Rule based quality control for automated generalisation and conflation

Nicolas Regnauld,[1] and Derek Howland[2]

1. 1Spatial
2. Ordnance Survey GB

Abstract: This paper shows through a couple of use cases the importance of quality management in automated systems. Managing quality of the data is the first thing that comes to mind, but quality of the software is also of prime importance to ensure the system can be maintained and can evolve throughout its life time. The first use case presented is about the new production system used at Ordnance Survey GB to derive a range of products from higher resolution data. The second reports on a study conducted at 1Spatial for deriving a thematic map using data from multiple sources and conflated automatically. In both cases, the flexibility of the rules authoring system has been key to define and deploy the right quality rules for the job.

Keywords: Quality control, automatic generalisation, data conflation, production system, on demand mapping

1. Introduction

In organisations such as National Mapping Agencies, responsible for maintaining consistent, up to date map coverage for a given territory, often at various scales, the strategy consists more and more in collecting and maintaining topographic data in a single high resolution database, used to derive various products, as efficiently as possible. The cost of deriving maps at smaller scales manually from it is prohibitively expensive, so automatic generalisation is required. Beyond the creation of these traditional maps, NMAs make available more and more of their data for users to use in their own applications. At the same time, data about a wide range of topics are also becoming available, often due to pressure from governments to provide access to public data. This brings the prospect of creating thematic maps on demand.

The purpose of this paper is to look at the importance and the role of quality management throughout systems to process geospatial data to derive maps. The first part of the paper provides a general introduction to the need for geospatial data quality management in production systems, it also presents the approach used at 1Spatial for authoring rules. The second part analyses how data quality is managed at Ordnance Survey GB, in their new highly automated production system. The paper then moves on to describe a study conducted at 1Spatial to derive a thematic map after conflating data from multiple sources. The focus is also on the importance of quality management, and the additional challenges introduced by data integration. The paper concludes with a summary of the key challenges related to quality management in production systems to support todays map production systems and tomorrows on demand services.

2. Assessing geospatial data quality

A company offering geospatial datasets or map products needs to ensure that the final product is fit for purpose. The production system designed and built to create and maintain the product needs to be used during its full lifetime. Three key aspects need to be carefully managed and will be discussed in more detail in this section:

- Quality of the generalised dataset: quality control is used to certify the product against requirements

- Quality throughout the system: to minimise the amount of manual fixes at the end of the generalisation process, quality needs to be managed throughout the system.
- Consistency over time: During the lifetime of the system, things will change. The source data could change, the requirements of the product could evolve, and the software could also evolve. Still, the system needs to produce consistent output.

2.1 Controlling the quality at the end of the process

Evaluating the quality of a generalised map or dataset is a difficult task. [Stoter et al. 2009] explain that it can either be done manually by an expert cartographer eyeballing the results, or automatically using a set of constraints that collectively define the map specifications. Visually checking the data is very time consuming, and issues can easily get missed. There are also issues, like topological ones, which are extremely hard to spot visually. Automating the evaluation using rules or constraints can dramatically reduce costs. It is also more reliable than eyeballing the data. Fixing issues can either be done manually or by automatic fixes depending on the type of issue. The main limitation of automatic quality control is that it is very hard to come up with a set of rules that capture all the aspects of the quality of a map. It is therefore a good idea to always keep a small proportion of eyeballing the result, to pick up any unexpected problems. Any new issue detected should then lead to writing a new quality rule that can pick up the issue automatically in the future. It's therefore important to have a tool for authoring the quality rules which is flexible, and a quality control system capable of taking on new rules easily.

2.2 Controlling the quality during the process

Monitoring quality during the process

A good automated generalisation system should generate as few issues as possible, to reduce the cost of fixing them to the minimum. Automated generalisation is a complex process, made of many smaller sub processes that feed into each other. It's critical to ensure that each step produces good enough data for the next one to avoid a snowball effect. A lot of research has been done on constraint based strategies to perform automated generalisation, and those have emphasised the value of modelling each generalisation algorithm as an entity which has pre and post constraints, stating the conditions that can be checked automatically [Edwardes and Mackaness 1999].

