

Data Validation and Quality Assurance with FME

First, Some Background

Mark Stoakes, head of the Professional Services department at Safe Software, recently gave a presentation on FME and its use in data validation and Quality Assurance. This article provides a summary of the presentation.

Mark began his presentation by acquainting the team with the ISO 19100 series of international standards that define quality requirements for geographic data. How does FME's data validation functionality measure up against these standards? As a platform designed specifically for spatial ETL, FME provides a superior tool for managing data quality. Taken together, FME's data model restructuring capability, FME Workbench's graphical interface for easy control of the data restructuring process, and the Universal Viewer application for checking data at different stages of the transformation, all combine to provide powerful and eminently user-friendly data validation capability. However, not all QA issues are equal – some are much more challenging to overcome than others. For a quick comparison, the following tables summarize how easily different feature geometry validation and attribute validation challenges can be addressed with FME. The degree of difficulty of each fix is indicated by the number of checkmarks, as follows:

- ✓✓✓ In most cases, the problem is easy to address with FME
- ✓✓ Problems can be addressed with careful configuration of the data transformation
- ✓ Advanced FME skills are needed. In these cases, for instance, data validation might require deconstruction of the features, checking their orientation, and reconstruction of the features
- ? FME's ability in this area has not yet been assessed.

Table 1: Meeting ISO Standards for GIS Data Validation – a Comparison of the Degree of Difficulty of Addressing Geometry Issues with FME

ID	Element Issue	Feature Type	Description of the Element	FME
1	Loop backs – self intersections	Line, Polygon	Termed “butterfly” polygons.	✓✓
2	Unclosed Polygons/Rings	Polygon	The start node and end node of the polygon or ring is not the same. This means that the feature cannot be closed.	✓✓✓



3	Internal Polygons with Incorrect Rotation	Polygon	Requirement for the internal polygon and the external polygon to have the order of nodes or vertices in a specific rotation direction. The external polygon should be clockwise and the internal polygon should be counter clockwise.	✓✓
4	Duplicated Points	Point, Line, Polygon	A point that duplicates exactly the same X,Y coordinates as another point.	✓✓
5	Kick Backs	Line, Polygon	Digitising error leading to an inconsistency in the line.	✓✓
6	Spikes	Line, Polygon	Digitising error leading to a spike inconsistency in the line. Similar to kick backs.	✓✓
7	Minimum Area	Polygon	A polygon feature should not be less than a specified area.	✓✓✓
8	Slivers or Gaps	Polygon	Very small overlaps or gaps between the boundaries of adjacent polygon features	✓✓
9	Overlapping Polygons	Polygon	A gross overlap of one polygon feature onto another	✓✓
10	Duplicate Polygons (duplicate polygons with same attributes)	Line, Polygon	A polygon that duplicates exactly the same geometry and attribution as an underlying polygon	✓✓✓



11	Short Segments	Line, Polygon	A very short distance between two nodes or vertices. This distance is specified and would be expected to be the same as the cluster tolerance on the dataset.	✓✓
12	Null Geometry – Table records with Null Shape	Point, Line, Polygon	No geometry is held against an attribute.	✓✓✓
13	Segment Orientation	Line, Polygon	Similar to Ring / Polygon rotation but at a finer granularity. The rotation between two nodes or vertices is checked rather than the entire feature	✓
14	Empty Parts – geometry has multiple parts and one is empty	Line, Polygon	Similar to null geometry. One geometry in a multipart feature is empty	✓

Table 2: Meeting ISO Standards for GIS Data Validation – a Comparison of the Degree of Difficulty of Addressing Various Attribute Validation Issues with FME

ID	Element	FME
1	All attribute headings are described in the attribution look-up table	✓✓
2	Each feature is described by a name and/or description	✓✓✓
3	Each feature within a dataset has a unique identifier / reference code	✓✓
4	All mandatory fields are populated	✓✓



