Topology in the Geodatabase: An Introduction

Colin Zwicker
Erik Hoel

ESRI Super Secret Topology Laboratory, May 2017
Agenda

- ArcGIS Topology defined
- Validating a topology
- Editing a topology
- Geoprocessing tools
- Ephemeral topologies
- Data model design and performance
- Topology at 10.1 and 10.3

- Various cheesy demos
ArcGIS Topology

• Fundamentally, topology is used to manage how a set of simple feature classes share geometry
What Happens to the Features?

- Nothing
- No change to storage of features
- No change to behavior of features

So, what changes?
What Changes with a Topology?

- Topology tracks edits you make to the features
- Flags areas that have been modified
- **Validate Topology** command analyzes the edited areas and evaluates the rules
- If a rule is violated, a topology error feature is created
- You correct the errors or mark them as exceptions to the rule (or you may ignore the errors altogether)
Topology Rules

- Define integrity rules and constraints for the feature classes
- Can include point, line and polygon features
- Evaluated when the topology is validated
- Violations are expressed as topology errors
- 31 different rule types
  - 14 ArcGIS Topology rules can emulate coverage model
  - More may follow
- It is not possible to create custom rules
Rules to Model Coverage Topology

- **Arc rules**
  - Line must not have dangles
  - Line must not intersect or touch interior
  - Line must not have pseudos
  - Line must be single part
  - Line must not self-intersect

- **Arc-Node rules**
  - Line endpoint must be covered by point
  - Point must be covered by line endpoint

- **Polygon rules**
  - Polygon must not have gaps
  - Polygon must not have overlaps
  - Polygon must be covered by line
  - Line must be covered by boundary of polygon

- **Region rules**
  - Polygon must be covered by boundary of polygon

- **Route rules**
  - Line must be covered by feature class of line

- **Label Rules**
  - Point must be properly inside polygon
Dirty Areas

• Define locations where the spatial integrity of the topology is unknown
  - I.e., validate has not been run following edits
  - Used to optimize the validate process

• May contain existing or undiscovered topology errors
  - A clean area does not contain any undiscovered errors

• Stored in the geodatabase

• Created as a byproduct of editing features

• Created when shape or subtype changes
Cluster Tolerances

- Cluster tolerance is a property of the topology
  - Cannot be set on a per feature class basis
- Minimum cluster tolerance is the default
- Support separate cluster tolerances for XY and Z
Cluster Tolerances

• NOT an upper bound for point movement
• Instead, a lower bound for line to line and point to line separation

• Used during validation to ensure:
  - No two points are closer than the cluster tolerance
  - No point is closer than the cluster tolerance to a line of which it is not an endpoint
  - No two lines intersect except at their endpoints
Topology Ranks

- Each feature class has a rank
- Ranks are used to establish a priority ordering
- Vertex with the highest rank determines the cluster location
- Nothing is sacred
- Ranks supported for
  - XY
  - Z
Validating a Topology
Validation

• Integrates the geometries:
  - Segmentize the geometries
  - Cracking and clustering
    (this may update the geometries of the features)

• Evaluates topology rules
• Generates topology errors
• Deletes errors if rules are no longer violated
• Removes validated portions of dirty area
Cracking and Clustering

• Can go through a number of iterations
  - Maximum of six iterations

• In each iteration, a point may move no further than the cluster tolerance times the square root of two
  - Maximum movement is $6 \times \text{tolerance} \times \sqrt{2}$
  - The probability of this occurring is completely negligible
  - Point movement should be considered a statistical parameter, much like the standard deviation for a normal distribution
Validation

- Any part of the dirty area can be validated
- Not required to validate before saving changes or reconciling
- Users can define their own workflow
  - Validate after every edit operation
  - Validate once prior to saving
  - Validate once a week
Validation

• A vertex may move up to the square root of 2 (~1.4) times the cluster tolerance during each iteration of cracking and clustering
  - Thus, two vertices that are separated by ~2.8 times the cluster tolerance may be clustered together

\[ \text{Cluster tolerance} \times 1.4 \times 2 = 2.8 \]
Three endpoint vertices and one line with specified priorities

Black (insert point V6 vertices in polyline while V1 is in feature class with higher rank)
Z-Aware Data

• Two different models based on Z-cluster tolerance

• “Terrain” model
  - All coincident vertices share a single elevation
  - Use a large Z-cluster tolerance

• “Building” model
  - Coincident vertices may or may not have common elevation value
  - Use a smaller Z-cluster tolerance
Clustering Z-Aware Features

If neither point moved farther than the cluster tolerance, the cluster is formed.

Because the two vertices were of the same rank, we average the Z-values:

\[ Z = \frac{15 + 10}{2} = 12.5 \]

Vertices of lower ranks never influence vertices of higher ranks, only the lower rank Z can be adjusted.

Because the Z of the lower rank vertex did not have to move more than the tolerance, the cluster is formed and it inherits the Z of the higher vertex.