Driving the process using quality criteria

One of the big challenges in designing automated solutions is the orchestration of generalisation tools. [Mackaness and Ruas 2007] explain the benefits of performing quality evaluation throughout the process. It enables the development of autonomous systems, in which qualitative evaluation is used to drive decisions taken by the system. This can take the form of rule based systems, where rules are used to trigger specific tools when a specific situation is detected, or optimisation systems, in which an optimisation engine drives a set of tools, with the objective of minimising the value of a cost function [Sester 2005], or maximising the overall satisfaction of a set of constraints [Barrault et al. 2011], [Ruas and Duchene 2011]. These are not mutually exclusive, 1Generalise [Regnauld 2016] uses rules to control the overall process, and these can trigger Agent based processes to resolve specific issues.

2.3 Ensuring consistency over time

Ensuring that a complex system can evolve over time, cope with change both in its input and output specifications, and still deliver the required output, is a massive challenge. Development techniques have greatly improved in the past decade, and techniques like test driven development and methodologies like AGILE have done a lot towards ensuring that each individual tool and each system delivers results which meet customer requirements. A good coverage of the code with unit tests limits the risk of individual algorithms producing unexpected results. Integration tests ensure that a complex system delivers expected results against known scenarios.

These development techniques do not remove the need to check the data during or at the end of the process. However, they create a framework in which it's safe to introduce change, as most adverse impacts of the change will be immediately detected and corrective actions can be taken before the change is deployed in production.

2.4 1Spatial rule authoring system

The use cases described in the next two sections use 1Spatial software to define and run business rules on the data. 1Validate is used for data validation, 1Generalise for reducing the level of detail in the data, and 1Integrate for data conflation. In all these software components, the rules are defined via a web-based user interface and applied to data either via a user interface or programmatically via a web services API. Rules can access both spatial and non-spatial data and can either check the quality of the data, improve, enhance or transform it according to the need. These rules, which are stored in xml, are centrally managed so over time they evolve and contribute to an ever growing 'knowledge repository'.

3 The GenIE System (Generalising Information Engine)

Over the past 3 years the GenIE programme has been building and delivering an automated system, jointly developed by Ordnance Survey and 1Spatial, to generate derived products from a maintained source database (MAIA). The vision behind GenIE is:

'To improve the currency and consistency of current products and allow simpler creation of new derived products that meet changing business and customer needs.'

The programme focused on the district resolution products (i.e. resolution of about 1/25000) required by the business to meet its Open Data commitment. The first results were resented in [Howland 2015]. The GenIE system was deployed to production in January 2017. Close working between Ordnance Survey and 1Spatial has made GenIE a real success story!

3.1 Overview of the system

3.1.2 High-level system architecture

The high level architecture of the system is illustrated in Figure 1. It shows the main components of the GenIE system in yellow. We can distinguish three main automatic phases. The first two follow a similar structure, generalising the data before validating it, performing manual editing and storing the result. The third one focuses on the publication and packaging of products, which we won't address in this paper.

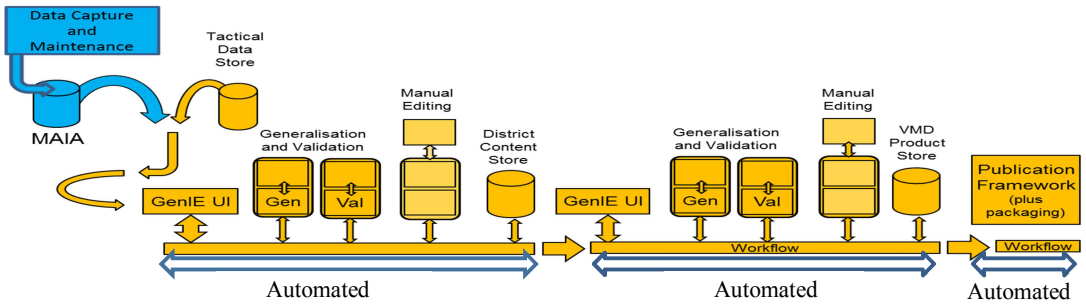

Figure 1: high level architecture of the GenIE system

Note 1: MAIA is the version-enabled (Workspace Manager) Oracle Spatial database which is the OSGB's core maintenance data platform.

Note 2: The Tactical Datastore stores data required for product creation which is not maintained within the core MAIA database eg. 3rd Party data, Product extents, Grids etc.