5	Each area or linear feature has a measurement and a unit of measurement specified	✓✓✓
6	Blank and zero values have been qualified	✓✓
7	Date and time values conform to ISO 8601 standard	✓✓✓
8	References to countries or their subdivisions conform to ISO 3166 standard	?
9	References to language conform to ISO 639-2 standard	?
10	Fields are populated appropriately including coding and formatting	✓✓
11	Addresses conform to BS7666 part 3 standard	✓✓

FME's Data Validation Capability at Work

So how exactly does a user apply FME's spatial ETL capability to data validation projects? As mentioned earlier, one of FME's great strengths is the FME Workbench application. Within FME Workbench, the data transformation is designed and controlled via a graphical interface. Once the source and destination data formats have been specified, the user designs the required data transformation workflows by simply dragging pre-packaged data transformations, or FME transformers, from the transformer gallery and onto the FME Workbench canvas. Required parameters are set for each transformer, and the transformers are then connected together to form a graphical model representing a data flow pipeline. When the transformation is run, various transformations are applied to the data as it "flows" through each transformer connected into the pipeline.

Clearly, the number of transformers required to complete a data validation task will vary depending on the complexity and condition of the source data. In some instances, data validation can be ridiculously easy. Other projects require a staggering number of transformers and call for careful organization of the data flow pipeline – especially if the FME data transformation file is used in a team environment.

For a look at some of the key FME transformers most commonly involved in data validation transformations, we've included here an overview of two projects Mark described during his presentation. The projects represent two extreme ends of the spectrum of possible data validation projects: the first project was a relatively simple data cleaning task, but the second required complex configuration of hundreds of data transformations.



Quality Validation Project # 1: We Wish They All Could Be Like the California Coastline

The client involved in this project had been wrestling with line-work data covering a section of California coastline for quite some time. Beginning with a relatively small dataset – a section of the coastline that included about 3,000 features and 100,000 vertices – the client had had numerous unsuccessful attempts to extend the coastline and wrap the lines around to form an area feature. But each attempt had been foiled by the poor quality of the data. In some places, the line-work actually traced back over itself, making construction of a polygon almost impossible.

