Data Models
For GIS-Transportation

Kyle Gonterwitz
Kansas Department of Transportation
April 24, 2019

A DEEP DIVE
A note about the use of UML for geodatabase design

There are several strategies for designing and creating geodatabases:

- Existing databases can be migrated to the geodatabase
- Use the schema management and creation tools within ArcGIS
- Use one of the Pre-defined ArcGIS data models to jump-start your design
- Use Unified Modeling Language (UML) to design a geodatabase schema and then import the schema into your geodatabase.

We recommend using Sparx Systems’ Enterprise Architect for users interested in geodatabase design with UML. Enterprise Architect uses Geodatabase XML and the XML import\export functionality for the creation of geodatabases based on a UML design. Because of this, many of the limitations that existed with previous UML modeling solutions and CASE tools are not present in Enterprise Architect.

Some of the benefits of using Enterprise Architect include:

- The ability to model a larger set of datasets within the geodatabase, such as network datasets, topology, mosaic datasets and other datasets responsible for modeling richer geographic behavior as part of its UML profile for ArcGIS.
- The ability to reverse engineer an existing geodatabase into UML, since Enterprise Architect uses Geodatabase XML and the XML Workspace Import/Exporting functionality. Using the Export XML Workspace Document wizard or geoprocessing tool, the geodatabase schema can be exported to an XML workspace document. The XML document can then be imported into Enterprise Architect to provide a visual model of your geodatabase, based on industry-standard UML notation.
- Traceability to other analysis and design models
- Access to UML design techniques, such as the ability to logically group elements through the use of abstract classes.
- The ability to share your geodatabase schema and geospatial concepts with analysts and architects versed in UML design.

For more information on using Enterprise Architect for geodatabase design, visit Sparx System’s website (http://www.sparxsystems.com/arcgis/index.html).

Data Modeling Design Tips

Geodatabase data models are designed to be used in practical application scenarios by a wide range of users. To ensure that each design is easy to understand and implement, each data model was built to support easy migration from existing data structures and has been designed to be flexible, extensible, and easily adapted by your organization. Here are a few final design tips to help you with your design implementation:

- **Build on your existing GIS designs.**
  Most existing database designs are suitable for moving forward. You can build on what has worked in the past and find new geodatabase capabilities that will improve on your past efforts.

- **Use generic geodatabase types whenever feasible.**
  Combining generic data structures with very rich GIS tools provides the best solutions that will scale and support multiple users and applications. Leverage the ArcGIS software logic as much as possible for your work. Only use customized GIS data structures as a last resort.

Data Modeling Design Tips

- **Integrate related feature classes using topology.**
  
  Legacy ArcGIS Desktop Advanced users with coverages will find many opportunities to integrate feature classes using topologies in the geodatabase. Learn how to use geodatabase topology and its rules. This will create real savings during editing, minimize the amount of customization work you’ll need, and increase user productivity. Even small GIS organizations will see up to a 40 percent increase in efficiency for data maintenance.

- **Combine GIS design concepts from this section with traditional relational database design methods.**
  
  Both database management system (DBMS) and GIS design methodologies are critical for good GIS design. One is not sufficient without the other. Learn to use and apply both sets of techniques.

- **Prototype and pilot your geodatabase design.**
  
  Prototyping a design with ArcGIS Desktop using a file or personal geodatabase, or a geodatabase for SQL Server express, is easy, fun, and effective. You’ll be surprised at how much insight you’ll gain through experimentation and how much more effective and efficient your design process will become.

  During the final stage of design, you’ll want to test scalability and workflows that represent the work that your organization will perform with your geodatabase. Use this to make final adjustments to your design. Be practical in your final test phase and adjust your design as necessary.

Geodatabase Design for ArcGIS

The ArcGIS system, developed by Esri, supports the development and management of geodatabases. As for other databases, it is useful to model the design of a geodatabase using a standard notation such as UML. You can perform such modeling in Enterprise Architect, using the UML profile for ArcGIS, which is part of the built-in MDG Technology for ArcGIS. Once you have modeled an ArcGIS schema in Enterprise Architect, you can export the model to ArcGIS as an XML Workspace document. You can also visualize an existing ArcGIS geodatabase schema, by importing the ArcGIS XML Workspace document into Enterprise Architect.