Note 3: The District Content Store is a generalised (resolution approx. 1:25,000, product agnostic, interim data store.

Note 4: VMD is the 'open-data' VectorMap District product.

3.1.3 High-level data architecture

Figure 2 shows the data architecture used by the GenIE system. The strategy used to derive multiple products uses a mix of the star and ladder strategies discussed in [Stoter 2005]. It follows a two steps ladder strategy by generalising first the core data to a single District resolution database. A second step is used to derive from it a variety of products. The star strategy means there is no dependency between content stores at different resolution, or between products of a same resolution family. Generating a set of products from the same content store helps ensuring consistency in content and currency between them.

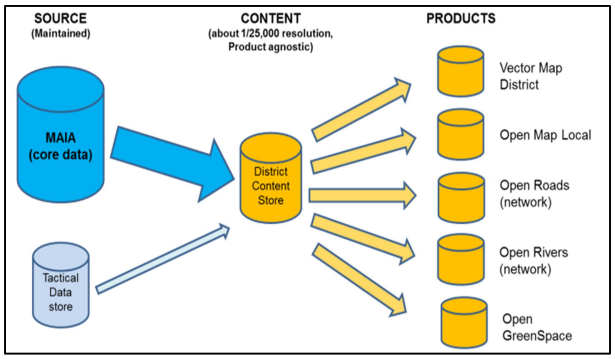

Figure 2: High level data architecture

3.1.4 GenIE End to End process

Data is extracted from the MAIA database plus a small tactical data store used for additional product information.

Due to the volume of the source data, data is processed in partitions (partition jobs run in parallel) and then 'stitched back together'. The partitions are dynamically created using the national road network, and by a grid outside of that network. After processing, the data is stored in a 'product agnostic' content store and validated. Following validation, manual editing is required to resolve any critical non-conformances. (Minimal manual editing - about 650 features edited (out of 24 ½ million features in the content store!). The requirement for editing mainly being due to poor source data, invalid geometries being created during processing or errors created during the road collapse process.

The process is then repeated to populate each product store, applying generalisation and validation to meet individual product specifications. Each product is generalised using its own flowline in 1Generalise, and has its own set of validation rules in 1Validate, although lots of components (actions and rules) are reused across products and tuned. A post process 'stiches' data together across partition edges.

After completion, the publication framework automatically creates national coverage products in multiple formats (vector and Raster) and packages them ready for customer supply.

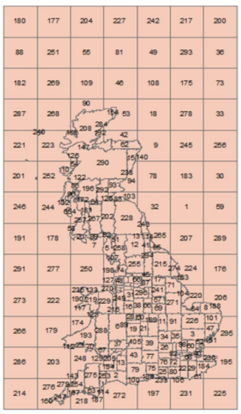

Figure 3: District Content Store partition

3.2 Quality control during development

GenIE was built using an AGILE methodology, which allowed it to be built in many iteration cycles, delivering richer and richer functionalities until a first product could be derived, soon complemented by other products. The risk is that a new development can adversely impact those already in place. To avoid this, an automated test framework is used to assure all new generalisation flowline development or changes to existing flowlines. Rules are built within 1Validate to test each flowline action against test cases with test data created to simulate all expected scenarios.
All new generalisation actions and development code is automatically passed through the test framework prior to being promoted through the Integration, Test and Production environments. This is to validate that:

- A new or amended generalisation action is delivering the required and expected outcome prior to being added to flowline, and is acting as expected within a flowline
- A new or amended generalisation action has no adverse impact on data as it is processed by subsequent actions in the flowline

3.3 Quality control in the production system

3.3.1 Overview

Quality Control is used throughout the system, each time data is committed to a persisted store. Automated validation is applied using rules created in 1Validate to ensure that each action in the generalisation processes (Source to Content, Content to Product) has been correctly performed. Validation is used to:

- Identify non-conformances. Non-conformances are categorised as either 'critical' or 'warning'.
- Maximise production efficiency. Data is validated after processing and committal to the content store followed by manual correction where required. Ensuring the data is valid in this interim store minimises the need for manual editing after the data has gone through product specific generalisation and removes any need for duplication of editing in the individual product stores.
- Ensure consistency in the published products.
- Remove the need for expensive manual 'eye-balling' of the data. Confidence in data quality gained during development means that following production processing, only a small sample, reflecting a range of geographies and feature types, is viewed for product assurance.