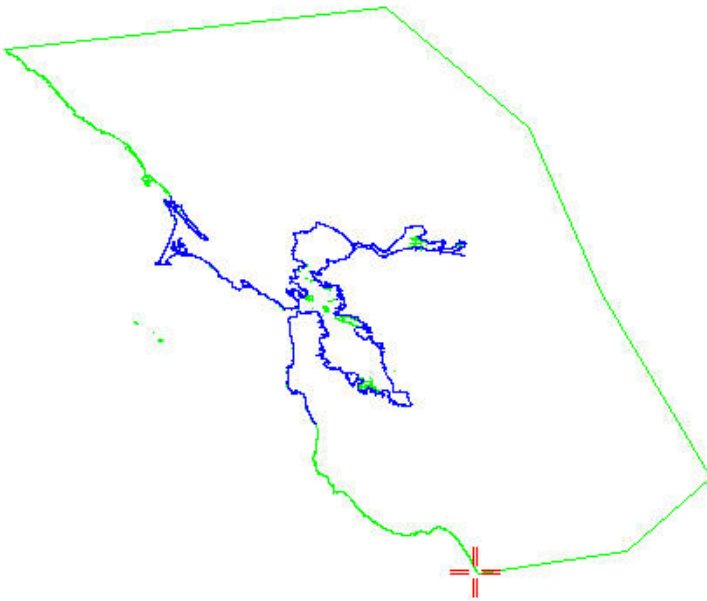
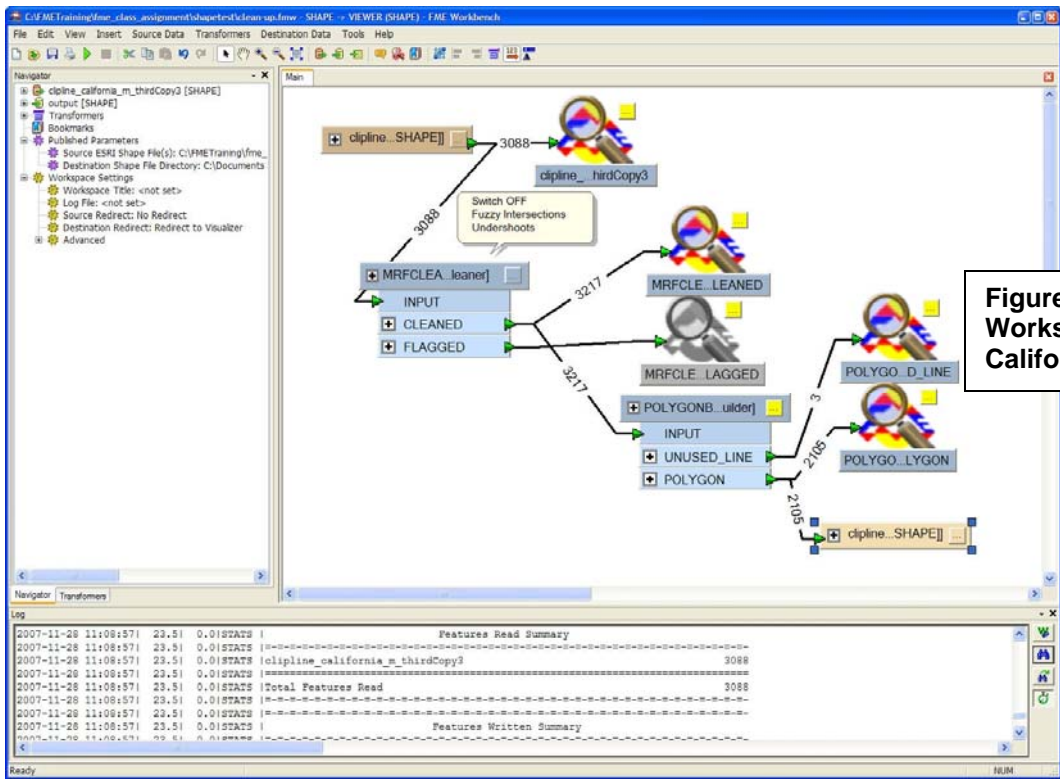


Figure 1: California Coastline Data – It Looks So Benign at this Scale...

When Safe Software became involved, the client was actually considering re-digitizing the entire coastline. Fortunately, Mark's team was able to save the client considerable trouble and expense with what turned out to be a very easy fix. Just two FME transformers – the MRFCleaner and the PolygonBuilder connected together in a simple data transformation workflow – took care of all the data quality issues in the dataset. The MRFCleaner offers many different data cleaning options. Although the MRF Cleaner can require some experimentation to get the parameters just right, it provides a very easy solution for exactly this type of problem, i.e. a small to medium-sized dataset and line-work that forms areas. In this case, the MRFCleaner took only a few minutes to process the data.





Quality Assurance Project # 2: Cell Phone Signal Strength Contours – An Acid Test for any Quality Assurance Tool

The second project, completed by Mark's team earlier this year, involved loading a set of cell phone signal strength contours in MapInfo TAB into ESRI ArcSDE for use with ESRI ArcIMS. The data had been generated by building a set of contours from a raster dataset, resulting in large datasets with poorly conditioned area features. In this state, the data could not be loaded into ESRI ArcSDE, since ArcSDE requires well-formed geometries.

Unlike the previous project, the size and complexity of the data sets tipped the needle on the difficulty meter well into the red zone for this project. The data consisted of multi-part donut polygons, with many single donuts in some polygons reaching in the order of 12,000 parts and 230,000 vertices. The largest dataset to be processed had 500 features and an average of 1,200 parts and 30,000 vertices. In other datasets, the number of parts and vertices ran much higher, reaching a maximum of 24,000 parts and 500,000 vertices.