Notice of Acknowledgement:

Support for modeling ArcGIS databases in Enterprise Architect was developed in collaboration with the Commonwealth Scientific and Industrial Research Organization (CSIRO), who defined mappings between UML 2 and ArcGIS concepts, and prototyped an automated import and export capability for ArcGIS geodatabase schemas represented in UML.

Notes

- The MDG Technology for ArcGIS is available in the Professional, Corporate, Unified and Ultimate editions of Enterprise Architect
Importing a model

![Importing a model.png]
Importing a model

[Image of a software interface showing the import of an ArcGIS XML package]
Importing a model

Import Package from ArcGIS XML

Enterprise Architect

XML contains a Model which may be placed at the root level in the Project Browser. Press Yes to import as Root Model, No to place under selected package.

Yes  No

Recent

EA DataModels
EAExample
Importing a model
Result in Sparx
Kansas Railroads, ALRS

- Transportation Planning produces Railroad Map
- Transportation funding for rail improvements
- Planning for track weight, unit trains, intermodals
- Tracking counts, owners, operators, leases
- Rail Crossing Collector App data model, FRA
ALRS Minimum Schema

- Railroad feature class populated with Route ID
- Track Mileposts at begin/end from track charts
- Crossing Mileposts set to Route Measures
- Crossings snapped to highway “intersections”
- Crossing Mileposts appended to Calibration Pts
- All of that cleaned up to non-monotonic state
A feature dataset is a collection of related feature classes that share a common coordinate system. Feature datasets are used to spatially or thematically integrate related feature classes. Their primary purpose is for organizing related feature classes into a common dataset for building a topology, a network dataset, a terrain dataset, or a geometric network.

Use feature datasets to organize spatially related feature classes into a common dataset:

- To add a topology
- To add a network dataset
- To add a geometric network
- To add a terrain dataset
- To add a parcel fabric

A Linear Referencing System Data Model may also apply

Feature Datasets

- First, create ALRS
- Next, Add Feature Dataset
- Use M Tolerance from Network to Feature Dataset
- Then Move Feature classes to Dataset

A Linear Referencing System Data Model may also apply

A Linear Referencing System Data Model may also apply:

1. Create ALRS
2. Add Feature Dataset
3. Use M Tolerance from Network to Feature Dataset

Railroad ALRS - Minimum
Railroad ALRS - Minimum

TIP! Export XML from a File Geodatabase as a baseline
Export XML Workspace Document

This wizard lets you export data from this geodatabase to an XML workspace document file.

Exporting data from: sde.DEFAULT - khdbtransdev

What do you want to export:
- Data
- Schema Only

How do you want the geometry to be represented in the XML document:
- Binary (smaller)
- Normalized (larger)

Specify the output XML file:
C:\gisdata\Railroads\XMLExport.xml

- Export Metadata
Railroad ALRS - Minimum
Import to Sparx
Import to Sparx
Import to Sparx
Import to Sparx
Validate, get Errors & Warnings
Validate, Fix Errors & Warnings
ALRS Activity Type - Default domain required for minimum schema

BEST PRACTICE
The advanced linear referencing system (ALRS) supports the use of a single polyline feature class, known as centerlines, to store the geometry for multiple routes. Centerline geometry, along with route definitions, is stored in a network feature class. A many-to-many relationship exists between the routes in the network and centerlines proving the geometry. This means that routes are typically made up of multiple centerline features, and centerline features can participate in multiple routes in multiple networks. The many-to-many relationship between routes and centerlines is maintained through a cross-reference table known as the centerline sequence table. The centerline sequence table contains a reference to each centerline to indicate in which LRS Network each route participates. Since route IDs are not unique in the ALRS, the centerline sequence table also contains a reference to the network ID field of the LRS Network. The combination of the Network ID and Route ID creates a way of uniquely identifying each route in the ALRS.


