Caveats

- Presumed knowledge of the Geodatabase

- Concentrate on key issues
  - Plenty of documentation (printed and online) that covers basic material
  - Finish up by talking about what’s in 10.1

- Lots of material, little time
  - solution: talk fast, ignore questions
Agenda

• Overview of the Model
• Editing and Analyzing
• Versioning
• Performance and Other Key Issues
• What’s in 10.1
Geometric Networks

• Motivated by utility and natural resources industries

• Contain edges and junctions

• Connectivity relationships between network feature classes
  - Connectivity based upon geometric coincidence of vertices
  - Connectivity represented in a connectivity index
  - Connectivity is always maintained

• All participating features are custom (i.e., not simple features)
Network Feature Classes

- Network features only live in a geometric network

- Three types:
  - Simple junction
  - Simple edge
  - Complex edge

- Orphan junction feature class
  - Used to maintain integrity
    - Edges must always have a junction at their endpoints
  - System controlled – do not add attributes, etc.
Logical Network

- Physical representation of network connectivity – an index

- High performance graph engine
  - Allows fast network traversals
  - Very compact and optimized
  - Connectivity and weights (attributes) stored in BLOBs

- Analysis (e.g., tracing) is performed within the logical network

- Also used with the Network Dataset
Simple Versus Complex Edges

• Simple edges
  - No mid-span connectivity
  - Resources flow from endpoint to endpoint
    - e.g., service laterals, driveways, city streets

• Complex edges
  - Allow mid-span connectivity
  - Resources flow along, but may be siphoned off periodically
    - e.g., water mains, highways

• Deciding whether a feature class should be simple or complex?
  - Ask yourself whether resources will be siphoned along the edge
## Comparison

<table>
<thead>
<tr>
<th>Network Dataset</th>
<th>Geometric Network</th>
</tr>
</thead>
<tbody>
<tr>
<td>transportation</td>
<td>utilities/natural resources</td>
</tr>
<tr>
<td>pathfinding and allocation operations</td>
<td>network tracing functionality</td>
</tr>
<tr>
<td>turns supported</td>
<td>turns not supported</td>
</tr>
<tr>
<td>uses simple features: points and lines</td>
<td>uses custom features: simple/complex edge features and junctions</td>
</tr>
<tr>
<td>more robust attribute (weight) model</td>
<td>weights based on feature attributes</td>
</tr>
<tr>
<td>user controls when connectivity is built</td>
<td>system automatically maintains connectivity</td>
</tr>
</tbody>
</table>
Demo

Creating a geometric network
Editing and Analyzing
Editing

- Same workflow as editing simple features
  - Specific tools/commands on the Geometric Network Editing toolbar

- Connectivity maintained by the GN
  - Based on geometric coincidence of vertices

- Use snapping and the Feature Cache (nee Map Cache)

- Junction subsumption
  - Snapping junctions to orphan junctions

- Exhaustive network editing examples in the Help
  - See About Editing Geometric Network Features for more examples
Flow Direction

- **Setting Flow Direction**
  - Within an Edit Session
  - Must have at least one Simple Junction with an Ancillary Role field
  - Do this after:
    - Network creation
    - Feature creation or change in connectivity of existing features
    - Source/sink changes

- **Does not follow digitized direction by default (prior to 10.1)**
  - New GP tool at 10.1 supports setting this

- **Arrows are drawn at mid-point on the edge features**
Flow Direction - Indeterminate Flow

- Multiple sources and sinks cause conflicting flow direction
  - Yields indeterminate flow direction
- Consider the following case where edge 3 has indeterminate flow
Flow Direction - Indeterminate Flow

- Consider flow direction when only the Source is set

- Consider flow direction when only the Sink is set
Flow Direction

- This results in a conflict

Flow direction

- If the flow direction is in agreement between both the source-only and sink-only cases, the flow direction is set to that direction
- If the flow direction is in conflict between the source-only and sink-only cases, flow is set to indeterminate

How to set flow direction manually?

