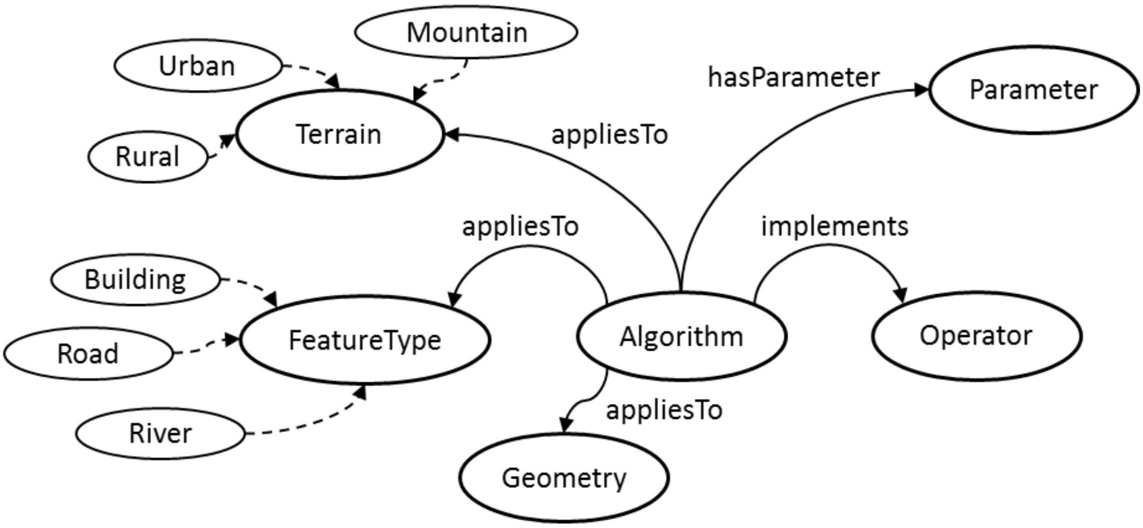

Figure 5 Algorithm ontology (some sub-classes omitted for clarity)

An example query, that aims to find an algorithm for road smoothing might be:

Algorithm and *implements* some **Smoothing** and *appliesTo* some **RoadFeatureType**

In practice an algorithm can be regarded as an abstraction for a *web service*. It is not practical, or necessary, to have a model where we search for a service that implements a particular algorithm. It is unlikely that any service could advertise itself only by the algorithm it implements.

As with the operator ontology, the algorithm ontology requires further refinement. Algorithms that implement multiple operators, such as line simplification *and* smoothing, need to be modelled. Further consideration of which concept should sit within which ontology may be necessary. For example, should the ‘terrain’ concept sit within the operator ontology?

5 Conclusions and further work

We believe that although there are still questions to be answered, the ontological approach to on-demand mapping merits further investigation. But, to what extent can we use ontological reasoning to develop a workflow for on-demand mapping? Is the ontological approach merely a stepping stone to help inform another approach or is it an end in itself? How could the ontologies be applied?

The standard for implementing geospatial web services is the OGC’s Web Processing Service (WPS) protocol. However, the protocol does not provide for *semantic interoperability* (Janowicz et al., 2010); there is no method of adding machine readable descriptions to a service. One possible solution that will be investigated further is the Semantic Enablement Layer (Janowicz et al., 2010) where a *Web Ontology Service* injects semantics into both data and processing services. A *Web Reasoning Service* can then be used to match a processing service to a dataset. Previous work on the generation of workflows for on-demand mapping from a set of tasks and task precedencies (Gould and Chaudhry, 2012) could be employed to generate valid workflows from the output of a Semantic Enablement Layer. The Web Ontology Service could be employed to maintain a shared set of on-demand mapping ontologies.

One major obstacle yet to be resolved is the problem of how to *automatically* provide parameter values to the selected services especially since any two algorithms performing the same generalisation operation may have different parameters. Even if the two algorithms had parameters with a common name such as *minimum distance*, their concept of what a minimum distance means may differ. One possible approach would be to define a common set of parameters to be used by all services, extending the work on line simplification ratios of Foerster et al. (2007b). It would then be the responsibility of any service implementing an algorithm to translate the common parameter values to local parameter values. Values for the common parameters could be derived from the geometric condition measures described in the Operator ontology. For example, a high value for a condition could lead to a correspondingly high value for a parameter for the selected algorithm.

Acknowledgements

This project is funded by the Dalton Research Institute at Manchester Metropolitan University and the Ordnance Survey of Great Britain. Thanks to Nico Regnauld and Sandrine Balley at the Ordnance Survey and Martin Stanton at MMU for their guidance and to the anonymous reviewers for their suggestions.

References

BALLEY, S. and REGNAULD, N., 2011. *Models and standards for on-demand mapping*. 25th International Cartographic Conference. Paris.

BERNIER, E. and BEDARD, Y., 2007. A Data Warehouse Strategy for on-Demand Multiscale Mapping. In: W. A. MACKANESS, et al. eds., *Generalisation of Geographic Information*, Amsterdam: Elsevier Science B.V., pp. 177-198.