Figure 4 shows a screenshot of 1Validate, with on the left the tree structure holding all the rules required to validate the Content District data, and on the right, the detail of a single simple rule in the rule editor.

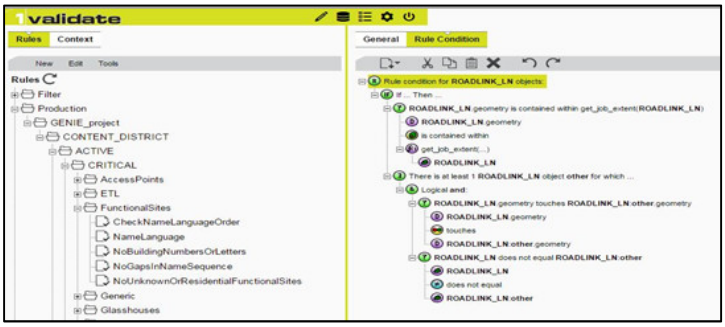

Figure 4: Validation rules for Content District

3.3.2 Categories of validation rule.

Critical

Critical rules are run to ensure that the data is valid and contains no data non-conformances that would negatively impact the usability of the product (e.g. connectivity issues in a network product).

Features identified by validation are grouped by proximity. Jobs are automatically created for each group of features, together with contextual data and a validation report extract. They are then presented to an editor for manual correction. Re-validation follows prior to commit to the target database.

Rule	Non Conformances	Processed Features
/CRITICAL/FunctionalSites/NoUnknownOrResidentialFunctionalSites	0	3428146
/CRITICAL/Generic/GeomIsValid	8	75160508
/CRITICAL/Generic/GeomMultiPart	0	75160508
/CRITICAL/Generic/GeomMustNotSelfIntersect	15	75160508
/CRITICAL/Glasshouses/GlasshousesMustNotHaveKickbacks	0	6557
/CRITICAL/Glasshouses/GlasshousesMustNotHaveSpikes	0	6557
/CRITICAL/Glasshouses/GlasshousesWithSmallHole	0	6557

Figure 5: Critical error report

Figure 6: Automatically created Edit Jobs – lifecycle management

Warnings

Rules classified as 'warnings' are also run, but non-conformances do not result in manual edit jobs. These are run to understand and monitor the general quality of the output data, and used to initiate quality improvement programs of the source data or the software.

3.3.3 Feature Counts

In addition to automated validation, automated feature count checks are made on the output data at each stage of processing. This can also show issues that have arisen during the end-to-end process. Validation is not very good at identifying data that is missing! These values are compared with previous runs, and differences analysed. Small differences that can be explained by differing source data currency are acceptable, bigger differences often highlight a change in the data that resulted in the system failing to extract or process them. These need investigation.

3.3.4 Causes of non-conformances

Errors in the output data can occur due to:

- Errors in the source data – errors are fed back to a data improvement team
- Scenarios in source data not previously identified – these are verified against the data specification, and referred to the development team
- Errors in a custom generalisation action – these are reported as defects back to the development team
- Failure of a standard generalisation action – these are fed back to the software supplier (1Spatial)
- Environmental failures – these are reported back to the appropriate team

3.4 Results and performance

All the products generated by the GenIE system are freely available at the following location: https://www.ordnancesurvey.co.uk/business-and-government/products/opendata-products.html

The derivation of the content store and the five related products has taken the following time:

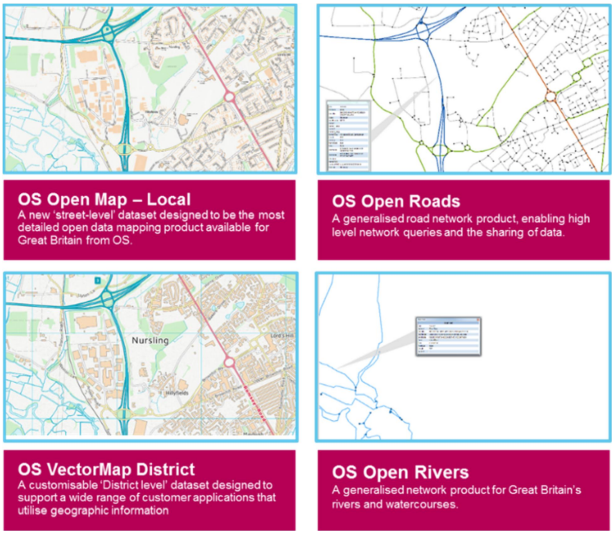

Figure 7: Examples of the National products created through the GenIE system

- Source to Content Store: 7 days processing + 5 man days for manual editing
- Content Store to Product Store (average for 5 products): 3 days processing + 2 man days for manual editing + 3 days post processing
- Publication and Packaging (average for 5 products, average 3 formats per product) – 2 days