- Samples from ArcObjects Online

Knowledge Base Article 20685
Network Connectivity and Verification Tools

• **Rebuild** connectivity tool
  - Selectively recreate all connectivity over an area

• **Repair** connectivity command (intended for larger areas)
  - Correct connectivity within a network
  - Does not require entire rebuild of network connectivity, only affects features with inconsistent connectivity
  - Warnings can be raised
  - Optional log file can be created

• Operate on network being edited for File Geodatabases; entire version for Enterprise Geodatabases
Demo

Creating new network features, editing existing features, and performing tracing with flow direction
Versioning

- Geodatabase uses an optimistic concurrency approach
- No locks applied when features/objects modified
  - Other editors may edit same features, at the same time
- Introduces the potential for feature conflicts
- A conflict may occur when
  - Two editors are editing the same data in the same version at the same time
  - The same feature is modified in two different versions
- How to manage this?
  - Use workflow management to prevent conflicts
  - Manage the conflicts once they occur
Versioning – Rules for Reconcile

1. GN editing rules apply to Reconcile / Conflict replacement
   a. i.e. Orphan junctions cannot subsume each other
2. New features are not created during Reconcile
3. Conflicts result if same features modified in two versions
   a. Update could be to connectivity and/or to geometry/attributes
   b. Conflicts may be propagated due to connectivity changes
      i. Features may be in conflict even though not directly edited in both versions
   c. Newly created features may also be propagated to conflicts
4. Changes **only to the connectivity** of a feature in two versions will not result in conflicts
   a. Reconcile will filter these features
5. Disconnected state of a features is not considered/maintained
Versioning – Rules for Conflict Management

1. GN editing rules apply to Reconcile / Conflict replacement
   a. i.e. Orphan junctions cannot subsume each other
   b. Restoring features:
      i. Restoring an edge restores the endpoint junctions
      ii. Restoring a junction will not restore connected edges
   c. Removing features:
      i. Removing an edge will not remove the junctions
      ii. Removing an endpoint junction will remove the edge

2. Conflict resolution can create new features
   a. Default junctions from connectivity rules are honored
Versioning – Scenarios

- Two versions, Edit and Target
- The current Edit version is a child of the Target version
- Edit version is reconciled against Target version
- The default behavior will be for the features in Target version to take precedence over the features in Edit version
Versioning – Scenarios

1. Change geometry of a complex edge in two versions
2. Change connectivity of a junction in two versions
3. Delete a feature in the Target version, change it in the Edit version
4. Update a network weight in the Target version, change the connectivity of the feature in the Edit version
Versioning – Scenario 1

• Target Version
  - A standard junction is added (vertex also added)
Versioning – Scenario 1

- **Target Version**
  - A standard junction is added (vertex also added)

- **Edit Version**
  - A simple edge is added to the same complex edge
Versioning – Scenario 1

• Target Version
  - A standard junction is added (vertex also added)

• Edit Version
  - A simple edge is added to the same complex edge

• Reconcile
  - Update-update conflict on the horizontal edge

Change geometry of a complex edge in two versions
Versioning – Scenario 1

- **Target Version**
  - A standard junction is added (vertex also added)

- **Edit Version**
  - A simple edge is added to the same complex edge

- **Reconcile**
  - Update conflict on the horizontal edge

**Common Ancestor**

**Versioning – Rules for Reconcile**

1. GN editing rules apply to Reconcile / Conflict replacement
   - i.e. Orphan junctions cannot subsume each other
2. New features are not created during Reconcile
3. Conflicts result if same features modified in two versions
   - a. Update could be to connectivity and/or to geometry/attributes
   - b. Conflicts may be propagated due to connectivity changes
     - i. Features may be in conflict even though not directly edited in both versions
   - c. Newly created features may also be propagated to conflicts
Versioning – Scenario 1

• **Target Version**
  - A standard junction is added (vertex also added)

• **Edit Version**
  - A simple edge is added to the same complex edge

• **Reconcile**
  - Update-update conflict on the horizontal edge
  - Due to the geometry as well as the connectivity being modified on each
Versioning – Scenario 2

