Assembling Vermont’s Intersection Data

Kerry Alley
GIS-T 2019
Interconnected Processes

Data Creation – Maintenance – Use – Evolution
Today’s Presentation

- Our objectives/approach to Intersection Data
- Internally developed automated processes
  - (facilitating data creation and maintenance)
- Future directions
VTrans Objectives/Approach

- MIRE and Safety Analyst compatible
- Extensible and flexible
- Requires little change in ongoing data maintenance efforts
- Leverages existing data
- Facilitates Agency-wide data integration
VTrans Objectives/Approach

- Intersection data capable of supporting intersection complexity and a generalized representation as needed, depending on use scenario
  - Major and Minor Routes AND Individual Legs, Complex geometry
VTrans Objectives/Approach

- Intersection data capable of supporting intersection complexity and a generalized representation as needed, depending on use scenario
  - Major and Minor Routes OR Legs, Complex geometry

- Intersection features “wedded” to Road Centerline
  - Approach (Leg) attributes extractedUPDATED from road centerline attributes
  - Relationships between features indicated by
    - Intersecting geometry
    - Unique IDs shared between related features
Two Intersection Feature Classes

Nodes
&
NodeLegs

Nodes
Point features

NodeLegs
Line features
VTrans Objectives/Approach

Intersection Complexity ➔ Generalized Intersection

4 7
VTrans Objectives/Approach

HOW?
VTrans Objectives/Approach

- One Node is the Principal Node
- A subset of the NodeLegs are Principal NodeLegs

Complex Intersection

Road Centerline

Nodes

NodeLegs
VTrans Objectives/Approach

- One Node is the Principal Node
- A subset of the NodeLegs are Principal NodeLegs
VTrans Objectives/Approach

- One Node is the Principal Node
- A subset of the NodeLegs are Principal NodeLegs
VTrans Objectives/Approach

- One Node is the Principal Node
- A subset of the NodeLegs are Principal NodeLegs
Nodes and Legs → Intersections

- **Simple Intersections**
  - Single Node and its Legs

- **Complex Intersections**
  - Multiple Nodes and all their Legs
  - Generalized to a single Node and one Leg per ‘approach’

- Also → Nodes with NodeLegs, but **Not** Intersections:
  - Dangles (one Leg, dead ends)
  - Pseudonodes (two Legs, split polyline)
  - Not-at-grade Nodes
  - Transitions between single and dual carriageways
Published “Safety Performance Functions” (SPFs) are based on safety statistics and real intersections.

Safety Analyst assumes intersection attributes are consistent with assumptions underlying each SPF.

Intersection Geometry and Stop Control assumed to be consistent with definitions of Major and Minor routes.

Major Route – Thru-route with more traffic.

Minor Route – Intersects Major route, has stop control.
Sort NodeLeg records in order of decreasing AADT!
How does knowing the rank of NodeLegs help us identify an intersection’s Major and Minor routes?

Which NodeLegs provide values for a Node’s attributes for Major and Minor routes?

How do we keep track of this in our data?
Leg Rank → Major & Minor?

- **Major** route attributes – Transferred from a Node’s “Major Leg” (highest AADT)

- **Minor** route attributes – Transferred from a Node’s Leg with highest AADT that *intersects* the Major route
How To “Automate”?

- Major 1 always highest AADT (except some T intersections)

- Major 2 always cross-street of Major 1
  - Major 2 = Leg 2 if Angle 1-2 ≈ 180°
  - Major 2 = Leg 3 if Angle 1-2 ≈ 180°

- Minor 1 is intersecting leg with highest AADT

- Minor 2 is cross-street of Minor 1
<table>
<thead>
<tr>
<th>Angle_1_2</th>
<th>Angle_1_3</th>
<th>Angle_2_3</th>
<th>Angle_1_4</th>
<th>Angle_2_4</th>
<th>Angle_3_4</th>
</tr>
</thead>
<tbody>
<tr>
<td>179</td>
<td>89</td>
<td>90</td>
<td>90</td>
<td>91</td>
<td>179</td>
</tr>
</tbody>
</table>
How To “Automate”?

Query subset of Nodes that will have same relationship between Leg rank and Major/Minor designation

(Angle_1_3 >= 175 AND Angle_1_3 <= 185) AND
(Angle_1_2 >= 75 AND Angle_1_2 <= 95) AND
AADT_NullCount = 0

(Look at Nodes where all Legs have AADT values)
How To “Automate”? 

- Bulk populate Major_1, Major_2, etc. fields indicating Leg rank and corresponding NodeLegID

- From previous query, Legs 1-4 (sorted by AADT) are: Major_1, Minor_1, Major_2, Minor_2 (respectively)

- In other words:
  - Major_1, Minor_1, Major_2, Minor_2 are Legs 1, 3, 2, 4

<table>
<thead>
<tr>
<th>Major_Leg</th>
<th>Major_Leg_2</th>
<th>Minor_Leg</th>
<th>Minor_Leg_2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>
ArcMap AddIns

- **HMS Toolbar**
  - **Node-centric tools** – select intersecting Road Centerline arcs and (depending on tool) the Node its NodeLegs
    - LegsOnly
    - AddAll
    - UpdateAll
    - UpdateNodeLegCount
  - **Centerline-centric tools** – select any number of Road Centerline arcs
    - CheckRelationalAttributes
    - CheckRoadwayAttributes
    - UpdateRelationalAttributes
    - UpdateRoadwayAttributes
ArcMap AddIns

Before Road Centerline Edits

After Road Centerline Edits

After Running UpdateAll (attributes too!)

UpdateAll Button

Intersection Toolbar
AfterSplitCreateNode CheckRelationalAttributes CheckRoadwayAttributes
LegsOnly AddAll UpdateAll UpdateRelationalAttributes UpdateRoadwayAttributes UpdateAllAttr UpdateNodeLegCount
Future Directions

- Complex (or all!) Intersections in Tables
  - Geometry of Intersections (points) and Approaches (lines) = geometry of Principal Nodes and Principal NodeLegs
  - Attributes represent the entire intersection or approach direction

(Nodes and NodeLegs datasets reserved for feature-specific attributes)
Future Directions

- Maintain rules for summarizing Intersections (and complex approaches) within Table attributes
  - Add AADT of dual carriageways
  - Average Leg skew angles
  - Represent the entire intersection or approach direction
Thanks! Questions?

VTrans Mapping Unit:
Johnathan Croft
Sara Moulton
Michael Trunzo

Contact:
Kerry.Alley@vermont.gov