The linear referencing system (LRS) is a collection of feature classes and tables that allow the storage and editing of calibrated routes.

The LRS supports the use of a single polyline feature class, known as centerlines, to store the geometry for multiple routes. Centerline geometry, along with route definitions, is stored in a network feature class.

A many-to-many relationship exists between the routes in the network and centerlines providing the geometry. This means that routes are typically composed of multiple centerline features, and centerline features can participate in multiple routes in multiple networks.
Spell Checking, adding Definitions
Spell Checking, adding Definitions
Validation Results – Feels Good!

System Output

ArcGIS Model Validation

Start Time: 12:35:01 PM
Loading Feature Dataset packages...
...done
Loading Spatial Reference elements...
...done
Loading other workspace elements...
...done
Validating Workspace Properties...
Validating Spatial References...
Validating Domains...
Validating Workspace Packages...
Validating Default Subtype Codes for Feature Classes...

Validation Complete: Found 0 errors, 0 warnings.

End Time: 12:35:10 PM

Best Practice
Feels Good Right? Think again.
Double Validation – Import XML

```python
...
Created on Jun 22, 2016
use this when you have an issue importing from Sparx EA
first pass validation
next, see if you can do this.
You might have to use it a lot it you are learning sparx.
things to look for:
domain values have the correct field type for the values
SmallInteger domains have code values less than about 30000
don't be tempted to go changing all the xml around in one session without frequently using this script

@author: kyleg
...
if __name__ == '__main__':
    pass

def ValidateToFileGDB():
    from arcpy import ImportXMLWorkspaceDocument_management, Exists, CreateFileGDB_management, Delete_management

    xmlIn = r"C:\temp\ValidateThis.xml"
gdbname = "Validate"
GDB_In = r"C:/temp/"+gdbname
gdbin = GDB_In+".gdb"
if Exists(gdbin):
    Delete_management(gdbin, "Workspace")
    CreateFileGDB_management(r"C:/temp", gdbname, "CURRENT")
    print("new geodatabase created")
else:
    CreateFileGDB_management(r"C:/temp", gdbname, "CURRENT")
    ImportXMLWorkspaceDocument_management(gdbin, xmlIn, import_type="SCHEMA_ONLY")
ValidateToFileGDB()
print("validated")
...
Python Script to Double-Validate

```python
# dont be tempted to go changing all the xml around in one session without frequently using this script

if __name__ == '__main__':
    pass

def ValidateToFileGDB():
    from arcpy import ImportXMLWorkspaceDocument_management, Exists, CreateFileGDB_management, Delete_management

    xmlIn = r"C:\temp\ValidateThis.xml"
    gdbname = "Validate"
    GDB_In = r"C:\temp\"+gdbname
    gdbin = GDB_In+".gdb"
    if Exists(gdbin):
        Delete_management(gdbin, "Workspace")
        CreateFileGDB_management(r"C:\temp", gdbname, "CURRENT")
        print(\"new geodatabase created\")
    else:
        CreateFileGDB_management(r"C:\temp", gdbname, "CURRENT")
        ImportXMLWorkspaceDocument_management(gdbin, xmlIn, import_type="SCHEMA_ONLY")
    print(\"validated\")
    ValidateToFileGDB()
```

new geodatabase created
Traceback (most recent call last):
  File "C:\workspace\PyDev106\railalrs\ValidateModel.py", line 34, in <module>
    ValidateToFileGDB()
  File "C:\workspace\PyDev106\railalrs\ValidateModel.py", line 33, in ValidateToFileGDB
    ImportXMLWorkspaceDocument_management(gdbin, xmlIn, import_type="SCHEMA_ONLY")
  File "C:\Program Files (x86)\ArcGIS\Desktop10.6\ArcPy\arcpy\management.py", line 1371, in ImportXMLWorkspaceDocument
raise e
arcrestscripting.ExecuteError: Failed to execute. Parameters are not valid.
ERROR 001162: Invalid XML file.
Failed to execute (ImportXMLWorkspaceDocument).
```
Python Script to Double-Validate

don't be tempted to go changing all the xml around in one session without frequently using this script