• Target Version
  - A simple edge is deleted
Versioning – Scenario 2

- **Target Version**
  - A simple edge is deleted

- **Edit Version**
  - An adjacent edge is added

![Diagram showing change connectivity of a junction in two versions]
**Versioning – Scenario 2**

- **Target Version**
  - A simple edge is deleted

- **Edit Version**
  - An adjacent edge is added

- **Reconcile**
  - No conflicts are detected

---

**Change connectivity of a junction in two versions**

- **Common Ancestor**
- **Edit Version**
- **Target Version**

*reconcile*
Versioning – Scenario 2

- Target Version
  - A simple edge is deleted

- Edit Version
  - An adjacent edge is added

- Reconcile
  - No conflicts are detected

Versioning – Rules for Reconcile

4. Changes only to the connectivity of a feature in two versions will not result in conflicts
   a. Reconcile will filter these features

5. Disconnected state of a feature is not considered/maintained
Versioning – Scenario 2

- **Target Version**
  - A simple edge is deleted

- **Edit Version**
  - An adjacent edge is added

- **Reconcile**
  - No conflicts are detected
  - Only the connectivity of the highlighted junction has changed; Reconcile filters any conflict
Versioning – Scenario 3

- **Target Version**
  - An orphan junction is deleted (along with simple edges)

Delete a feature in the Target version, change it in the Edit version

Common Ancestor

Target Version
Versioning – Scenario 3

- **Target Version**
  - An orphan junction is deleted (along with simple edges)

- **Edit Version**
  - An adjacent edge is added

- **Delete a feature in the Target version, change it in the Edit version**

Common Ancestor

Edit Version  Target Version
Versioning – Scenario 3

- **Target Version**
  - An orphan junction is deleted (along with simple edges)

- **Edit Version**
  - An adjacent edge is added

- **Reconcile**

Delete a feature in the Target version, change it in the Edit version.
Versioning – Scenario 3

- **Target Version**
  - An orphan junction is deleted (along with simple edges)

- **Edit Version**
  - An adjacent edge is added

- **Reconcile**
  - Delete-update conflict on the junction
  - Conflict propagation on the new edge

Delete a feature in the Target version, change it in the Edit version

Common Ancestor

Edit Version

Target Version

reconcile

delete-update and conflict propagation
Versioning – Scenario 3

• **Target Version**
  - An orphan junction is deleted (along with simple edges)

• **Edit Version**
  - An adjacent edge is added

• **Reconcile**
  - Delete junction
  - Conflict propagation on the new edge

---

**Versioning – Rules for Reconcile**

1. GN editing rules apply to Reconcile / Conflict replacement
   a. i.e. Orphan junctions cannot subsume each other
2. New features are not created during Reconcile
3. Conflicts result if same features modified in two versions
   a. Update could be to connectivity and/or to geometry/attributes
   b. Conflicts may be propagated due to connectivity changes
      i. Features may be in conflict even though not directly edited in both versions
   c. Newly created features may also be propagated to conflicts

---

**Delete a feature in the Target version, change it in the Edit version**
Versioning – Scenario 4

- Target Version
  - ENABLED value is updated on complex edge

Update a network weight in the Target version, change the connectivity of the feature in the Edit version
Versioning – Scenario 4

- **Target Version**
  - ENABLED value is updated on complex edge

- **Edit Version**
  - orphan junction on same complex edge is deleted

Update a network weight in the Target version, change the connectivity of the feature in the Edit version

Common Ancestor

Edit Version

Target Version
Versioning – Scenario 4

- **Target Version**
  - ENABLED value is updated on complex edge

- **Edit Version**
  - orphan junction on same complex edge is deleted

- **Reconcile**
  - Update-update conflict on the horizontal edge
  - Update-delete conflict on junction

Update a network weight in the Target version, change the connectivity of the feature in the Edit version
Versioning – Scenario 4

- **Target Version**
  - ENABLED value is updated on complex edge

- **Edit Version**
  - orphan junction on same complex edge is deleted

- **Reconcile**
  - Update-update conflict on the horizontal edge
  - Update-delete conflict on junction due to conflict propagation