This performance, allowing the district content store which contains 24 ½ million features, and the five products to be refreshed in less than 20 days, has been made possible by the heavy use of automated processes. 1Generalise and 1Validate have the ability to use parallel processing to process large areas, spreading the workload across a grid of processing nodes. The data flows automatically between all the steps in the process, and most importantly the manual editing jobs are generated automatically. This reduces the need for eyeballing the data to detect errors, which saves a lot time.

4 Case study for data conflation

Generating products on demand automatically has been the focus of researches for a few years. [Regnauld 2007] proposed a conceptual model for automating it. It relies on ontologies to automatically relate requirements to relevant data and to various processes. [Touya et al 2012] have gone further to model the spatial relations between thematic data and background data, in order to guide their integration and generalisation. This also relies on formalising the description and conditions of use of the generalisation operators [Gould and Chaudhry 2012]. In the present study, we are leaving the conceptual models aside, and focus on implementing a practical solution for a given requirement. The thinking is that with a few more practical examples, we can start classifying, parameterising and describing these rules according to the formal models which have been described, and start pushing the automation.

The aim of this study is to demonstrate how the rule based technology developed by 1Spatial, used in the Genie system to control generalisation as seen in section 2, can also be used to drive data conflation, with quality enforcement rules, so that the conflated data can be automatically generalised to produce a thematic map at smaller scale. Our target map is a small scale map of a Dutch city, showing the fastest through routes in the city. For this we use TOP10NL data from the Dutch Kadaster, the authoritative data provider in the country. We also need some speed limit data, available in Open Street Map (OSM). We also noticed that some main roads were missing from TOP10NL, as our OSM dataset was more current. So we need two types of conflation: addition of newly built major roads not yet in TOP10NL but available in OSM, and import of the speed limit values from OSM. Note that such conflation is only possible if the licencing of all data used allows it. In this case both OSM licence (Open Database Licence) and TOP10NL licence (CC-BY-3.0) allow their use for creating derivative databases.

4.1 Updating TOP10NL major roads using OSM

First we needed rules for matching and conflating two road datasets with different currency. Figure 8 shows a large section of motorway in OSM that is missing from TOP10NL.

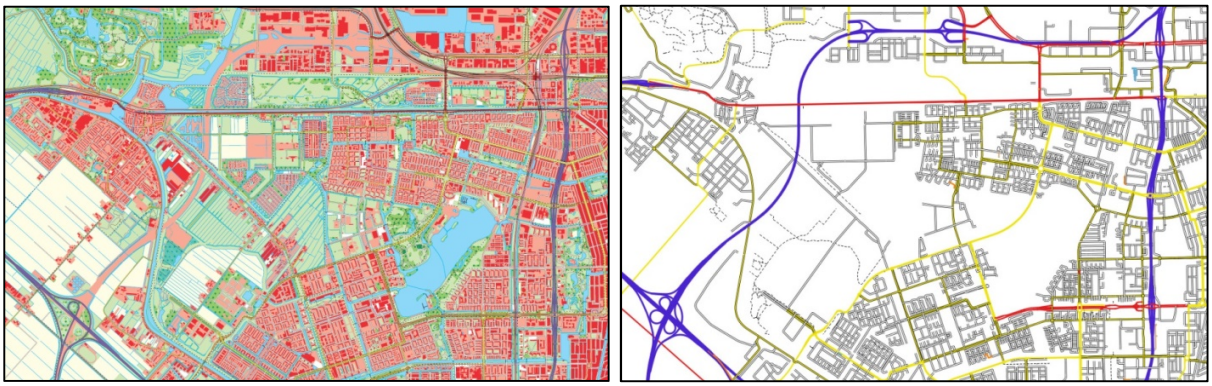

Figure 8: TOP10NL data (left) and OSM roads (right)

The conflation used in this case study did not go as far as solutions like Road Matcher [Vivid Solution] or the solution in Geoxygene [Mustière and Devogèle 2006]. These solutions aim at producing a full matching between two networks at different resolutions using only geometric and topologic properties. Here we only need to do a partial matching of networks (only major roads) which are at similar resolution. The advantage of our approach is that it can be quickly set up, and lets the user make full use of the semantic information available in both datasets.