@author: kyleg

```python
if __name__ == '__main__':
    pass

def ValidateToFileGDB():
    from arcpy import ImportXMLWorkspaceDocument_management, Exists, CreateFileGDB_management, Delete_management

    xmlIn = r"C:\temp\ValidateThis.xml"
gdbname = "Validate"
GDB_In = r"C:/temp/" + gdbname
gdbIn = GDB_In + ".gdb"
if Exists(gdbIn):
    Delete_management(gdbIn, "Workspace")
    CreateFileGDB_management(r"C:/temp", gdbname, "CURRENT")
    print("new geodatabase created")
else:
    CreateFileGDB_management(r"C:/temp", gdbname, "CURRENT")
    ImportXMLWorkspaceDocument_management(gdbIn, xmlIn, import_type="SCHEMA_ONLY")
ValidateToFileGDB()
print("validated")
```

<terminated> ValidateModel.py [C:\Python27\ArcGIS10.6\python.exe]
new geodatabase created
validated
Python Script to Double-Validate

ArcGIS Model Validation
=================================================================
Start Time: 1:36:11 PM
Loading Feature Dataset packages... done
Loading Spatial Reference elements... done
Loading other workspace elements... done
Validating Workspace Properties... Validating Spatial References...
Validating Domains...
Validating Workspace Packages...
Validating Default Subtype Codes for Feature Classes...
Validation Complete! Found 0 errors, 0 warnings.
End Time: 1:36:17 PM
=================================================================

35 print("validated")

Console

ValidateModel.py [C:\Python27;ArcGIS10.6;python.exe]
new geodatabase created
validated
To Design! Sparx Version Control
Data Model Version Control - TFS
Adding Package to Version Control
Comment on Version with Check-in

Please enter a comment for Package Addition.

EA Package addition: 4/1/2019 2:10:10 PM
Now..To Design!

**Add Comment**

Please enter a comment for Package Addition.

- EA Package addition: 4/1/2019 2:10:10 PM
- Initial Railroad ALRS design check-in to TFS Version Control

[OK] [Cancel]
To Design! Abstract Event Class

Minimum event fields

The minimum event fields are as follows:

<table>
<thead>
<tr>
<th>Field</th>
<th>Data Type</th>
<th>Length</th>
<th>IsNullable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event ID</td>
<td>String</td>
<td>Between 32 and 255</td>
<td>No</td>
<td>The unique ID for each event record.</td>
</tr>
<tr>
<td>Route ID</td>
<td>String</td>
<td>Length of the Network RouteID field or greater</td>
<td>No</td>
<td>The unique ID for each route in the network.</td>
</tr>
<tr>
<td>From Date</td>
<td>Date</td>
<td>8</td>
<td>Yes</td>
<td>The date that the event becomes active on the route.</td>
</tr>
<tr>
<td>To Date</td>
<td>Date</td>
<td>8</td>
<td>Yes</td>
<td>The date that the event is retired on the route.</td>
</tr>
<tr>
<td>Measure (point events only)</td>
<td>Any Numeric</td>
<td></td>
<td>No</td>
<td>The measure on the route where the event is located.</td>
</tr>
<tr>
<td>From Measure (line events only)</td>
<td>Any Numeric</td>
<td></td>
<td>No</td>
<td>The measure on the route where the beginning of the event is located.</td>
</tr>
<tr>
<td>To Measure (line events only)</td>
<td>Any Numeric</td>
<td></td>
<td>No</td>
<td>The measure on the route where the end of the event is located.</td>
</tr>
<tr>
<td>Location Error</td>
<td>String</td>
<td>100</td>
<td>Yes</td>
<td>The location error for the event.</td>
</tr>
</tbody>
</table>