Update a network weight in the Target version, change the connectivity of the feature in the Edit version.
Versioning – Scenario 4

- **Target Version**
  - ENABLED value is updated on complex edge

- **Edit Version**
  - Orphan junction on same complex edge is deleted

- **Reconcile**
  - Update conflict on horizontal edge
  - Update delete conflict on junction reconcile

**Versioning – Rules for Reconcile**

1. GN editing rules apply to Reconcile / Conflict replacement
   a. i.e. Orphan junctions cannot subsume each other
2. New features are not created during Reconcile
3. Conflicts result if same features modified in two versions
   a. Update could be to connectivity and/or to geometry/attributes
   b. Conflicts may be propagated due to connectivity changes
      i. Features may be in conflict even though not directly edited in both versions
   c. Newly created features may also be propagated to conflicts
Versioning - Recommendations

• Use workflow management techniques to prevent conflicts
  - Avoid editing features in multiple locations in same session
  - Avoid changing large/long features in different versions
  - Plan for bulk updates or edits

• Manage the conflicts once they occur
  - Use different Reconcile options
    - Define conflicts “By Attribute”
    - “In favor of the Edit Version”
  - Resolve conflicts at the top level or class level
    - Resolve junctions first to avoid errors
  - Still can’t resolve the conflict?
    - Consider moving on, and re-doing the edits in another version
Performance and Other Key Issues
• Use Logical Network API for navigation and tracing whenever possible
  - IForwardStar
• Navigational APIs available at the Geometric Network feature level
  - Intended for small tactical navigation
• Analysis algorithms should always consume the Logical Network APIs
  - Several orders of magnitude faster
  - INetwork, INetTopology, ...
Performance

- Connectivity maintained on the fly
  - Connectivity based upon coincidence
  - When adding a new feature, all other network feature classes are searched
  - Use the feature cache

- Minimize the number of network feature classes
  - Utilize subtypes

- Subtypes not for you? Consider lumping of classes
  - Handle unpopulated attributes

- Data model structure is critical
  - Empty classes as expensive as heavily populated
  - Relationship messaging and event handling
Dropping Networks

• Why?
  - Add a new populated class
  - Snapping tolerance too small on previous build

• What happens?
  - Network classes revert to simple classes
  - Network index (logical network) deleted
  - Orphan junction class will be deleted
  - Re-specify connectivity rules and weights
  - Enabled and ancillary role fields retained
  - If snapped during first build, may not need to specify snapping again
Preparing Your Data for the Geometric Network

• Ideally, your data is clean before you build a network
  • Features that should be connected are geometrically coincident
  • no overshoots or undershoots

• If your data is not clean or you are not sure, you can use one of the following workflows:
  • Enable snapping during the network creation
  • Use Topology to find and correct errors

• May still encounter invalid geometries if either method is used
Preparing Your Data for the Geometric Network

- Enable snapping during the network creation
  - Good option if:
    - You’re confident with your overall data quality
    - Minor corrections are needed in your data

- Use Topology to find and correct errors
  - Good option if:
    - You’re unsure of your overall data quality
    - Know that major edits and corrections are needed in your data to ensure geometric coincidence

- More rules available that help to discover common data errors for geometric networks
Prototyping

- **Largest mistake made with the Geodatabase**

- **Structure is critical – data quantity is not**

- **Prototype as soon as a first pass model is available**
  - General structure; small details unimportant
  - Load a modest amount of data (on versioned SDE)
  - Empty classes are OK

- **Try editing, observe system performance**

- **Repeat this process as necessary**
What’s in 10.1
What’s New at 10.1

• Geometric Network functionality available through geoprocessing

• Geometric Network creation and management
  - Creation of network and ability to remove empty feature classes
  - Connectivity rule management

• Network Tracing
  - Trace and Set Flow Direction

• Persist settings made to the Utility Network Analyst toolbar in map documents
Steps to Evaluate UC Sessions

- My UC Homepage > “Evaluate Sessions”
- Choose session from planner
  OR
- Search for session
- Offering IDs
  - Wednesday – 716
  - Thursday – 862

www.esri.com/ucsessionssurveys
Questions?