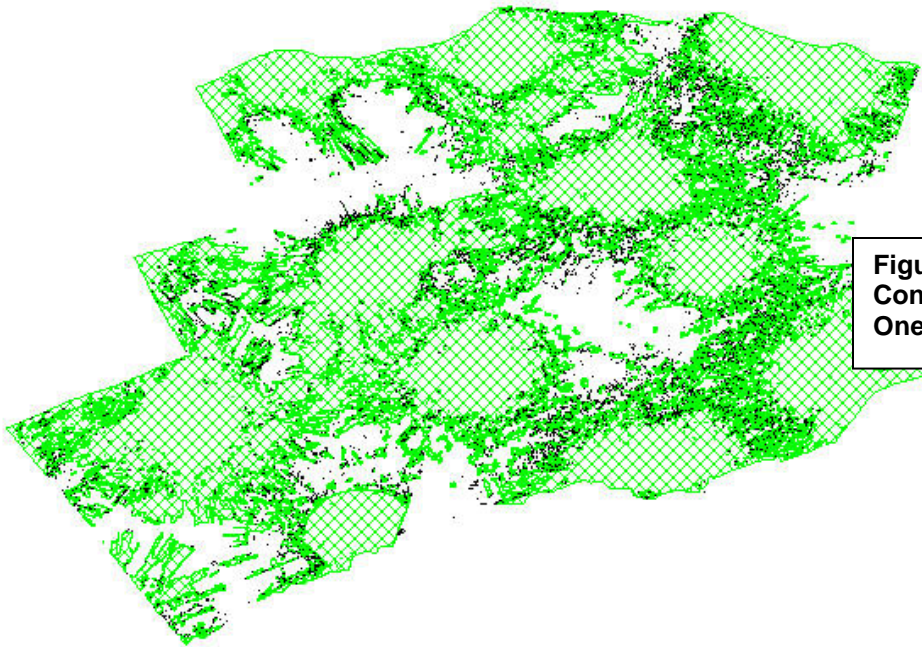


Figure 3: Cell Phone Signal Strength Contours in Need of Validation – No One's Leaving Work Early Today

Some of the data quality issues that had to be overcome included:

- donut holes that touched each other inside the polygon
- donut holes that touched the shell
- lines that reversed or overlapped
- duplicate line segments
- spikes and kick-backs
- areas that overlapped, including holes that overlapped with the shell of another donut



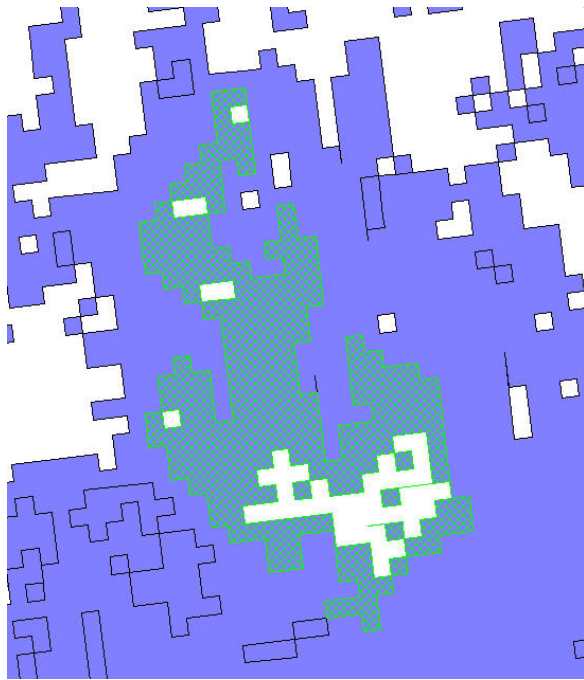


Figure 4: A Close-Up Look at Some of the Data Validation Challenges

There's no two-transformers-only fix for this one. Unscrambling these datasets required a complex FME Workbench workspace of multiple FME transformers arranged into five groupings to represent different phases of the data processing. The various phases of data processing were:

- primary dataset restructuring
- cleaning of the donut shells
- cleaning the donut holes
- re-cleaning the donut holes
- rebuilding the donuts
- then loading the data into ArcSDE.