To Design! Abstract Event Class

All Events shall have these fields
Event Abstract Class

All Events shall have these fields
Event Abstract Class – Attributes
Event Abstract Class
## Features & Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Scope</th>
<th>Stereotype</th>
<th>Alias</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROUTE_ID</td>
<td>esriFieldTypeString</td>
<td>Public</td>
<td>Field</td>
<td></td>
</tr>
<tr>
<td>FROM_MEASURE</td>
<td>esriFieldTypeDouble</td>
<td>Public</td>
<td>Field</td>
<td></td>
</tr>
<tr>
<td>TO_MEASURE</td>
<td>esriFieldTypeDouble</td>
<td>Public</td>
<td>Field</td>
<td></td>
</tr>
<tr>
<td>FIELD_ESTABLISH_DATE</td>
<td>esriFieldTypeDate</td>
<td>Public</td>
<td>Field</td>
<td></td>
</tr>
<tr>
<td>SYSTEM_CREATE_DATE</td>
<td>esriFieldTypeDate</td>
<td>Public</td>
<td>Field</td>
<td></td>
</tr>
<tr>
<td>SYSTEM_MOD_DATE</td>
<td>esriFieldTypeDate</td>
<td>Public</td>
<td>Field</td>
<td></td>
</tr>
<tr>
<td>RH_FROM_DATE</td>
<td>esriFieldTypeDate</td>
<td>Public</td>
<td>Field</td>
<td></td>
</tr>
<tr>
<td>RH_TO_DATE</td>
<td>esriFieldTypeDate</td>
<td>Public</td>
<td>Field</td>
<td></td>
</tr>
<tr>
<td>USER_CREATE</td>
<td>esriFieldTypeString</td>
<td>Public</td>
<td>Field</td>
<td></td>
</tr>
<tr>
<td>USER_MOD</td>
<td>esriFieldTypeString</td>
<td>Public</td>
<td>Field</td>
<td></td>
</tr>
<tr>
<td>LocError</td>
<td>esriFieldTypeString</td>
<td>Public</td>
<td>Field</td>
<td>LocError</td>
</tr>
<tr>
<td>SHAPE</td>
<td>esriFieldTypeGeometry</td>
<td>Public</td>
<td>RequiredField</td>
<td></td>
</tr>
<tr>
<td>SHAPE.STLength()</td>
<td>esriFieldTypeDouble</td>
<td>Public</td>
<td>Field</td>
<td></td>
</tr>
</tbody>
</table>
# Event Abstract Class – Idaho

## Features & Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Scope</th>
<th>Stereotype</th>
<th>Alias</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectId</td>
<td>esriFieldTypeOID</td>
<td>Public</td>
<td>RequiredField</td>
<td>Object ID</td>
</tr>
<tr>
<td>EventID</td>
<td>esriFieldTypeString</td>
<td>Public</td>
<td>Field</td>
<td>Event ID</td>
</tr>
<tr>
<td>RouteID</td>
<td>esriFieldTypeString</td>
<td>Public</td>
<td>Field</td>
<td>Route ID</td>
</tr>
<tr>
<td>FromDate</td>
<td>esriFieldTypeDate</td>
<td>Public</td>
<td>Field</td>
<td>From Date</td>
</tr>
<tr>
<td>ToDate</td>
<td>esriFieldTypeDate</td>
<td>Public</td>
<td>Field</td>
<td>To Date</td>
</tr>
<tr>
<td>FromMeasure</td>
<td>esriFieldTypeDouble</td>
<td>Public</td>
<td>Field</td>
<td>From Measure</td>
</tr>
<tr>
<td>ToMeasure</td>
<td>esriFieldTypeDouble</td>
<td>Public</td>
<td>Field</td>
<td>To Measure</td>
</tr>
<tr>
<td>EstablishedDate</td>
<td>esriFieldTypeDate</td>
<td>Public</td>
<td>Field</td>
<td></td>
</tr>
<tr>
<td>SystemCreateDate</td>
<td>esriFieldTypeDate</td>
<td>Public</td>
<td>Field</td>
<td>System Create Date</td>
</tr>
<tr>
<td>UserCreate</td>
<td>esriFieldTypeString</td>
<td>Public</td>
<td>Field</td>
<td>User Create</td>
</tr>
<tr>
<td>SystemModifyDate</td>
<td>esriFieldTypeDate</td>
<td>Public</td>
<td>Field</td>
<td>System Modify Date</td>
</tr>
<tr>
<td>UserModify</td>
<td>esriFieldTypeString</td>
<td>Public</td>
<td>Field</td>
<td>User Modify</td>
</tr>
<tr>
<td>LocError</td>
<td>esriFieldTypeString</td>
<td>Public</td>
<td>Field</td>
<td>Location Error</td>
</tr>
<tr>
<td>Shape</td>
<td>esriFieldTypeGeometry</td>
<td>Public</td>
<td>RequiredField</td>
<td></td>
</tr>
<tr>
<td>Shape.STLength()</td>
<td>esriFieldTypeDouble</td>
<td>Public</td>
<td>Field</td>
<td></td>
</tr>
</tbody>
</table>
## Event Abstract Class – Indiana