The following rules bring together roads from two datasets and ensure that the quality of the network is good enough to support the automatic generalisation of the street network (density reduction):

1. Identify and transfer major roads that are present in OSM and haven't got a match in TOP10NL.
2. Enforce connectivity. Due to the differences in accuracy, the roads from OSM don't connect exactly to the TOP10NL ones. This rule therefore identifies these cases and enforces connectivity.
3. Split roads at junctions. When a new road joins an existing one at a new junction, the existing road is split in two. This step is necessary to ensure that the network is structured in a consistent way

Figure 9 illustrates rules 1 and 2. On the left, purple lines come from OSM, they do not connect properly to TOP10NL roads in green. One overshoot and on undershoot have been circled in red. On the right, these have been fixed as a result of applying rule 2.

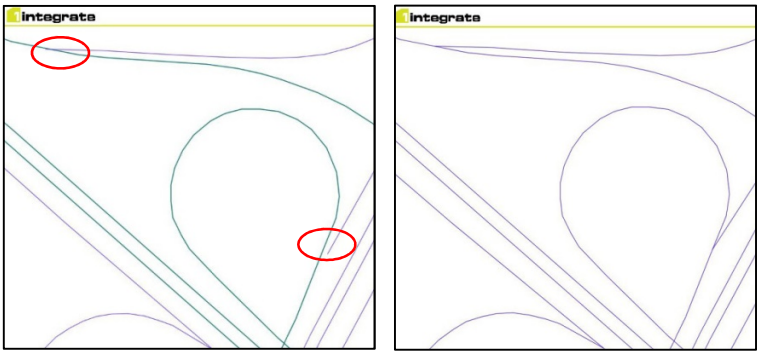

Figure 9: results of rule 1 (left) and 2 (right)

4.2 Bringing OSM speed limit values toTOP10NL roads

Analysing the OSM dataset has shown that some speed limits are invalid (800 km/h is too much!). We have also noticed that the speed limit is not systematically populated. As the aim of this conflation is to produce a small scale map of the city showing the fast routes across the city, we have no need to process minor streets, which are not used for through traffic. We have also allowed ourselves to infer speed limit values missing from OSM, when possible, based on the values of connected roads of the same type.

We therefore wrote the following three rules to bring speed limit values to the TOP10NL roads, and ensure validity and consistency:

1. The first rule, shown in Figure 10, creates for each object from class "ROAD_LINK" (containing TOP10NL roads) a new object in the new class ROAD_LINK_CONFLATED. The attributes are unchanged, except for the speed limit. For each ROAD_LINK, we check all objects from class "roads" (containing OSM roads) that are within 15 meters of our road, and we compute the area of the intersection between the buffers created at a distance of 15m around these two roads. The OSM road for which this computed area is highest is selected as the best match for the conflated road being processed, and will give it its speed limit value. .
2. Remove any speed limits which is not valid. Here we used a simple rule that detects invalid values (negative or above 130km/h). We could write a more complete rule that takes the semantics of the road into account to set the highest valid speed limit for each type of road.
3. Fill in empty speed limits based values of adjacent roads of the same type.