(note: \( \sqrt{2} \) factor does not apply to Z clustering process)
Topology Errors

- Created during validate when a topology rule is violated
- Contain information necessary to:
  - Understand the rule violation
  - Draw the rule violation
- Have geometry based on violated rule type
Topology Errors

• Visible through the topology layer

• Cannot be deleted by the user

• User has three options:
  - Leave the error
  - Fix the error
  - Elevate the error to exception status
    - E.g., rule applies everywhere except ‘here’
Editing a Topology
Hopefully, another decent demo

Colin Zwicker
Versioning a Topology
Versioning

- Topology is fully supported in the versioned environment
  - Multiple simultaneous editors of same feature

- No restriction on reconciling and posting only fully validated topologies
  - Dirty areas and errors correctly handled

- Reconcile will result in the areas containing new or updated features becoming dirty
  - Subsequent validation is fast as integrate (cracking and clustering) typically does not need to update the feature geometries
Versioning Example

Validate the new dirty area in DEFAULT Example with the No Overlaps rule
Create a child version off of DEFAULT
Edit polygon P2 in the child version
Edit DEFAULT – create new polygon P3
Validate the new dirty area in the child version
Reconcile child with DEFAULT
Validate dirty area – error detected
Versioning Workflow

- Workflow is critical with versioning

- Recommended workflow:
  1. Load all the data
  2. Define the topology
  3. Validate the entire topology
  4. Finally, register the topology as versioned

- Bonus question: Why?
Geoprocessing Tools
Geoprocessing Tools

• There are a number of GP tools available that can be used to:
  - Manage topologies
  - Validate topologies
  - Perform bulk topological updates
  - Output new feature classes based upon topological operations

• Like all Geoprocessing tools, these are scriptable in Python, and may be used with the GP Service
Managing Topologies

- There are many GP tools that can be used to manage topologies:
  - Add Feature Class
  - Add Rule
  - Create
  - Remove Feature Class
  - Remove Rule
  - Set Cluster Tolerance
  - Validate
Geoprocessing Tools

• There is a variation of the Validate operation called “Integrate”, but only applies the cracking and clustering tolerance to a collection of feature classes
  - It updates the geometry of every feature within each feature class as a result of running all of the features through the Topology Engine

• At 10.0, there are GP tools to perform bulk editing
  - These are basically the ARC/INFO capabilities found within ArcEdit
  - Some of the more interesting are “Extend Line” that extends undershoots, and “Trim Line” that removes overshoots (dangles)
Geoprocessing Tools

• These tools output new feature classes based upon basic topological operations:
  - Feature To Line
    - Generates a planar output of the lines of a collection of input polygons and/or lines (e.g., generating the Arcs)
  - Feature To Polygon
    - Generates a planar output of the polygons of a collection of input polygons and/or lines with a single point feature class providing the attributes (like ARC/INFO CLEAN)
  - Polygon To Line
    - Generates the planar output of lines from a single polygon feature class assigning the left and right of the polygon to the lines (like the Arcs of a Coverage)
Ephemeral Topologies

• Topologies that are short lived and intended to support QA/QC processes on data prior to incorporation into another controller dataset
  - E.g., geometric networks, network datasets, or utility networks (at 10.6/2.1)

• Once data is considered good, the topology is deleted and the participant features are then incorporated into another dataset
Data Models and Performance
Good Ideas

• Choose your spatial domain carefully
  - Resolution should be much smaller than data capture accuracy

• Use the default cluster tolerance

• Couple with network dataset for network routing

• Model only those relationships that are important

• Minimize class count and utilize subtypes
  - Cursors are expensive
  - Pro 2.1 will introduce Subtype Layers that will aid this process
Good Ideas

• Avoid classes containing small number of large features
  - Nation or state polygon

• Consider periodic GP tools for QA/QC
  - Avoidance of large polygons (e.g., nation or state)

• Always prototype the data model
Bad Ideas

• Using a topology with a very large cluster tolerance to clean up bad data
  - Instead, use GP tools (e.g., Integrate)

• Coincident geometrically identical features
  - Stack of polygons representing a multi-story condo with no-overlap rule

• Extremely small features
  - Approximately cluster tolerance size
Topology at 10.1

• Support for schema changes to versioned topologies without having to un-version the feature dataset
  - Add a feature class to or remove one from the topology
  - Add or remove a topology rule
  - Change tolerance or rank

• New GP tool (Export Topology Errors) that allows you to export topology errors as feature classes

• Improved editing tools for maintaining topological data
  - According to Colin, some users are stoked...
Topology today

- Topology was not supported in the early releases of ArcGIS Pro
  - It is supported in ArcGIS Desktop 8.3

- Topology is supported in ArcGIS Pro 1.3
Topology Summary
Summary

• Very flexible
  - You pick the topology rules
  - You decide how to handle errors
  - You control the workflow
  - You can extend the functionality

• Easy to use
  - Wizard for defining
  - Topological editing tools
  - Error correction tools
Summary

• **High performance**
  - User controlled validation process
  - Incremental validation

• **Scalable**
  - Nationwide datasets
  - Versioned RDBMS

• **Complete geodatabase integration**
  - Versioning
  - Copy/paste, XML import/export
  - Replication, distributed geodatabases
  - ArcGIS Server