DUNKARS, M., 2004. *Automated Generalisation In A Multiple Representation Database*. 12th Int. Conf. on Geoinformatics - Geospatial Information Research: Bridging the Pacific and Atlantic. University of Gävle, Sweden.

FOERSTER, T., et al., 2007a. *Towards a formal classification of generalization operators*. International Cartographic Conference 2007. Moscow.

FOERSTER, T., STOTER, J., KÖBBEN, B. and VAN OOSTEROM, P. 2007b. *A Generic Approach to Simplification of Geodata for Mobile Applications*. 10th AGILE International Conference on Geographic Information Science. Aalborg University, Denmark.

GEORGAKOPOULOS, D., et al., 1995. An overview of workflow management: From process modeling to workflow automation infrastructure. *Distributed and Parallel Databases*, 3 (2) pp. 119-153.

GENESERETH, M. R., and NILSSON, N. J., 1987. *Logical Foundations of Artificial Intelligence*. Morgan Kaufmann Publishers.

GUERCKE, R. and SESTER, M., 2011. *Building Footprint Simplification Based on Hough Transform and Least Squares Adjustment*. ICA Workshop on generalisation. Paris, France.

GOMEZ-PEREZ, A., BENJAMINS, R. 1999. Overview of Knowledge Sharing and Reuse Components. Ontologies and Problem Solving Methods. In: *Proceedings of the IJCAI-99 Workshop on Ontologies and Problem-Solving Methods (KRR5)*, Stockholm.

GOULD, N., 2012. *Semantic description of generalisation web services for on-demand mapping.* 1st AGILE PhD School, Wernigerode, Germany, Shaker Verlag, 13-14th March 2012.

GOULD, N. and CHAUDHRY, O. Z., 2012. *Generation and validation of workflows for on-demand mapping*. Geoprocessing 2012. Valencia, Spain.

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

HARRIE, L. and WEIBEL, R. 2007. Modelling the Overall Process of Generalisation. In: MACKANESS, W. A., RUAS, A. and SARJAKOSKI, L. T. (eds.) *Generalisation of Geographic Information*, Amsterdam: Elsevier Science B.V. pp. 67-87.

HORRIDGE, M., 2011. *A Practical Guide to Building OWL Ontologies Using Protege 4 and CO-ODE Tools Edition 1.3*. [online] http://owl.cs.manchester.ac.uk/tutorials/protegeowltutorial/ Accessed 11th May 2012.

JANOWICZ, K., et al., 2010. Semantic Enablement for Spatial Data Infrastructures. *Transactions in GIS*, 14 (2) p. 111.

KILPELÄINEN, T. 2000. Knowledge Acquisition for Generalization Rules. *Cartography and Geographic Information Science*, 27(1) pp. 41-50.

KLUSCH, M., et al., 2005. Semantic Web service composition planning with OWLS-Xplan. *AAAI Fall Symposium - Technical Report*, FS-05-01 pp. 55-62.

LEMMENS, R., et al., 2007. Enhancing Geo-Service Chaining through Deep Service Descriptions. *Transactions in GIS*, 11 (6) pp. 849-871.

LI, Z., 2007a. Digital map generalization at the age of enlightenment: A review of the first forty years. *The Cartographic Journal*, 44 (1) pp. 80-93.

LI, Z., 2007b. *Algorithmic foundation of multi-scale spatial representation*. CRC Press.

LUTZ, M., 2007. Ontology-Based Descriptions for Semantic Discovery and Composition of Geoprocessing Services. *GeoInformatica*, 11 (1) pp. 1-36.

MCMASTER, R. B. and SHEA, K. S., 1992. *Generalization in digital cartography*. Washington, D.C.: Association of American Geographers.

MUSTIERE, S., 2005. Cartographic generalization of roads in a local and adaptive approach: A knowledge acquisition problem. *International Journal of Geographical Information Science*, 19 (8-9) pp. 937-955.

NEUN, M., BURGHARDT, D. and WEIBEL, R. 2009. Automated processing for map generalization using web services. *GeoInformatica*, 13 (4) pp. 425-452.

NOY, N. F. and MCGUINNESS, D., 2001. *Ontology Development 101: A Guide to Creating Your First Ontology*. Stanford University.

REGNAULD, N. and REVELL, P., 2007. Automatic Amalgamation of Buildings for Producing Ordnance Survey 1:50 000 Scale Maps. *The Cartographic Journal*, 44 (3) pp. 239-250.

RIEGER, M.K. and COULSON, M.R.C., 1992. Consensus or Confusion: Cartographers' Knowledge of Generalization. *Cartographica*, 30 (2 & 3) pp. 69-80.

ROTH, R., et al., 2011. A typology of operators for maintaining legible map designs at multiple scales. *Cartographic Perspectives*, 68.

SARJAKOSKI, L. T. 2007. Conceptual Models of Generalisation and Multiple Representation. In: MACKANESS, W. A., RUAS, A. and SARJAKOSKI, L. T. (eds.) *Generalisation of Geographic Information*, Amsterdam: Elsevier Science B.V., pp. 11-35.

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

TOUYA, G., et al., 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.