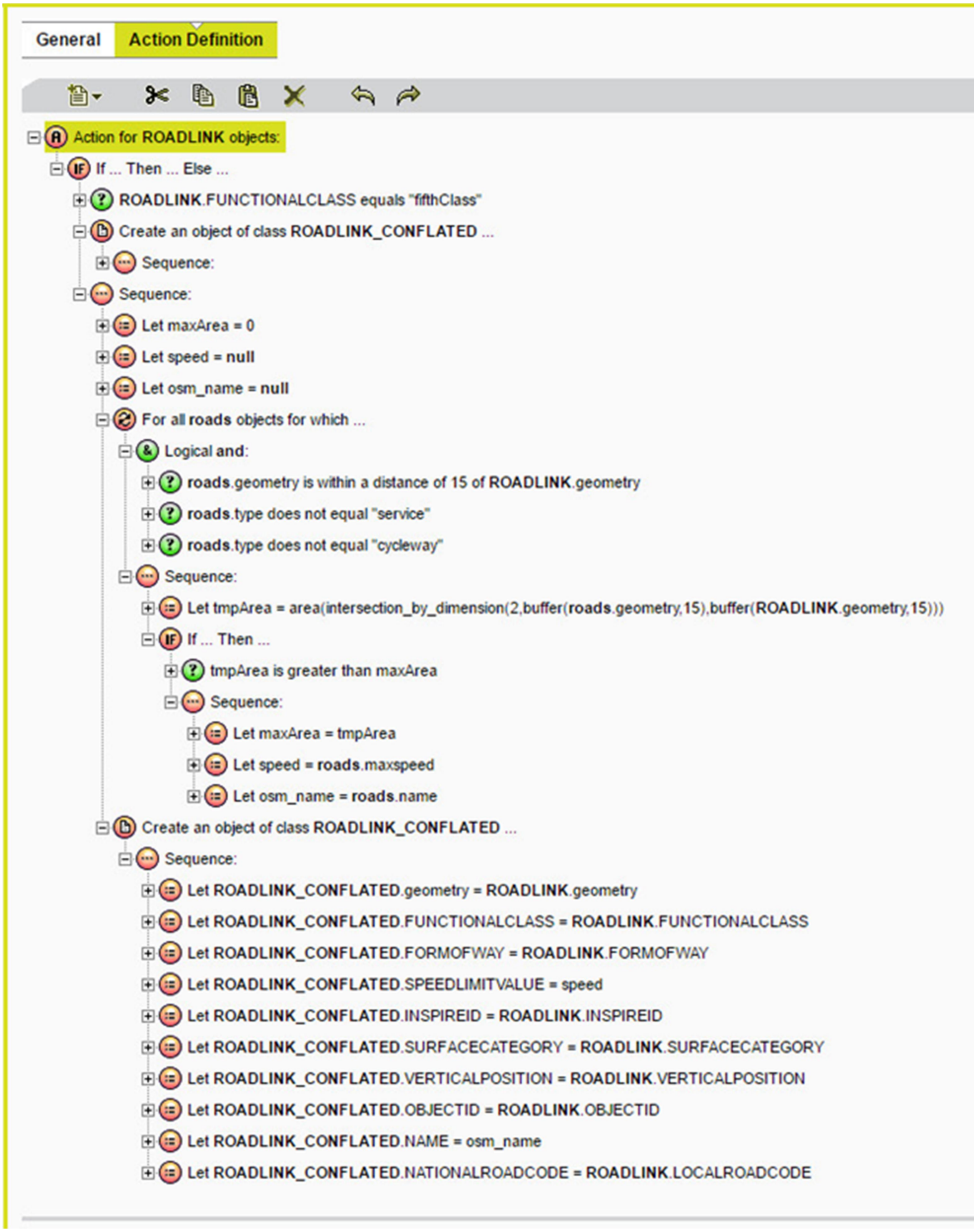

Figure 10: rule to transfer OSM speed limitations to TOP10NL roads

4.3 Results

The results of these conflations can be seen on Figure 11. On the left, the test area is styled for 1:25,000. We can see that the missing motorway is present on our generalised map. On the right a thematic map highlights the best through routes in the city, roads are styled based on their type (width) and speed limit (colour).