*As of about March 17, 2016*

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Scope</th>
<th>Stereotype</th>
<th>Alias</th>
</tr>
</thead>
<tbody>
<tr>
<td>FROM_DATE</td>
<td>esriFieldTypeDate</td>
<td>Public</td>
<td>Field</td>
<td>From Date</td>
</tr>
<tr>
<td>TO_DATE</td>
<td>esriFieldTypeDate</td>
<td>Public</td>
<td>Field</td>
<td>To Date</td>
</tr>
<tr>
<td>EVENT_ID</td>
<td>esriFieldTypeString</td>
<td>Public</td>
<td>Field</td>
<td>EventID</td>
</tr>
<tr>
<td>ROUTE_ID</td>
<td>esriFieldTypeString</td>
<td>Public</td>
<td>Field</td>
<td>RouteID</td>
</tr>
<tr>
<td>FROM_MEASURE</td>
<td>esriFieldTypeDouble</td>
<td>Public</td>
<td>Field</td>
<td>From Measure</td>
</tr>
<tr>
<td>TO_MEASURE</td>
<td>esriFieldTypeDouble</td>
<td>Public</td>
<td>Field</td>
<td>To Measure</td>
</tr>
<tr>
<td>RECORD_STATUS</td>
<td>dRecordCode</td>
<td>Public</td>
<td>Field</td>
<td>Record_Status</td>
</tr>
<tr>
<td>CREATED_BY</td>
<td>esriFieldTypeString</td>
<td>Public</td>
<td>Field</td>
<td>Created By</td>
</tr>
<tr>
<td>DATE_CREATED</td>
<td>esriFieldTypeDate</td>
<td>Public</td>
<td>Field</td>
<td>Date Created</td>
</tr>
<tr>
<td>EDITED_BY</td>
<td>esriFieldTypeString</td>
<td>Public</td>
<td>Field</td>
<td>Edited By</td>
</tr>
<tr>
<td>DATE_EDITED</td>
<td>esriFieldTypeDate</td>
<td>Public</td>
<td>Field</td>
<td>Date Edited</td>
</tr>
<tr>
<td>LOCERROR</td>
<td>esriFieldTypeString</td>
<td>Public</td>
<td>Field</td>
<td>LocError</td>
</tr>
<tr>
<td>DATE_ATTR_EFFECTIVE</td>
<td>esriFieldTypeDate</td>
<td>Public</td>
<td>Field</td>
<td>Date Attribute Effective</td>
</tr>
<tr>
<td>SHAPE</td>
<td>esriFieldTypeGeometry</td>
<td>Public</td>
<td>RequiredField</td>
<td>SHAPE</td>
</tr>
<tr>
<td>GLOBALID</td>
<td>esriFieldTypeGlobalID</td>
<td>Public</td>
<td>Field</td>
<td></td>
</tr>
<tr>
<td>SHAPE_Length</td>
<td>esriFieldTypeDouble</td>
<td>Public</td>
<td>RequiredField</td>
<td></td>
</tr>
</tbody>
</table>
**Event Abstract Class – Arizona**

