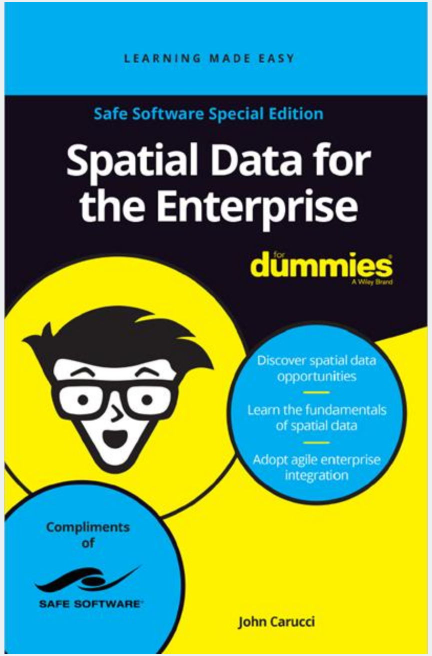
Data Validation Done Dynamically

Michael Oberdries | locusglobal.com

LOCUS

FME by Safe Software

Spatial Data for the Enterprise (2023)



The four "Vs" of (Spatial) Data Integration Platforms ...

- Variety
- **Veracity**

  Defined as the accuracy or quality of a dataset, this is something that organisations strive to constantly improve.
- Volume
- Velocity

Why is Data Veracity is important to us …

- **Data Veracity** refers to the quality, accuracy, integrity, and trustworthiness of data

- **Data Veracity** is crucial because organisations must have confidence in the accuracy and completeness of their data, as this directly affects the quality of their decision-making processes.

How we manage Data Veracity …

Consider an Enterprise Geodatabase, we typically have a **Data Model** that defines schema-level rules and constraints in our spatial databases

- Data Types (attributes and geometry)
- Attribute Domains (permitted codes and descriptions)
- <Mandatory> Attribute Rules
- <Null> Attribute Rules
- Relationship Classes (to manage dataset associations)
- Geometry Rules (topological relationships)
- Attribute Rules (sequences and triggers)

A Data Migration case-study …

OBJECTIVE: Migrate Oracle Spatial 11g (OnPrem) environment to an ArcGIS Enterprise Geodatabase (Azure)

Comprising **105** core spatial datasets (maintained internally)

- **69** tables with geometry (point, polyline, polygon)
- **31** tables (no geometry)
- **3** attributed relationship classes (M:M)
- **2** relationship classes (1:M)

** Significant variance between the source and target Data Models

Data Migration Strategies using FME …

Since the arrival of the **SchemaMapper** transformer (circa 2011), the best way to manage the “**data-mappings**” between source and target data schema that do not align, has been to use an Excel spreadsheet

This approach allows us to “**de-couple**” the schema Data-Mappings from the FME Data processing

<sample data model spreadsheet>

From Data Mapping to Data Modelling …

For most Data Migration processes, **Data Mapping** alone (source to target) is often not enough …

… we need to **Validate** our data as well

<sample data model spreadsheet>

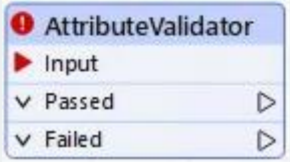
Case-study Data Model spreadsheet …

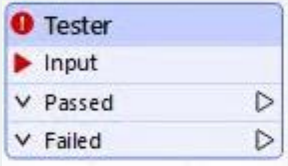
- **100** feature classes and tables,
- comprising **2,856** attributes,
- where each attribute being subject to **6** data validation tests

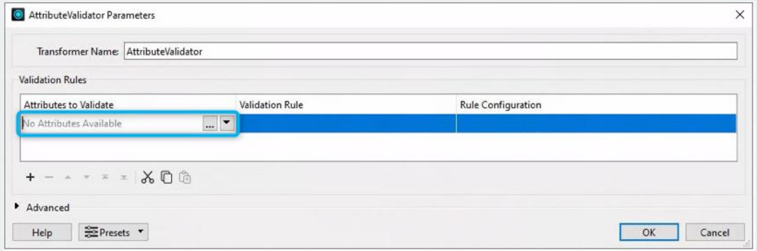
= **17,136** tests (too many!)

Embedding a Data Model in FME …

All the transformers we might use for FME data validation require us to reference the <attribute> being validated …







Is there a better way ?

Can we leverage the efficiencies inherent to FME **Dynamic Schema** to enable **Dynamic Validation**?

Dynamic Schema revisited …

FME Workspace DEMO …

Showing the differences Static Schema and Dynamic Schema …

For a Static Schema we expose schema when we **READ** a data source

versus

For a Dynamic Schema we expose schema when we **RUN** a workspace

Dynamic Validation enabled …

FME Workspace DEMO continued …

How can we enable Dynamic Validation when none of our Dynamic Schema attributes are **exposed** … and when the FME Transformers we use for "data validation" all require an **<attribute name>** as an input?

Dynamic Validation …

DEMO RECAP : A single FME workspace that enables us to process:

- **100** feature classes and tables (using a single FeatureReader);
- comprising **2,856** attributes;
- where each attribute is subject to **6** data validation tests;
- meaning total data validation checks executed is **17,136**

- **Data validation** gets simplified down to "this <attr_name> with this <attr_value> complies with this <data_model> constraint" using just (3) "abstracted" attributes

Data Validation Done Dynamically requires us …

To Read and Write the data using **Dynamic Schema**

- FME Feature Information window to inspect unexposed attributes

To use the **AttributeExploder** to “abstract” **unexposed** Dynamic Schema attributes as **exposed** [attr_name] and [attr_value] pairs

Thereafter, any FME data-validation transformer need only reference the **exposed** name-value-pair attributes, and the data model constraint attributes they are being tested against

Reporting Dynamically …

- Allows data-validation messaging at runtime (for fast fixes)
- Simplifies the generation of custom FME log files
- Simplifies the generation of custom Stat files (via Python shutdown scripts)

Benefits of Dynamic Data Validation …

- A single workspace handles <u>all</u> Data Migration and <u>all</u> Data Validation
- De-coupling the Data Model (as a spreadsheet) means updates can be made without those changes impacting the FME workspace
- Merging “abstracted” **Dynamic Schema** attributes with the **Data Model** gives us everything needed for **Dynamic Data Validation**
- Working with “abstracted” <u>name-value-pair</u> attributes greatly simplifies the FME workspace data processing and authoring

Thank you!

Michael Oberdries | locusglobal.com