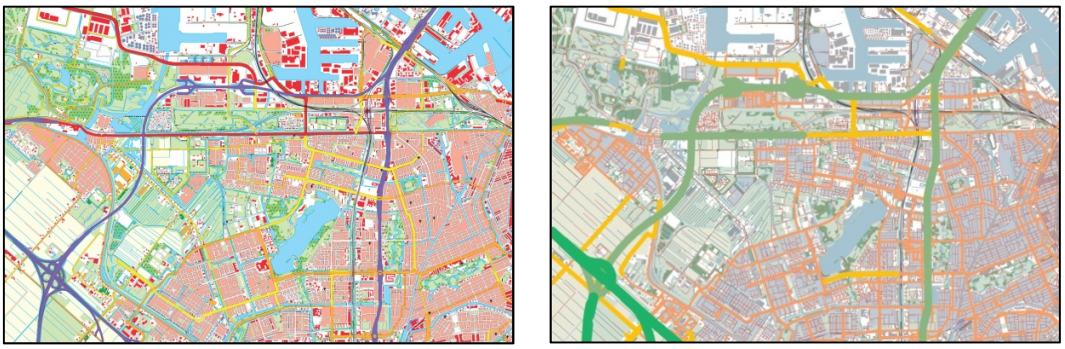

Figure 11: Conflated data generalised and styled (left), styled according to speed limits (right)

4 Summary and conclusions

This paper has focused via two use cases, on the importance of managing quality throughout the generalisation and data integration processes. In both cases, 1Spatial rules technology was used:

- It was used in the GenIE system of Ordnance Survey of Great Britain to control the quality of the data throughout the product derivation process. The system in place is one of the most advanced and efficient of its kind, maximising the automation of the generalisation tasks, validating the results automatically, and generating manual editing jobs to fix the residual amount of remaining issues. It also has a test framework allowing it to safely evolve over time, to support new product without putting existing ones at risk.
- It was used to demonstrate how to quickly conflate two datasets, and use the result for producing a thematic map (after generalisation).

This has shown the very high adaptability of these rules, they can be written to process any data, taking advantage of semantic geometric and topologic characteristics, discover within it all sorts of issues or correlation, and driving actions (corrections, generalisation, conflation, or producing reports).

The next challenge will be to make these rules more generic, or describe them in a machine readable way. This is a precondition to being used by a system that chooses and combines them automatically, to integrate data and generalise it automatically to produce thematic maps on demand.

References

Barrault M, Regnauld N, Duchêne C, Haire K, Baeijs C, Demazeau Y, Hardy P, Mackaness W, Ruas A, Weibel R (2001). Integrating multi-agent, object-oriented, and algorithmic techniques for improved automated map generalisation. In: Chinese Society of Geodesy Photogrammetry and Cartography (eds) Proceedings of the 20th international cartographic conference, Beijing, 2001

Gould N. and Chaudhry O. (2012). An Ontological approach to On-demand Mapping. In 15th Workshop of the ICA commission on Generalisation and Multiple Representation. Istanbul, Turkey, 13th-15th September 2012.

Howland D. (2015). OSGB Multi-Resolution Data Programme (MRDP). 2nd ICA / EuroSDR NMA Symposium, Amsterdam 3/4th December 2015
http://generalisation.icaci.org/images/files/workshop/symposium2015/OSGB_-_Presentation_Abstract.pdf

Mustière S., Devogele T. (2008). Matching networks with different levels of detail , GeoInformatica, Vol.12 n°4, pp 435-453

Regnauld N, 2007, A distributed system architecture to provide on-demand mapping. In International Cartographic Conference. Moscow, 4-10 August 2007.

Regnauld N. (2016). Automatic Generalisation for production, 19th ICA Workshop on Generalisation, 14th of June 2016, Helsinki, Finland.
https://kartographie.geo.tu-dresden.de/ica_gen/images/files/workshop/workshop2016/genemr2016_paper_09.pdf

Sester M. (2005). Optimisation approaches for generalisation and data abstraction. International Journal of Geographic Information Science, 19 (8-9), 871 – 897.

Stoter, J. (2005) Generalisation within NMAs in the 21st Century. 22nd International Cartographic Conference (ICC2005), A Coruña, Spain.

Stoter, J., D. Burghardt, C. Duchêne, B. Baella, N. Bakker, C. Block, M. Pla, N. Regnauld, G. Touya and S. Schmid (2009). Methodology for evaluating automated map generalisation in commercial software, *Computers, Environment and Urban Systems*, vol. 33, n. 5, pp. 311—324.

Touya G, Balley S, Duchêne C, Jaara K, Regnauld N and Gould N, 2012, Towards an Ontology of Spatial Relations and Relational Constraints. In 15th Workshop of the ICA commission on Generalisation and Multiple Representation. Istanbul, Turkey, 13th-15th September 2012

Vividsolution, Road Matcher: http://www.vividsolutions.com/products.asp?catg=spaapp&code=roadmatcher