*As of about 2016*

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Scope</th>
<th>Stereotype</th>
<th>Alias</th>
<th>Initial Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>OBJECTID</td>
<td>esriFieldTypeOID</td>
<td>Public</td>
<td>RequiredField</td>
<td>OBJECTID</td>
<td></td>
</tr>
<tr>
<td>FromDate</td>
<td>esriFieldTypeDate</td>
<td>Public</td>
<td>Field</td>
<td>FromDate</td>
<td></td>
</tr>
<tr>
<td>ToDate</td>
<td>esriFieldTypeDate</td>
<td>Public</td>
<td>Field</td>
<td>ToDate</td>
<td></td>
</tr>
<tr>
<td>EventId</td>
<td>esriFieldTypeGUID</td>
<td>Public</td>
<td>Field</td>
<td>EventId</td>
<td></td>
</tr>
<tr>
<td>RouteId</td>
<td>esriFieldTypeString</td>
<td>Public</td>
<td>Field</td>
<td>RouteId</td>
<td></td>
</tr>
<tr>
<td>FromMeasure</td>
<td>esriFieldTypeDouble</td>
<td>Public</td>
<td>Field</td>
<td>FromMeasure</td>
<td></td>
</tr>
<tr>
<td>ToMeasure</td>
<td>esriFieldTypeDouble</td>
<td>Public</td>
<td>Field</td>
<td>ToMeasure</td>
<td></td>
</tr>
<tr>
<td>FromReferentLocation</td>
<td>esriFieldTypeString</td>
<td>Public</td>
<td>Field</td>
<td>FromReferentLocation</td>
<td></td>
</tr>
<tr>
<td>FromReferentMethod</td>
<td>dReferenceMethod</td>
<td>Public</td>
<td>Field</td>
<td>FromReferentMethod</td>
<td></td>
</tr>
<tr>
<td>FromReferentOffset</td>
<td>esriFieldTypeString</td>
<td>Public</td>
<td>Field</td>
<td>FromReferentOffset</td>
<td></td>
</tr>
<tr>
<td>ToReferentLocation</td>
<td>esriFieldTypeString</td>
<td>Public</td>
<td>Field</td>
<td>ToReferentLocation</td>
<td></td>
</tr>
<tr>
<td>ToReferentMethod</td>
<td>dReferenceMethod</td>
<td>Public</td>
<td>Field</td>
<td>ToReferentMethod</td>
<td></td>
</tr>
<tr>
<td>ToReferentOffset</td>
<td>esriFieldTypeString</td>
<td>Public</td>
<td>Field</td>
<td>ToReferentOffset</td>
<td></td>
</tr>
<tr>
<td>LocError</td>
<td>esriFieldTypeString</td>
<td>Public</td>
<td>Field</td>
<td>LocError</td>
<td></td>
</tr>
<tr>
<td>created_user</td>
<td>esriFieldTypeString</td>
<td>Public</td>
<td>Field</td>
<td>created_user</td>
<td></td>
</tr>
<tr>
<td>created_date</td>
<td>esriFieldTypeDate</td>
<td>Public</td>
<td>Field</td>
<td>created_date</td>
<td></td>
</tr>
<tr>
<td>last_edited_user</td>
<td>esriFieldTypeString</td>
<td>Public</td>
<td>Field</td>
<td>last_edited_user</td>
<td></td>
</tr>
<tr>
<td>last_edited_date</td>
<td>esriFieldTypeDate</td>
<td>Public</td>
<td>Field</td>
<td>last_edited_date</td>
<td></td>
</tr>
<tr>
<td>SHAPE</td>
<td>esriFieldTypeGeometry</td>
<td>Public</td>
<td>RequiredField</td>
<td>SHAPE</td>
<td></td>
</tr>
<tr>
<td>SourceDataset</td>
<td>esriFieldTypeString</td>
<td>Public</td>
<td>Field</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OnRoadError</td>
<td>esriFieldTypeString</td>
<td>Public</td>
<td>Field</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TempReviewStatus</td>
<td>TempReviewStatus</td>
<td>Public</td>
<td>Field</td>
<td></td>
<td>TempReviewStatus</td>
</tr>
<tr>
<td>TempLegacyId</td>
<td>esriFieldTypeString</td>
<td>Public</td>
<td>Field</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SHAPE.STLength()</td>
<td>esriFieldTypeDouble</td>
<td>Public</td>
<td>Field</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Enough information yet?