Here's a detailed description of the data transformations required in each of these phases:

Dataset Restructuring

The primary restructuring phase involved:

- deaggregation of individual polygons and donuts using the Deaggregator transformer
- reprojecting the data with the Reprojector
- filtering out non-area features with the GeometryFilter
- extracting the shells and holes using the DonutHoleExtractor



- then generating polygon centroids with the InsidePointReplacer. (Since the shells and holes had been separated out, the polygon centroids were needed to reconstruct the dataset correctly later.)

Cleaning Donut Holes and Shells

The second processing phase – cleaning of the donut shells and holes – was, in itself, a multi-step process. First, some of the smaller areas were dropped from the dataset by adjusting tolerance values.

The second step, cleaning the lines, involved deconstruction of the original donut features into individual two-point lines, with FME's Chopper transformer, then reconstructing the features again with the LineJoiner to create a dataset with fewer line segments. This simplified the features in preparation for cleaning the lines with the MRFCleaner. As a final step in the line cleaning process, the Intersector transformer was used to check for extraneous duplicate lines.

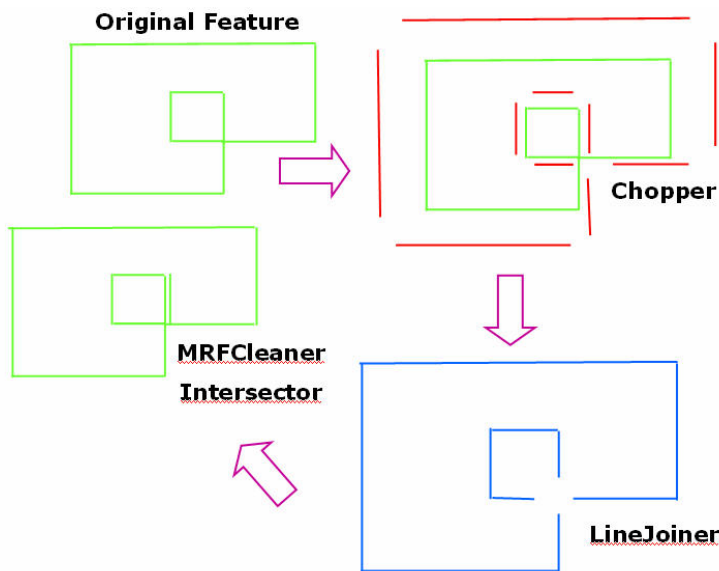


Figure 5: Steps Required to Clean the Donut Holes

At this point, the data consisted of simple line-work features. To rebuild the polygons, linework that closes was coerced into polygons using the GeometryCoercer. Other lines were built into polygons using the PolygonBuilder. Next, a PointOnAreaOverlayer was used to determine which of the holes belonged with each donut shell.

Re-cleaning Donut Holes

To re-clean the donut holes, the Bufferer transformer was used to shrink the holes so that none of the holes touched a surrounding shell, or each other. Applying the LineGeneralizer as the next step ensured FME did not treat the holes as self-intersecting areas. Lastly, the polygons were rebuilt using the PolygonBuilder.

Rebuilding Donuts

As a final processing step, the donuts were rebuilt using the DonutBuilder, then the data reprojected back to the original source data.



Summary and Conclusions

All in all, this second project presented an interesting challenge. And the results were encouraging: FME was able to process the entire dataset in 2-3 hours, whereas rivaling technology required many more hours to accomplish the same task.

As always, there remains more work to be done. The key takeaway was to continue working towards higher goals for FME, including handling even larger data sets while maintaining that FME's validation functionality is scalable. This may mean future improvements to FME's memory management or building check-pointing capability into the FME Workbench itself. Whatever the next need may prove to be, customers will continue to be forefront for Safe Software as future versions are planned and released.

If you require more information on data validation solutions and other services offered by Safe's Professional Services team, please visit Safe's website www.safe.com/services.