Urban Dictionary: posterize

a Basketball term meaning to embarrass some one usually while slamming the ball over them. It refers to the guy who’s being dunked on in basketball posters.
Highlights – Indiana

Guardrail and End Types

Impact Attenuators

Junctions and Intersections

Entity Status Code

Cross Section Information

Image Credits

Creator: Brian Spurlock
Credit: Brian Spurlock-USA TODAY Sports
Copyright: Brian Spurlock
Information extracted from IPTC Photo Metadata
Highlights – West Virginia

Designated Truck Routes

CRTS – Coal Resource Transportation System

Route Dominance Event

Special Speed Limit Zones

MIRE Intersection Info

Reference Tables/Objects

Image Credits
Credit: WVU PHOTO
Highlights – Idaho

Weight Capacity

Speed Zone Types

Speed Change Reasons

Referents – used sparingly

Built In Safety Scoring

MP Description Codes

Junctions and Intersections

Concurrency – Primary/Overlap

Image Credits

Credit: DREW NASH – TIMES NEWS
https://bloximages.chicago2.vip.townnews.com/magicvalley.com/content/tncms/assets/v3/editorial/7/6a/76a5792b-f102-56cc-ab99-cdb08d18b5c6/5b7f72d777caf.image.jpg
Highlights – Arizona

Paint domain and features

Referents, Geocoding

Virtual Deletion

Stationing

Walls

Junctions and Intersections

Overall Scope of LRS Management

Image Credits
Creator: Michael Gonzales
Credit: NBAE/Getty Images
Copyright: 2018 NBAE
Information extracted from IPTC Photo Metadata
"We are like dwarfs sitting on the shoulders of giants. We see more, and things that are more distant, than they did, not because our sight is superior or because we are taller than they, but because they raise us up, and by their great stature add to ours."

- John of Salisbury, *Metalogicon*

"What Descartes did was a good step. You have added much several ways, and especially in taking the colours of thin plates into philosophical consideration. If I have seen a little further it is by standing on the shoulders of Giants."

- Isaac Newton, a letter to Robert Hooke

**Special Thanks to these Giants** - Nicole Hanson (Idaho), Yueming Wu (West Virginia), Kevin Munro (Indiana), Kevin Hunt (New York State), Erin Lesh (North Carolina) and James Meyer (Arizona) for sharing your XML Models with me.

Also thank you to DTS and FHWA with whom KDOT collaborated and assembled developed MIRE, HPMS, and proposed KDOT data models during our ARNOLD Pooled Fund Study in 2016.
Best Practices in Data Modeling

Kansas Department of Transportation
Questions about the Presentation?
Email: Kyle Gonterwitz@ Kyle.Gonterwitz@KS.gov
Questions for States

WVDOT - seems all the domains are also included as tables, are theses artifacts of implementation or is there a purpose, such as extensive use of lookup tables as opposed to domains? If lookup tables, how are those coordinated with domains as values are added?

Idaho – you have some tables that seem to indicate they are used specifically for HPMS validation, can you explain what these are and how these are used?

Arizona – you have included referent fields in almost every event, I am interested to learn about your referent implementation. How are the referents utilized in your state?

Can you explain how Virtual Deletion is used?

Many fields use a columns called PAINT and OTT Year, what are those?

What is the relationship between OnRoadError and LocError?